

# Big Data Sandbox Workbook

ENTERPRISE EDITION ONLY

**The Big Data Sandbox is the easiest way for you to try the latest version of Pentaho with Hadoop in a virtual machine (VM) environment. With this workbook, you will get step-by-step exercises to help guide you through ingesting, onboarding, blending, and preparing data within Hadoop.**

# Contents

Contents .....	2
Use Case 1: Fill the Data Lake .....	4
What is it? .....	4
Why do it? .....	4
Value of Pentaho .....	4
What you will accomplish.....	5
Fill the Data Lake Exercise 1: Loading raw data to HDFS .....	5
Fill the Data Lake Exercise 2: Explore how metadata injection can be used to automate data onboarding.....	9
Use Case 2: Create a Data Refinery .....	17
What is it? .....	17
Why do it? .....	17
Value of Pentaho .....	17
What you will accomplish.....	17
Create a Data Refinery Exercise 1: Transform CDR data and blend with geography data ...	18
Create a Data Refinery Exercise 2: Use Pentaho with Cloudera Impala to blend CDR and IoT data.....	31
Create a Data Refinery Exercise 3: Extend the PDI job to blend data and load to multiple locations.....	35
Create a Data Refinery Exercise 4: Explore data in PostgreSQL with Pentaho Analyzer .....	41
Use Case 3: Self-Service Data Preparation .....	45
What is it? .....	45
Why do it? .....	45
Value of Pentaho .....	45
What you will accomplish.....	45
Self-Service Data Preparation Exercise 1: Use Pentaho to prepare data for analytics .....	46
Use Case 4: Self-Service Analytics .....	57
What is it? .....	57
Why do it? .....	57
Value of Pentaho .....	57
What you will accomplish.....	57

Self-Service Analytics Exercise 1: Use Pentaho to visualize data .....	58
--	----

# Use Case 1: Fill the Data Lake

## What is it?

The data lake is the foundation for the modern data pipeline. It is the repository that stores a massive volume of raw data, including structured, semi-structured, and unstructured data. The data pipeline involves many steps for going from raw data to business value. The first step involves ingesting or onboarding the data. If the data onboarding process is not built or managed properly, the data lake can become a data swamp: a disorganized, dumping ground for data. This first use case shows you how to ingest data into Hadoop using a variety of methods including file copy, metadata injection, and Apache Kafka stream processing.

## Why do it?

Onboarding data has always had challenges, and the challenges are greatly exacerbated in a big data environment. Achieving maximum ROI on your data lake requires implementing efficient tools, processes, and architecture. Modern data onboarding challenges go beyond just “connecting” to data sources or “ingesting” data into a store of choice and introduce significant new challenges related to managing many more sources of data that may change over time. Modern onboarding tools require a flexible, efficient, and governed process to be fully successful.

## Value of Pentaho

Pentaho allows you to define an ETL template for the overall data workflow without needing to specify any of the metadata detail. At runtime, metadata can be fed into the workflow via a process called metadata injection. This injection allows hundreds of data sources to be managed with a single, generic workflow template. Using a single template reduces development time, risk, maintenance, and expense.

We see our customers leverage this process for a variety of use cases including:

- Scalable data ingestion
- Data migrations
- Self-service data onboarding
- Dynamic data discovery and parsing

## What you will accomplish

In this section, you will move records from local CSV files to HDFS. In the first exercise, you'll create a Pentaho job to copy the data files for use later. In the second exercise, you'll use metadata injection to copy log files with different formats into one common format. You will ingest the following data sources to Hadoop:

- Call detail records (CDR)
- Area geography master records
- Hitachi Content Platform (HCP) logs
- Tower log files of various formats


These sources are data files, but you could also use sources via Kafka or other stream processing frameworks.

## Fill the Data Lake Exercise 1: Loading raw data to HDFS

This first exercise helps you load the sources of raw data to Hadoop distributed file system (HDFS) so that the data can be processed in Hadoop in a later exercise. You can follow the exercise step by step to create the job using Pentaho, or you can open the pre-built job to review and run. The pre-built job is at:

```
/pentaho/evaluation/01_Fill_the_data_lake/solutions/Fill Data Lake  
Exercise 1.kjb
```

### Exercise Steps:

1. Launch the PDI client-based authoring tool (Spoon) from the launch menu icon  at the bottom of your screen. Click the icon just once to launch Spoon.



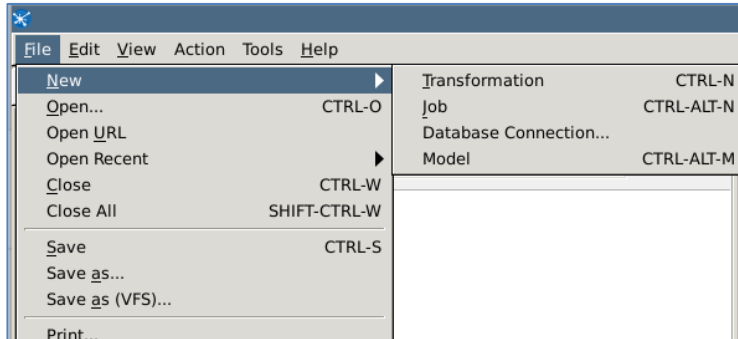
2. You should see the PDI splash screen appear while PDI loads.



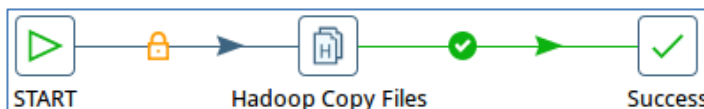
All open applications appear in the top left section of your screen. Once Spoon opens, you will see it as shown in the following screenshot:



- From the main menu choose **File | New | Job**.



- From the **Design** tab on the left, expand the **General** folder and drag the **START** and **Success** job steps onto the canvas.
- Expand the **Big Data** folder and drag **Hadoop Copy Files** onto the canvas between the **START** and **Success** steps.
- Connect the **Start** step and the **Hadoop Copy Files** step by creating a hop. A hop can be created by using the tooltip when highlighting over the step or by clicking the Shift key while moving the mouse from one step to the next.
- Create a hop between the **Hadoop Copy Files** step and the **Success** step to match the following image:



- Double-click on **Hadoop Copy Files** to open its properties.
- On the **Settings** tab check the following two items: **Create destination folder** and **Replace existing files**.
- In the **Files** tab select the cell beneath the **Source Environment** header on the **Files** tab and select **Local** from the drop-down list.
- Select the cell beneath the **Source File/Folder** step and select the button.

- Browse to the following file to select it, and then click the **OK** button to return to the **Copy Files** dialog box.

file:///pentaho/evaluation/01\_Fill\_the\_data\_lake/data/callrecords\_all.csv

13. Select the cell beneath the **Destination Environment** header on the **Files** tab and select **CDH** from the drop-down menu.

14. Select the cell beneath the **Destination File/Folder** step and enter:  
/BDO/callrecords/input

15. Repeat steps **10-14** to copy the **Source File/Folder**

file:///pentaho/evaluation/01\_Fill\_the\_data\_lake/data/callrecords\_10years.csv to the **Destination File/Folder** /BDO/callrecords/input

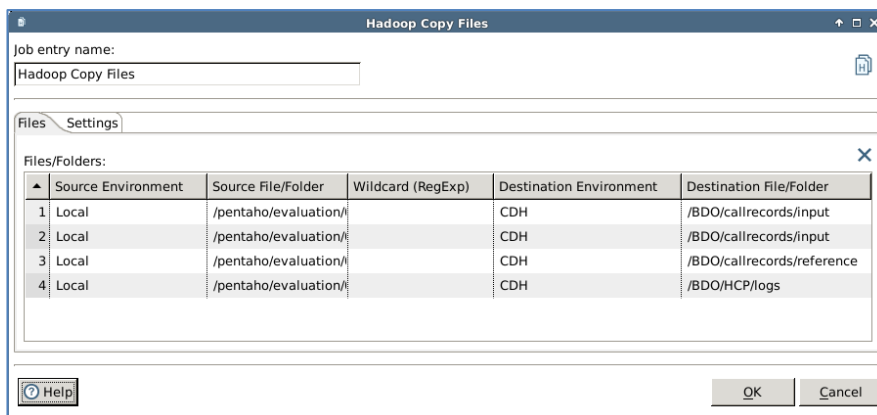
and

file:///pentaho/evaluation/01\_Fill\_the\_data\_lake/data/areacodes.csv to the **Destination File/Folder** /BDO/callrecords/reference

and

file:///pentaho/evaluation/01\_Fill\_the\_data\_lake/data/http\_gateway\_request.log.0 to the **Destination File/Folder** /BDO/HCP/logs

16. Your **Hadoop Copy Files** dialog box should match the following image (you can resize the window and table columns as needed).



17. Click **OK** to return to the canvas.

18. From the **File** menu, choose **Save**.

19. In the **Name** field, specify Fill Data Lake Exercise 1.kjb and save to the following location:  
/pentaho/evaluation/01\_Fill\_the\_data\_lake/student\_files

20. From the **Action** menu, choose **Run** and then click **Run**. On the **Job metrics** tab located in the bottom section of Spoon, you should see the job's progress.

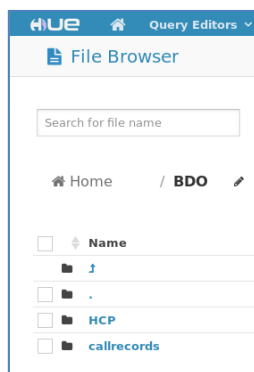
21. To view the newly copied CDR data in Hadoop, launch Firefox by single-clicking the Firefox icon at the bottom of your screen.



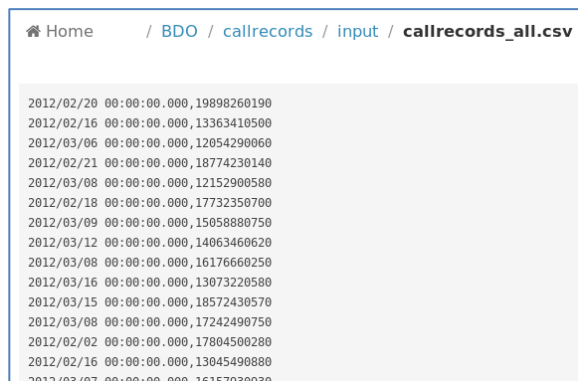
22. From the Firefox bookmarks bar click **Hue – File Browser**, and on the sign-in page, sign in with **demouser / demouser**.

23. Navigate to the following directory: `/BDO`

24. You should see your files in each of the specified folders. For reference, you can open the **Hadoop Copy Files** step in your PDI job and check the destination.



25. If your job executes successfully, you will see CDRs as shown in following screenshot:





## Fill the Data Lake Exercise 2: Explore how metadata injection can be used to automate data onboarding

This second exercise steps you through creating a transformation that uses the “metadata injection” method of ingestion. You will parse and ingest three different cell phone tower log CSV files into Hadoop. Each CSV file has a different format, so we leverage metadata injection to parse all three files and output them to Hadoop in one common format using only **one** transformation template. Each log file name will be matched to a sheet in a spreadsheet that contains log file metadata (e.g., field names, delimiter, enclosure, etc.). The metadata injection step injects that metadata into a transformation template to dynamically configure the CSV input and other steps.

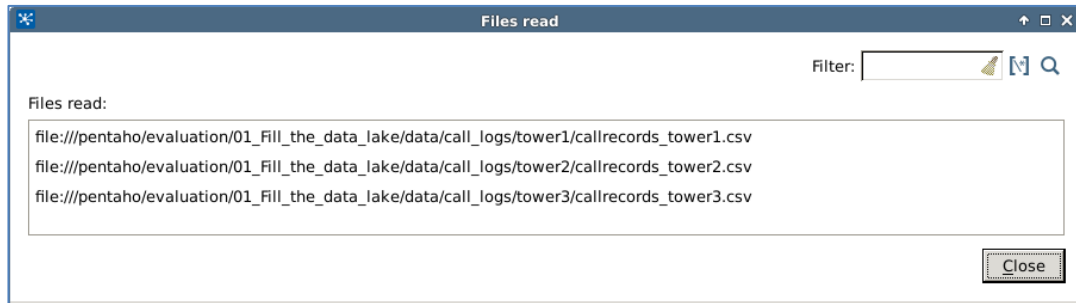
In this exercise, you will be creating two transformations named: `t_process_tower_logs` and `t_process_single_log_via_metadata_injection`. You will also modify the job that was created in exercise 1 and create `Fill Data Lake Exercise 1 – MI Updated`. You can follow the exercise step by step or you can open the pre-built transformations and job to review at: `/pentaho/evaluation/01_Fill_the_data_lake/solutions/` and run the job `Fill Data Lake Exercise 1 – MI Updated`.

### Exercise Steps:

1. In Spoon, create a new transformation.
2. From the **Input** folder, select and drag the **Get File Names** step onto the canvas.
3. Double-click the **Get File Names** step to update its properties.
4. In **File/Directory**, specify the directory where the files are stored:  
File: `///pentaho/evaluation/  
01_Fill_the_data_lake/data/call_logs`
5. Click **Add** to make sure it adds the directory under **Selected Files**.
6. In the **Selected Files** window, provide **Wildcard (RegExp)** = `.*csv*`
7. Ensure **Include Subfolders** is set to **Y**.

Selected files:			
File/Directory	Wildcard (RegExp)	Exclude wildcard	Require
1 /pentaho/evaluation/01_Fill_the_data_lake/data/call_logs/	.*csv		N

8. If everything is set up correctly, you should see a list of files when clicking on **Show filename(s)**.

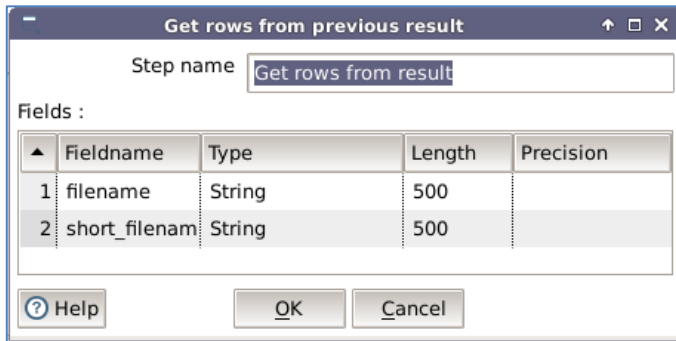


9. Click **Close**.
10. Next, you are going to add a **Transformation Executor** step in order to call another transformation you will be building. From the **Flow** folder in **Design** view, select and drag the **Transformation Executor** step onto the canvas.
11. Build a hop between the **Get File Names** and **Transformation Executor** steps.
12. Save this transform as `t_process_tower_logs`.



Now you are going to build the transform, which will be called by the **Transformation Executor** step above. In this transform, you will use a spreadsheet to track the metadata for each log file and dynamically inject it. You can then use one transformation to load all log files.

13. Start a new transformation.
14. From the **Job** folder, select and drag **Get rows from result** step onto the canvas. Double-click on the step to modify its properties. Add two field names: `filename` and `short_filename`. Set the **type** to `string`. Set the **length** to 500. The **properties** should resemble the screenshot below.



15. Change the **Step name** to Get Log File Name and click **OK**.

16. From the **Input** folder, select and drag the **Microsoft Excel Input** step onto the canvas. Double-click the step to modify its properties.

Note: No need to create any hops here yet.

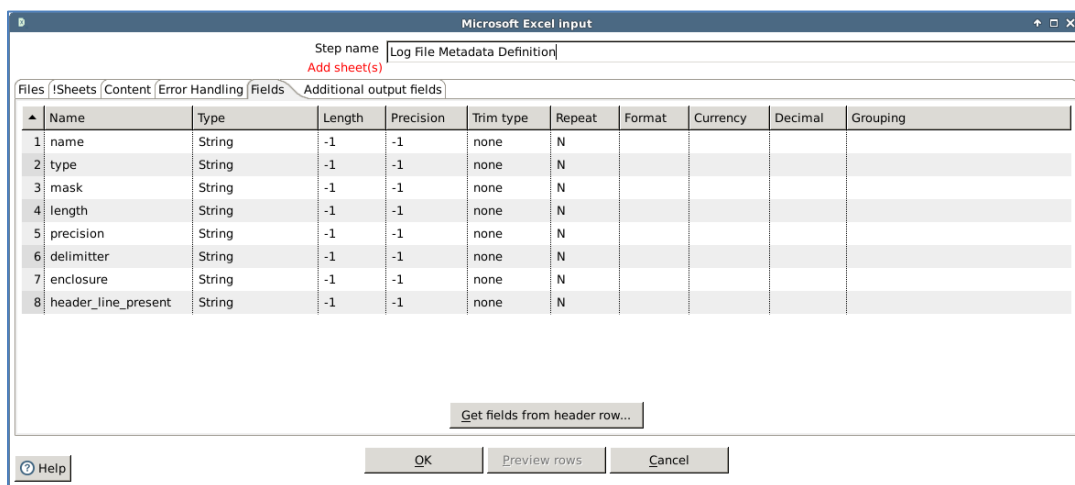
a. Name the step Log File Metadata Definition.

b. For the **Spread Sheet Type** (engine) set it to Excel 2007 XLSX (Apache POI) .

c. For **File/Directory** enter the following file and then click **Add**. The file is the spreadsheet that contains the metadata description of each log file.

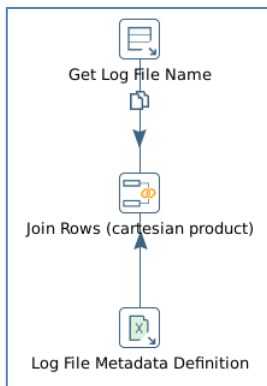
File:///pentaho/evaluation/01\_Fill\_the\_data\_lake/data/Tower\_logs\_metadata.xlsx

d. Click the **Fields** tab and then click **Get fields from header row...** to fill the fields. It should look like the screenshot below. If there are duplicate fields, then you will need to remove them.



e. Click the **Additional output fields** tab and enter log\_filename for Sheetname field.

- f. Click **OK**.
17. From the **Joins** folder, select and drag the **Join Rows (cartesian product)** step onto the canvas.
18. Create a hop between the **Log File Metadata Definition** step and the **Join Rows (cartesian product)** step. Create a hop between the **Get Log File Name** step to the **Join Rows (cartesian product)** step. The result should look like the image below:

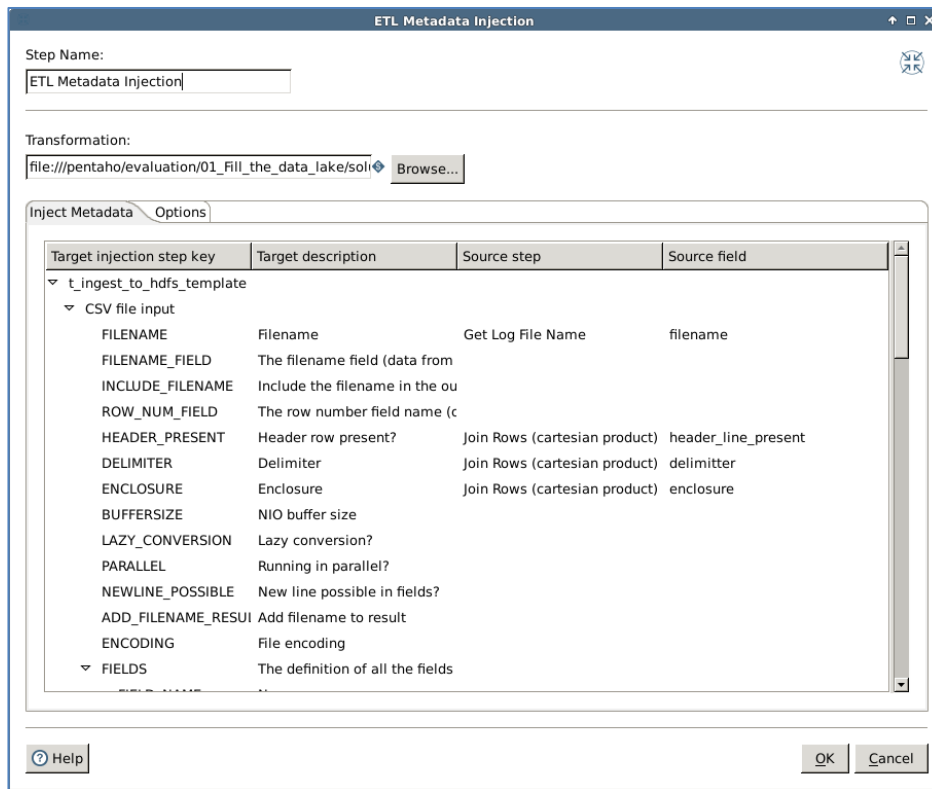


19. Double-click the **Join Rows (cartesian product)** step to set its properties.
20. Under **The condition** click the field and select `short_filename`. Click on the second field and select `log_filename`. The result should look like the following screenshot:

The screenshot shows the 'Join rows' configuration dialog box. The 'Step name' is 'Join Rows (cartesian product)'. The 'Temp directory' is '%%java.io.tmpdir%%' with a 'Browse...' button. The 'TMP-file prefix' is 'out'. The 'Max. cache size (in rows)' is '500'. The 'Main step to read from' is empty. Under 'The condition:', there is a table with two rows: the first row has 'short\_filename' in the first column, '=' in the second, and 'log\_filename' in the third; the second row has empty fields. At the bottom are 'Help', 'OK', and 'Cancel' buttons.

21. Click **OK**.
22. From the **Flow** folder, select and drag the **ETL Metadata Injection** step onto the canvas.
23. Create a hop between **Get Log File Name** and **ETL Metadata Injection**. When you see the warning message, click **Copy**.
24. Create a hop between **Join Rows (cartesian product)** and **ETL Metadata Injection**.

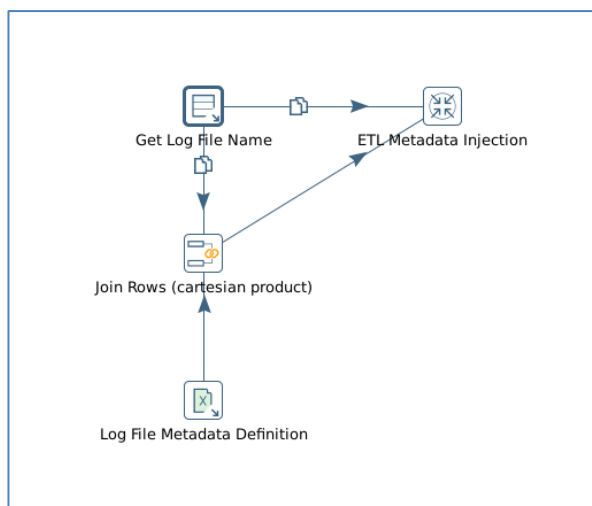
25. Double-click the **ETL Metadata Injection** step to modify its properties. Note: You may get an error on this step. Please click **OK** if you get an error.
26. In this step, you will be referencing a predefined template. Under **Transformation** click on **Browse** to select the following file:
- [file:///pentaho/evaluation/01 Fill the data lake/Solutions/t ingest to hdfs template.ktr](file:///pentaho/evaluation/01%20Fill%20the%20data%20lake/Solutions/t%20ingest%20to%20hdfs%20template.ktr)
- The template transform inputs the incoming file and writes it into a standard comma separated format (without headers) in HDFS.
27. Under the **Inject Metadata** tab inside the properties window, set the properties listed below. This will set the configuration for the CSV file input step of the transformation being called. The fields are dynamically determined for each log file.
- a. Click **FILENAME** and select **Get Log File Name : filename.**
  - b. Click **HEADER\_PRESENT** and select **Join Rows (cartesian product) : header\_line\_present.**
  - c. Click **DELIMITER** and select **Join Rows (cartesian product) : delimiter.**
  - d. Click **ENCLOSURE** and select **Join Rows (cartesian product) : enclosure.**
28. If everything is set up correctly, you should see the metadata as in the image below:



29. Click **OK** to close the **properties** window.

30. Save this transform to:

file:///pentaho/evaluation/01\_Fill\_the\_data\_lake/student\_files as  
t\_process\_single\_log\_via\_metadata\_injection



31. You are done with this transform. Note: **Please don't run this transform.** It will be executed by the t\_process\_tower\_logs transform you built earlier.

32. Let's get back to the **t\_process\_tower\_logs**. We need to modify the **Transformation Executor** step in that transform to execute the **t\_process\_single\_log\_via\_metadata\_injection** transformation.

33. In the **t\_process\_tower\_logs** transformation double-click on the **Transformation Executor** step to update the properties.

a. **File Name:**

```
file:///pentaho/evaluation/01_Fill_the_data_lake/student_files/t_process_single_log_via_metadata_injection.ktr
```

b. Parameters:

Parameters			Row grouping	Execution results	Result rows	Result files
	Variable / Parameter name	Field to use	Static input value			
1	filename	filename				
2	short_filename	short_filename				

c. Make sure that the **Inherit all variables** from the transformation box is checked.

34. Save and execute this transformation.



35. Using Hue, navigate to `/BDO/callrecords/input` to see the files that have been copied. You can click to view each file to see that they now have a common format.

36. The last thing you have to do in this exercise is modify the job you built in exercise 1 to take advantage of the metadata injection process you created.

37. Open the **Fill Data Lake Exercise 1** job from your student files.

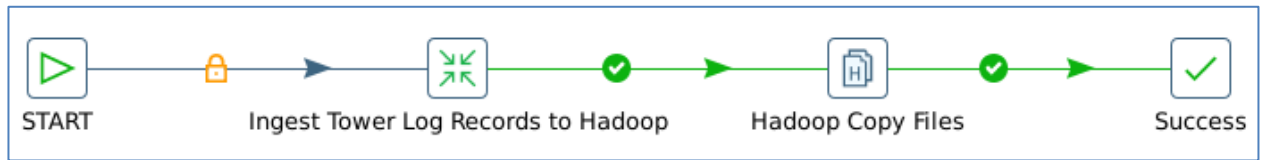
38. From the **General** folder select and drag the **Transformation** step onto the canvas.

39. Set that step between the **START** and **Hadoop Copy Files** steps. To insert a step, you will need to delete the original hop and then create a hop between **START** and **Transformation** as well as **Transformation** and **Hadoop Copy Files**.

40. Double-click the **Transformation** step. Rename it to `Ingest Tower Log Records to Hadoop`.

41. The transformation filename should be `t_process_tower_logs.ktr`. Make sure to browse to select the transformation you built earlier in this exercise. You will need to provide the full path.

42. Your job should look like this now:



43. Save this job as **Fill Data Lake Exercise 1 – MI\_Updated**.

44. Execute the job.



# Use Case 2: Create a Data Refinery

## What is it?

A data refinery is an enterprise solution for processing and blending data to make it governed, analytics-ready, and on-demand. The data refinery is powered by on-demand orchestration for blending traditional data and big data, and it is a first step toward governed data delivery. Governed data delivery is defined as the delivery of blended, trusted, and timely data to power analytics at scale regardless of data source, environment, or user role. It lays the groundwork for seamless end user exploration and analysis of validated data blends from across the organization.

## Why do it?

There are three core data delivery needs that are only being met on a limited basis in the market today:

- Orchestrate on-demand processing, blending, and modeling of user requested data sets to accelerate time to value in complex analytics initiatives
- Ensure proper data governance during the delivery process, such that risk is minimized and confidence is increased when making data-driven decisions
- Provide blended and enriched data in the end users' format of choice, so that business users can be more productive in deriving insight from diverse data

## Value of Pentaho

Pentaho's highly scalable data integration engine, managed through its intuitive end-user interface, provides the "glue" between the different data sources and big data stores in this architecture. The entire data integration process can be triggered on-demand with the following key capabilities: blending and orchestration, automatic modeling and publishing, and governance.

## What you will accomplish

You will complete **four exercises** to create a data refinery that processes and blends a combination of data sources from the previous exercises.

## Create a Data Refinery Exercise 1: Transform CDR data and blend with geography data

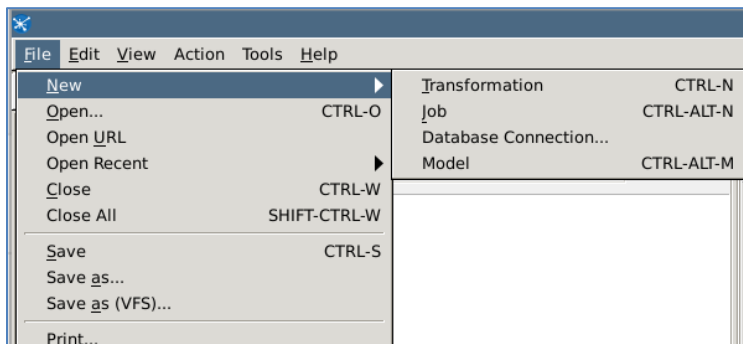
This exercise steps you through creating an advanced transformation to process and blend CDR data. You will process and blend file data and save it back to HDFS. The transform will include working with the **Stream Lookup** step to blend master geographic data. You will also add additional steps to enrich, filter, and sort the data as needed to complete the transformation. You will run this transformation within Spoon using the new functionality of Adaptive Execution Layer with Spark (AEL-Spark) offered in 7.1. Further information is at: <https://www.pentaho.com/product/version-7-1-update#adaptive-execution-with-spark>

You can follow the exercise step by step to create the transformation, or you can open the pre-built transformation and jump to step 66 to run the transformation. The pre-built transformation is at:

```
/pentaho/evaluation/02_Create_data_refinery/solutions/t_call_Vol_analysis_Spark.ktr
```

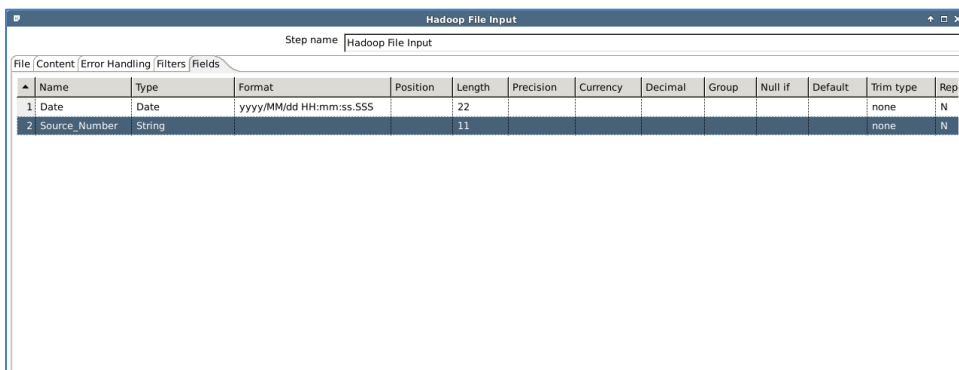
### Exercise Steps:

1. From the main Spoon menu choose **File | New | Transformation**.

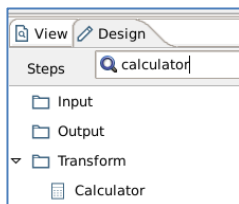


2. From the **Design** tab on the left, expand the **Big Data** folder; then, select and drag **Hadoop File Input** onto the canvas.
3. Double-click on the **Hadoop File Input** step to update its properties.
4. Under the **File** tab create one row with the following values:
  - a. **Environment:** CDH
  - b. **File:** /BDO/callrecords/input/callrecords\_10years.csv

- c. The source file does not include a header row, so on the **Content** tab make sure to leave the **Header** property unchecked.
- d. While the **Content** tab, also change the **Separator** from “;” to a “,”.
- e. At the bottom of the **Fields** tab, click **Get Fields**. When prompted for a **sample size**, type 0. Click **OK**.  
Note: You will notice that both fields are defined as “string.” You will need to change that.
- f. In the first row change the **Name** to Date, **Type** to Date, and **Format** to yyyy/MM/dd HH:mm:ss.SSS
- g. In the second row, change the **Name** to Source\_Number, **Type** to String, and **Length** to 11.
- h. The properties window should look like the screenshot below.

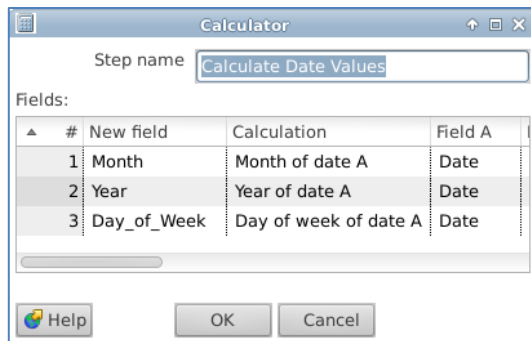


5. Click **OK** to return to the canvas.
6. Select and drag the **Calculator** step onto the canvas. In order to locate the **Calculator** step, you can search for it in the **Steps** search box.

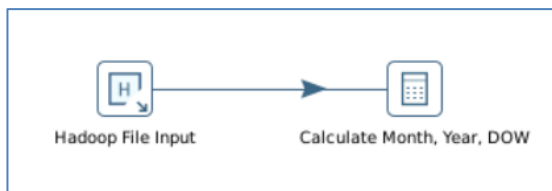


7. Double-click on the **Calculator** step to open its properties.
8. Change the **Step name** to Calculate Month, Year, DOW.

9. In the **Fields** section add Month, Year, and Day\_of\_Week as **New Fields**.
10. Under **Calculation** for Month select Month of date A.
11. Under **Calculation** for Year select Year of date A.
12. Under **Calculation** for Day\_of\_Week select Day of week of date A.
13. In **Field A**, Enter Date for each row. The properties for these fields should match the following screenshot:



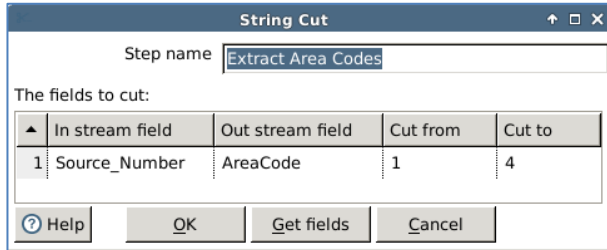
14. Click **OK** to return to the canvas.
15. Create a hop from the **Hadoop File Input** step to the **Calculate Month, Year, DOW** step.
16. The first part of your transformation should now match the following image:



Next, you'll need the location information. To derive location information from the data, you must know the area code within the phone number.

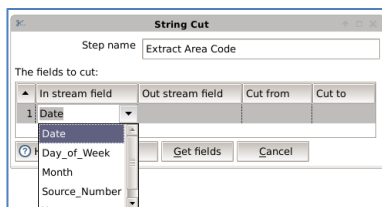
17. From the **Transform** folder select and drag the **Strings Cut** step onto the canvas.
18. Create a hop between the **Calculate Month, Year, DOW** step and the **Strings Cut** step.
19. Double-click on the **Strings Cut** step to open its properties.

20. In the **Step name** field, type `Extract Area Codes`. In the **Fields to cut** section select `Source_Number` as the **In stream field** and type `AreaCode` as the **Out stream field**. In the **Cut from** column enter 1 and in the **Cut to** column enter 4. The properties for these fields should match the following screenshot:



	In stream field	Out stream field	Cut from	Cut to
1	Source_Number	AreaCode	1	4

**Note:** To select the **In stream field**, you can click on the drop-down list and select it from the list of fields available. If no fields are available, click **Get fields** to get the list.

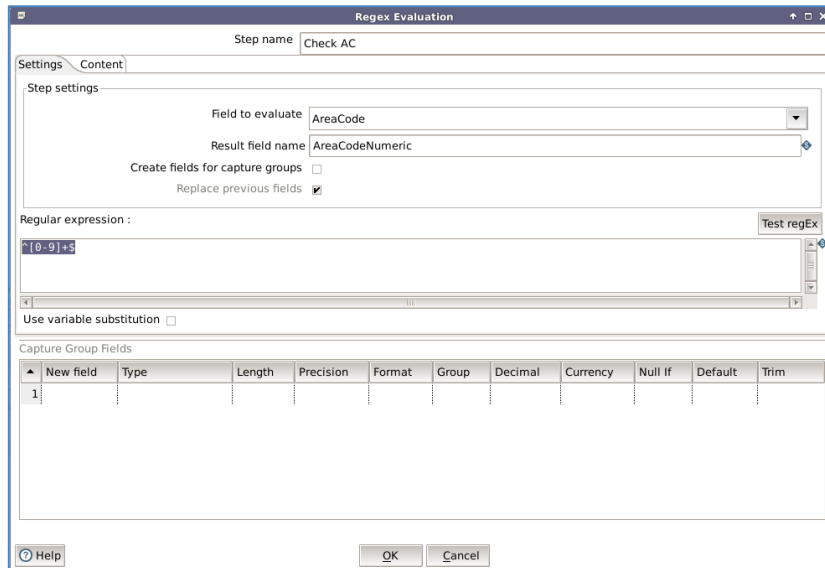


	In stream field	Out stream field	Cut from	Cut to
1	Date			

21. Click **OK** to return to the canvas.

Now that you know the area code, you will use a lookup file to map the area code to state, country, and time zone. Before you do that, let's verify the area codes are numeric and discard the records in which they are not.

22. From the **Scripting** folder, select and drag the **Regex Evaluation** step onto the canvas.
23. Create a hop from the **Extract Area Code** step to the **Regex Evaluation** step and select **Main output of step**.
24. Double click on the **Regex Evaluation** step.
25. Rename the step to `Check AC`. In the **Field to evaluate** property, select `AreaCode` from the drop-down list. The **Result** field should be named `AreaCodeNumeric`. In the **Regular Expression** window, type the following expression: `^[0-9]+$`
26. The properties of the **Check AC** step should look like this:



27. Click **OK** to return to the canvas.

28. From the **Flow** folder, select and drag the **Filter Rows** step onto the canvas.

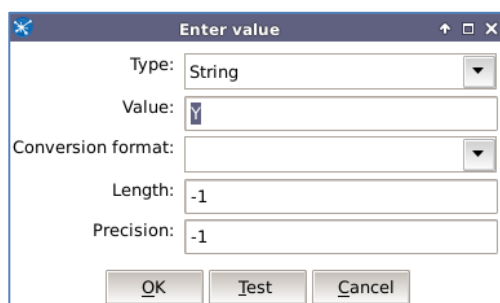
29. Create a hop between the **Check AC** step and **Filter rows** step and select **Main output of step**.

30. Double-click the **Filter rows** step to update its properties.

31. Rename this step to `Trash non-numeric`.

a. In **The condition** window, choose `AreaCodeNumeric` as the field.

b. Click on **Value**. It will bring up a pop-up window where you can set the value and the type. Change the **Type** to `String` and `Y` as the **Value**. The value properties should look like this:



c. Click **OK**.

d. The properties of the **Trash non-numeric** step will now look like this:

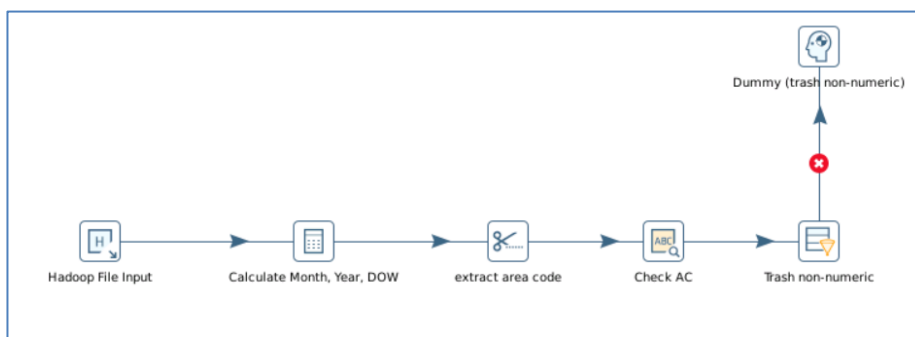
The screenshot shows the 'Filter rows' dialog box for a step named 'trash non-numeric'. It has fields for 'Send 'true' data to step:' and 'Send 'false' data to step:'. The 'The condition:' section shows a logical expression: 'AreaCodeNumeric' followed by an equals sign, a text input field containing 'Y', and '(Boolean)' in parentheses. At the bottom are 'Help', 'OK', and 'Cancel' buttons.

32. Click **OK** to return to the canvas.

33. Expand the **Flow** folder; then, select and drag the **Dummy (do nothing)** step onto the canvas, directly above the **Trash non-numeric** step. Double-click to open the properties of this step. Rename the step to **Dummy (trash non-numeric)**.

34. Create a hop between the **Trash non-numeric** step and the **Dummy (trash non-numeric)** step. When prompted, select the Result if FALSE option.

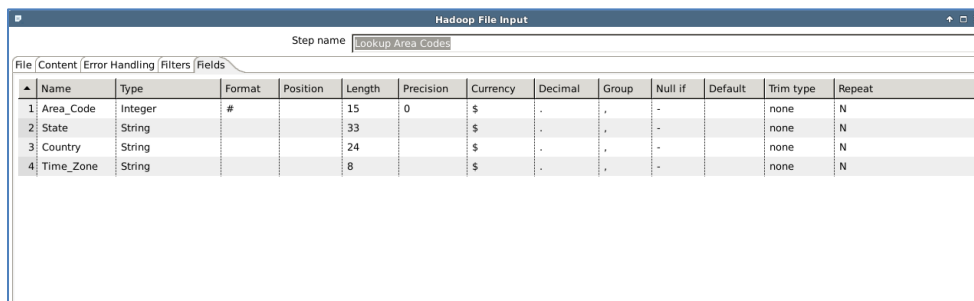
35. Your transform should look similar to this image now:



36. Expand the **Lookup** folder; then, select and drag **Stream Lookup** onto the canvas.

37. Expand the **Big Data** folder and select and drag the **Hadoop File Input** step onto the canvas directly above the **Stream Lookup** step.

38. Double-click on the **Hadoop File Input** step to open its properties. Update the properties as follows:
- Step Name:** Lookup Area Codes
  - Environment:** CDH
  - File:** /BDO/callrecords/reference/areacodes.csv
  - This file contains a header row, so make sure to check the box next to **Header** on the **Content** tab. On the **Content** tab also change the **Separator** to a "," and change **Format** to Unix.
  - Click the **Fields** tab and click on **Get Fields** to get the field names and data types. In the sample size, type 0.
  - The properties for this lookup file should resemble the image below:



	Name	Type	Format	Position	Length	Precision	Currency	Decimal	Group	Null if	Default	Trim type	Repeat
1	Area_Code	Integer	#		15	0	\$	.	-	-		none	N
2	State	String			33		\$	.	-	-		none	N
3	Country	String			24		\$	.	-	-		none	N
4	Time_Zone	String			8		\$	.	-	-		none	N

- Click **OK** to return to the canvas.
39. Create a hop from the **Trash non-numeric** step to the **Stream Lookup** step. When prompted, choose **Result is TRUE** to complete the hop connection.
40. Create a hop from the **Look Up Area Codes** step to the **Stream Lookup** step.
41. Double-click on the **Stream Lookup** step to open its properties.
- In the **Lookup step** select Lookup Areas Codes.
  - In the **Key(s) to Lookup Value(s)** section select AreaCode in the **Field** column and select Area\_Code as the **LookupField** column.
  - Click the **Get Lookup Fields** button (bottom right) to populate the **Fields to retrieve** section at the bottom.
  - Rename your **Stream Lookup** to **Lookup Country/State**. Highlight and delete Area\_Code from the fields to retrieve section. Your **Stream Lookup** dialog box should now look like this:



**Stream Value Lookup**

Step name:

Lookup step:

The key(s) to look up the value(s):

Field	LookupField
1 AreaCode	Area_Code

Specify the fields to retrieve :

Field	New name	Default	Type
1 State			String
2 Country			String
3 Time_Zone			String

e. Click **OK** to return to the canvas.

**!** Now let's convert the numeric values of day of week from the data set to the actual Day of Week values.

42. Expand the **Transform** folder and drag the **Value Mapper** step onto the canvas.
43. Create a hop between the **Lookup Country/State** step and the **Value Mapper** step.
44. Double-click the **Value Mapper** step. Rename it to **Day of Week**.
45. Update the properties of this step as follows:
  - a. Fieldname to use: `Day_of_Week`
  - b. Target Field name: `Weekday`
  - c. The **Field values** should look like this:

Field values:

	Source value	Target value
1	1	Sunday
2	2	Monday
3	3	Tuesday
4	4	Wednesday
5	5	Thursday
6	6	Friday
7	7	Saturday

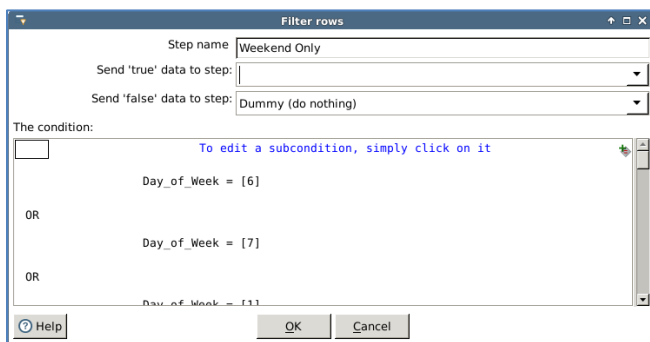
d. Click **OK** to return to the canvas



You need to apply a filter to the data to ensure you only get calls placed in the U.S. and calls made on weekends only, Saturday and Sunday. Calls placed outside of the U.S. and on days Monday through Friday will be discarded.

46. Expand the **Flow** folder; then, select and drag **Filter Rows** onto the canvas.
47. Create a hop between the **Day of Week** step and the **Filter Rows** step.
48. Double-click on the **Filter Rows** step to open its properties.
  - a. Rename this step to **US Calls Only**.
  - b. Click **OK** to return to the canvas.
49. From the **Flow** folder, select and drag the **Dummy (do nothing)** step onto the canvas above the **US Calls Only** step.
50. Create a hop from the **US Calls Only** step to the **Dummy (do nothing)** step. Select `Result Is False`.
51. Double-click the **US Calls Only** step to open its properties.
  - a. Under **The condition** section select the **<field>** on the left. From the pop-up window select `Country` and then click **OK**.
  - b. For the **<value>** field click and enter `UNITED STATES` as the value.
  - c. Click **OK** and then Click **OK** again to return to the canvas.
52. Add another **Filter Rows** step to the canvas. Double click and name it **Weekend Only**. Click **OK** to return to the canvas.
53. Create a hop between **US Calls Only** and **Weekend Only** and select **Result is TRUE** to complete the hop.
54. Create a hop between the **Weekend Only** and **Dummy (do nothing)** steps used above and select `Result is FALSE` to complete the hop. Double-click the **Weekend Only** step to open its properties.
  - a. Select the **<field>** on the left. From the pop-up window select `Day_of_Week` and then click **OK**.
  - b. Click the bottom right **<value>** box and type `6` as the **Value**.

- c. Click **OK**
- d. Click the **+** to the upper right of **The condition** section to add a condition.
- e. Click the **AND**, change it to **OR**, and click **OK**
- e. Click the **null = []**
- f. Select the **<field>** on the left. From the pop-up window select **Day\_of\_Week** and then click **OK**.
- g. Click the bottom right **<value>** box and type **7** as the **Value** in the pop-up window and click **OK**.
- h. Click the **UP** field to see all conditions.
- i. Repeat the steps d-i to add another condition where **Day\_of\_Week** is 1. Your step should look like the following:



- j. Click **OK** to return to the canvas.

**!** There are some area code values of 000. You will change those to null values.

55. Expand the **Utility** folder and drag the **Null if...** step onto the canvas.
56. Create a hop between the **Weekend Only** step and the **Null if...** step. Select **Result Is True** to complete the hop.
57. Double-click the **Null if...** step to open its properties.
  - a. Rename this step to **Replace Nulls**.
  - b. Under the **Fields** section select **AreaCode** for the **Name** field and type **000** in the **Value to turn to Null** field, then click **OK**.



To generate a call count measure, you will add a constant value of 1 to each record so you can aggregate the number of calls within analysis reports.

58. Expand the **Transform** folder; then, select and drag the **Add Constants** step onto the canvas.

59. Create a hop between the **Replace Nulls** step and the **Add Constants** step.

60. Double-click the **Add Constants** step to open its properties.

a. In the **Step Name** field, type `Add Call Count`.

b. Add a **Field** named `Calls` as a **Type** `Integer` with a **Value** of 1 as illustrated here:

Add constant values										
Step name: Add Call Count										
Fields :										
	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Value	Set empty string?
1	Calls	Integer							1	N

Buttons: ? Help, OK, Cancel

c. Click **OK**

61. From the **Big Data** folder, select and drag the **Hadoop File Output** step onto the canvas.

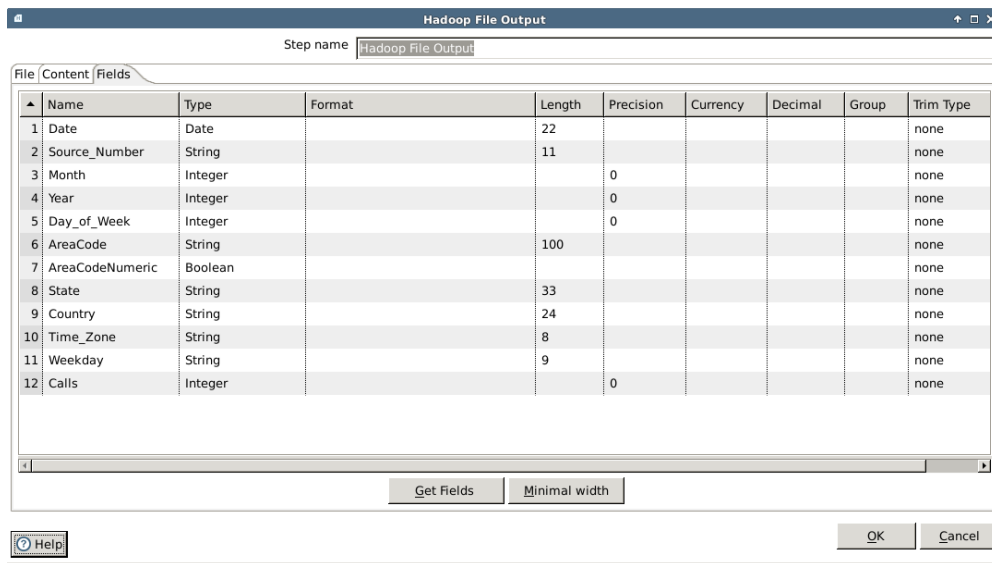
62. Create a hop between the **Add Call Count** step and the **Hadoop File Output** step.

63. Double-click the **Hadoop File Output** step to update its properties.

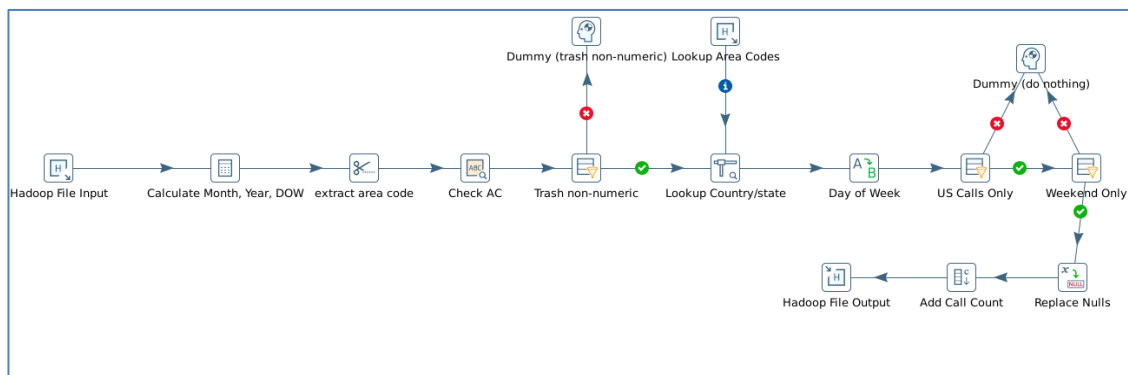
a. Under the **File** tab, select **CDH** for **Hadoop Cluster** and enter the following folder name for the **Folder/File** field: `/demo/data_refinery/callrecords_10years`

b. Be sure that the **Create Parent** folder is checked.

c. Under the **Fields** tab, click **Get Fields**. The fields should look like the following:



64. Your completed transformation should match the following image:



65. From the **File** menu, choose **Save As**. Save this transformation as `t_call_Vol_analysis_Spark` in the following directory:  
`file:///pentaho/evaluation/02_create_data_refinery/student_files`

66. Click the play icon  and run this transformation using Pentaho Local.

- From the Firefox bookmarks click **Hue – File Browser**. On the sign-in page, sign in with **demouser / demouser** if you are not already signed in.
- Navigate the output directory `/demo/data_refinery` to see the output file.

67. Click the play icon  and run this transformation using Spark.

- Select **Spark** from the run configuration drop-down list. Click **Run**.

68. Create another tab in the Firefox browser, click **Hue - Job Browser** to see that the job is accepted. You can monitor the job's progress from here or select the **Yarn Application** bookmark to see the job's progress.



The job may take some time to run. There is an initial amount of time to package any job and send it to Spark. Larger jobs will see a significant performance benefit from running on Spark. Small jobs like this one will take longer to run on Spark than if it were run on the local Pentaho server.

69. When the job is complete, click back to the browser tab **Hue - File Browser**.

70. Navigate to the following directory: `/demo/data_refinery`. Notice the directory with the name of the output file. Within the directory, you see that there are two files. Since the job was distributed, the output file was broken into more than one file and placed in a directory.

## Create a Data Refinery Exercise 2: Use Pentaho with Cloudera Impala to blend CDR and IoT data

This next exercise steps you through the process of creating a PDI job to execute a transformation for loading data to HDFS and creating an Impala table for querying with SQL.

You can follow the exercise step by step to create the transformation, or you can open the pre-built transformation and jump to step 48 to run the job. The pre-built job is at:

```
/pentaho/evaluation/02_Create_data_refinery/solutions/SDR_GeoLocation_Impala_Job_Ex2.kjb
```

### Exercise Steps:

1. From the main menu choose **File | New | Job**.
2. From the **File** menu, choose **Save**.
3. In the **Name** field, specify `SDR_GeoLocation_Impala_Job` and save to this directory:  
`file:///pentaho/evaluation/02_create_data_refinery/student_files`
4. From the **Design** tab on the left, expand the **General** folder and drag the following three steps onto the canvas: **START**, **Success**, and **Transformation**.



You need to create a directory on the HDFS to house the geolocation data file.

5. Expand the **File management** folder and drag the **Create a folder** step onto the canvas.
6. Create a hop between the **START** and **Create a folder** steps.
7. Double-click the **Create a folder** step to edit its properties. In the **Folder name** field type `hdfs://pentahobdvm.localdomain:8020/demo/data_refinery`
8. Uncheck the **Fail if folder exists** box.
9. Click **OK** to return to the canvas.



Next, you need to set HDFS permissions to allow Impala write privileges.

10. Expand the **Scripting** folder and drag the **Shell** step onto the canvas to the right of **Create a folder** and double-click to open.
11. In **Job entry name** enter **Set HDFS Directory Permissions**.

12. Check the box next to **Insert script**.

13. In **Working directory**, enter `/tmp`.

14. Switch to the **Script** tab and copy/paste the following:  
`hadoop fs -chmod -R 777 /demo/data_refinery`

15. Click **OK** to return to the canvas.

16. Create a hop between the **Create a folder** and **Set HDFS Directory Permissions** steps.

17. Create a hop between the Set HDFS Directory Permissions and Transformation steps.



Once the target directory is created and permissions for Impala access are granted, you need to create and load a text file with geolocation data to HDFS.

18. Double-click the **Transformation** step to edit its properties.

19. In the **Name of job entry** box, type `Load Geolocation Data`.

20. On the **Transformation Specification** tab click the browse icon for the **Transformation filename** field and browse to select the following file:  
`/pentaho/evaluation/02_create_data_refinery/solutions/SDR_GeoLocation_HDFS.ktr`



Note: To save time and reduce redundant work, you are using an *existing* transformation to load the geolocation data to HDFS.

21. Click **OK** to return to the canvas.



You will want to load data to an Impala table. But first you should validate that the target table exists within Impala. The table is named `call_detail_geo`.

22. Expand the **Conditions** folder and drag the **Table Exists** step onto the canvas.

23. Create a hop between the **Load Geolocation Data** and **Table Exists** steps.

24. Double-click the **Tables Exists** step to edit its properties.

25. In the **Job entry name** field, type `Does Impala table exist?`

26. In the **Connection** field select `Impala Apache`.



Note: This connection to Impala has already been established and is available for use.



27. In the **Table Name** field, type `call_detail_geo`.

28. Click **OK** to return to the canvas.



If the Impala table does exist, you will truncate the table data and load the geolocation data that was previously loaded to HDFS. To load the data, you will execute an SQL script.

29. Expand the **Scripting** folder and drag an **SQL** step to the right of the **Does Impala table exist?** step.

30. Rename this **SQL** step to `Load Impala Table`.



If the Impala table does not exist, you will create the table before loading the geolocation data using an SQL script.

31. Drag a second **SQL** step from the **Scripting** folder and place it below the **Does Impala table exist?** step.

32. Rename this **SQL** step to `Create Impala Table`.

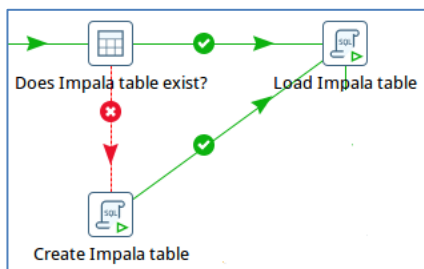
33. Create a hop from the **Does Impala table exist?** step to the **Load Impala table** step to the right. This hop should be green in color, indicating this path will be taken if the previous step returns a true evaluation.



Tip: To change the hop color, click the green arrow.

34. Create a hop from the **Does Impala table exist?** step to the **Create Impala table** step below. This hop should be red in color, indicating this path will be taken if the previous step returns a false evaluation.

35. Create a hop from the **Create Impala table** step to the **Load Impala table** step. The hops should match the following image:

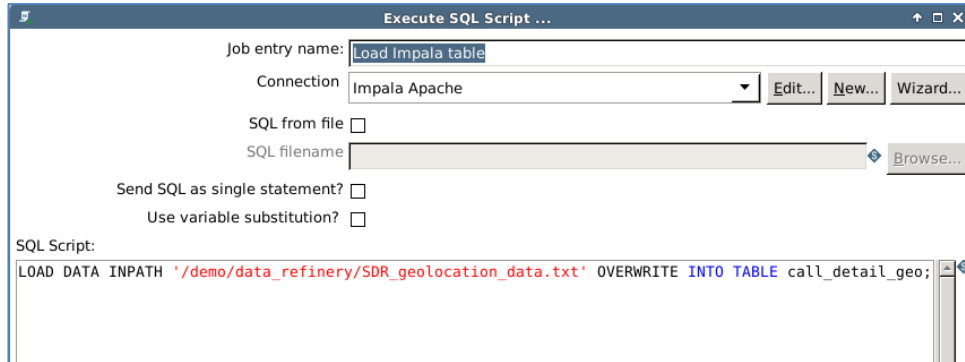


36. Double-click the **Load Impala table** step to edit its properties.

37. In the **Connection** field select `Impala Apache`.

38. In the **SQL Script** section type the following: `LOAD DATA INPATH '/demo/data_refinery/SDR_geolocation_data.txt' OVERWRITE INTO TABLE call_detail_geo;`

39. Your **SQL Script** step should now look like this:



40. Click **OK** to return to the canvas.

41. Double-click the **Create Impala table** step to edit its properties.

42. In the **Connection** field select Impala Apache.

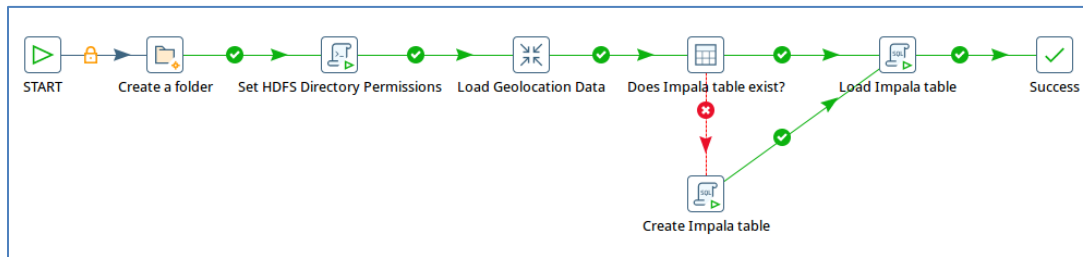
43. In the **SQL Script** section type or copy/paste the following SQL command:

```
CREATE TABLE call_detail_geo
(
    calldate VARCHAR(26),
    source_number VARCHAR(11),
    home_latitude DOUBLE,
    home_longitude DOUBLE,
    distance DOUBLE,
    direction VARCHAR(4),
    location_category VARCHAR(20),
    new_lat DOUBLE,
    new_long DOUBLE
)
row format delimited
fields terminated by '|'
STORED AS TEXTFILE;
```


44. Click **OK** to return to the canvas.

45. Create a hop from the **Load Impala table** step to the **Success** step.

46. Your PDI job should now look like this:



47. From the **File** menu, choose **Save**.

48. Execute the job by choosing **Action** → **Run** from the main menu or by clicking the run icon .

49. The **Execute a job** dialog box will appear. Click the **Launch** button at the bottom.

50. A green check will appear on the **Success** step when the job finishes without errors.

51. Keep this job open as it will be used in the next exercise.

## Create a Data Refinery Exercise 3: Extend the PDI job to blend data and load to multiple locations

Pentaho Data Integration (PDI) allows you to join data from multiple tables, transform it, and load it to multiple locations. You can take advantage of Impala queries to join two large tables into a combined data set. This combined set is loaded into a new combined Impala table and to a PostgreSQL database to enable high-performance OLAP analysis.

You can follow the exercise step by step to modify the job from the previous exercise, or you can open the pre-built job and jump to step 30 to run the job. The pre-built job is:

```
/pentaho/evaluation/02_Create_data_refinery/solutions/SDR_GeoLocation_Impala_Job_Ex3.kjb
```

### Exercise Steps:

1. Make sure the job, `SDR_GeoLocation_Impala_Job`, created in the previous exercise is open.
2. To save time you will use an existing job to load call detail records into Impala. From the **General** folder drag the **Job** step onto the canvas.

3. Drag the Job step between steps **Set HDFS Directory Permissions** and **Load Geolocation Data**.
4. Double-click the **Job** step to configure it.
  - a. Set **Entry Name** to Load Call Detail Records.
  - b. Navigate and select the following **Job**:
 

```
file:///pentaho/evaluation/02_Create_data_refinery/solutions/SDR_CallDetailRecords_Setup.kjb
```
  - c. Under the **Options** tab you can select **Local** since you will run this job locally.
  - d. Click **OK**.
5. Select the **Success** step and delete it.



Now you need to blend the geolocation data with the call data records and place the combined data set into a new Impala table using an SQL script.

6. From the **Design** tab on the left, expand the **Scripting** folder and drag the **SQL** step onto the canvas below the **Load Impala table** step.
7. Create a hop between the last step, the **Load Impala table** step, and the **SQL** step.
8. Double-click the **SQL** step to edit its properties. In the **Step name** field type Join CDR to Geo Location Data.
9. In the **Connection** field select Impala Apache.
10. In the **SQL** field type or copy/paste in the following SQL command:

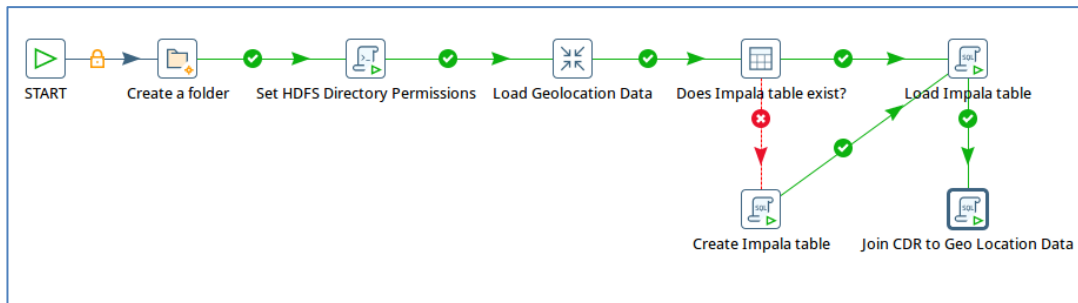
```
DROP TABLE IF EXISTS call_detail_combined;
CREATE TABLE call_detail_combined
(
  key INT
, source_number VARCHAR(11)
, call_date VARCHAR(26)
, call_month INT
, call_year INT
, day_of_week INT
, area_code INT
, state VARCHAR(11)
, country VARCHAR(13)
, time_zone VARCHAR(8)
, weekday VARCHAR(8)
, num_calls INT
, home_latitude DOUBLE
, home_longitude DOUBLE
, distance DOUBLE
, direction VARCHAR(4)
, location_category VARCHAR(20)
```

```

, new_lat DOUBLE
, new_long DOUBLE
);
INSERT INTO TABLE call_detail_combined
SELECT
    cdr.key
, cdr.source_number
, cdr.call_date
, cdr.call_month
, cdr.call_year
, cdr.day_of_week
, cdr.area_code
, cdr.state
, cdr.country
, cdr.time_zone
, cdr.weekday
, cdr.num_calls
, cdg.home_latitude
, cdg.home_longitude
, cdg.distance
, cdg.direction
, cdg.location_category
, cdg.new_lat
, cdg.new_long
FROM
    call_detail_records cdr JOIN call_detail_geo cdg ON
    (cdr.source_number = cdg.source_number);

```

11. Click **OK** to return to the canvas. Your job should match the following screenshot:



**!** To enable high-performance OLAP analysis and reporting, you will need to load the combined data set to a PostgreSQL database.

12. From the **Design** tab on the left, expand the **General** folder and drag the **Transformation** step onto the canvas to the right of the **Load Impala table** step.
13. Create a hop between the **Join CDR to Geo Location Data** step and the **Transformation** step.

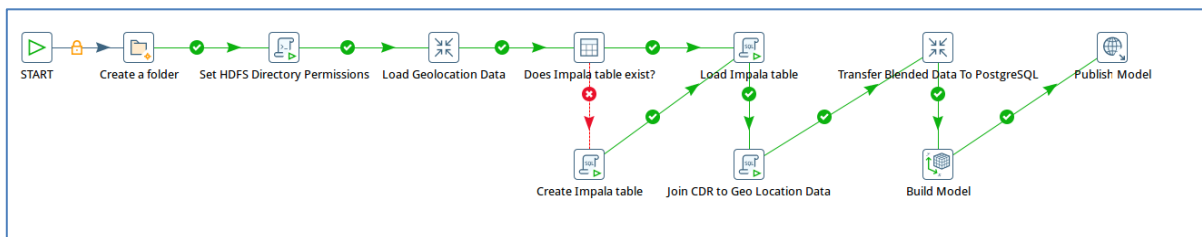
14. Double-click the **Transformation** step to edit its properties. In the **Entry Name** field type Transfer Blended Data to PostgreSQL.
15. On the **Transformation** field click browse to select the following file:  
file:///pentaho/evaluation/02\_create\_data\_refinery/solutions/SDR\_Geo  
CDR\_Transfer\_To\_postgres.ktr

Note: To save time and reduce redundant work, you are using an *existing* transformation to load the blended data to PostgreSQL.



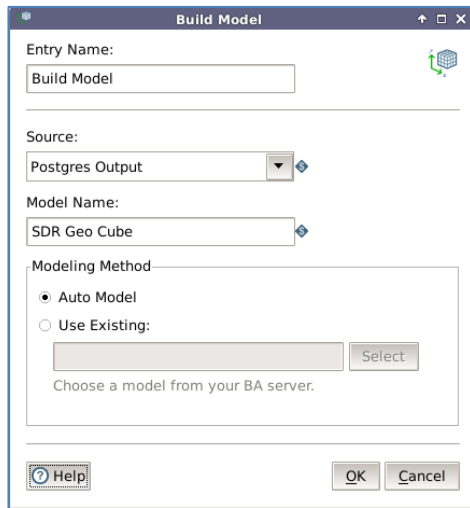
Now that the data is blended, you need to create and publish a model to the Pentaho Analytics server for web-based dimensional analysis and reporting.

16. From the **Design** tab on the left, expand the **Modeling** folder and drag the **Build Model** step onto the canvas below the **Transfer Blended Data to PostgreSQL** step.
17. Also from the **Modeling** folder drag the **Publish Model** step onto the canvas to the right of the **Transfer Blended Data to PostgreSQL** step.
18. Create a hop between the **Transfer Blended Data to PostgreSQL** step and the Build Model step.
19. Create a hop between the **Build Model** step and the **Publish Model** step to match the following screenshot:



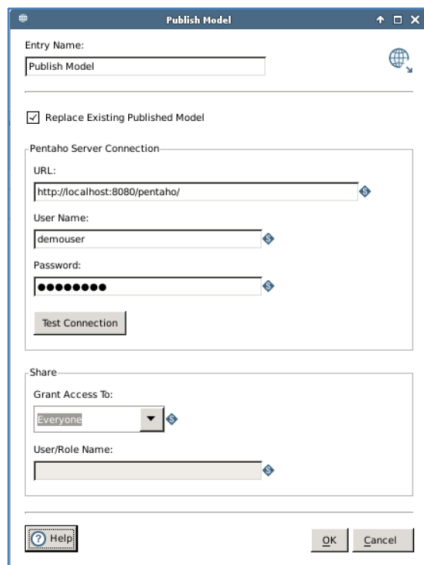
Note: The build model and publish model steps will automatically create and publish a metadata model to the analytics server for web-based dimensional analysis on the blended data created by this job.

20. Double-click the **Build Model** step to edit its properties. In the **Model Name** field type SDR Geo Cube and confirm that the **Source** is set to Postgres Output.



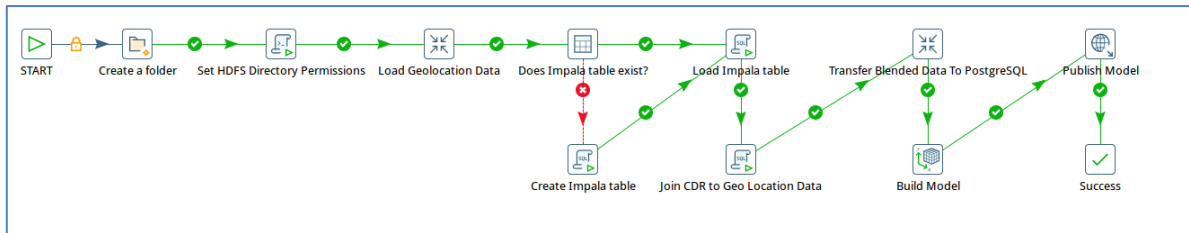
The 'Build Model' dialog box is shown. It has a title bar with standard window controls. The 'Entry Name' field is 'Build Model'. The 'Source' dropdown menu is set to 'Postgres Output'. The 'Model Name' field is 'SDR Geo Cube'. The 'Modeling Method' section has 'Auto Model' selected. There is a 'Select' button next to the 'Use Existing' option. At the bottom are 'Help', 'OK', and 'Cancel' buttons.


21. Double-click the **Publish Model** step to edit its properties.
22. Check **Replace Existing Published Model**.
23. For the **URL** enter: `http://localhost:8080/pentaho/`
24. Log in to the Pentaho User Console.
25. The remaining fields can be left as the default options.
26. Click **Test Connection**.



The 'Publish Model' dialog box is shown. It has a title bar with standard window controls. The 'Entry Name' field is 'Publish Model'. The 'Replace Existing Published Model' checkbox is checked. The 'Pentaho Server Connection' section has 'URL' set to 'http://localhost:8080/pentaho/', 'User Name' set to 'demouser', and 'Password' masked with dots. The 'Test Connection' button is visible. The 'Share' section has 'Grant Access To' set to 'Everyone' and 'User/Role Name' empty. At the bottom are 'Help', 'OK', and 'Cancel' buttons.

27. From the **Design** tab on the left, expand the **General** folder and drag the **Success** step onto the canvas below the **Publish Model** step.
28. Create a hop between the **Publish Model** step and the **Success** step to match the following screenshot:



29. From the **File** menu, choose **Save**.
30. Execute the job by choosing **Action** → **Run** from the main menu or by clicking the run icon .
31. The **Execute a job** dialog box will appear. Click the **Launch** button at the bottom.
32. A green check will appear on the **Success** step when the job finishes without errors.

Congratulations! You have completed the data integration work for implementing the streamlined data refinery (SDR) use case. A later section contains exercises for analyzing the data you loaded.



## Create a Data Refinery Exercise 4: Explore data in PostgreSQL with Pentaho Analyzer

Exercise 4 contains two parts showcasing how to use Pentaho Analyzer for analysis against a PostgreSQL database. Pentaho Analyzer offers an easy-to-use, graphical, drag-and-drop design environment that can be used by anyone who wants to dynamically explore data to discover anomalies or trends and create visualizations.

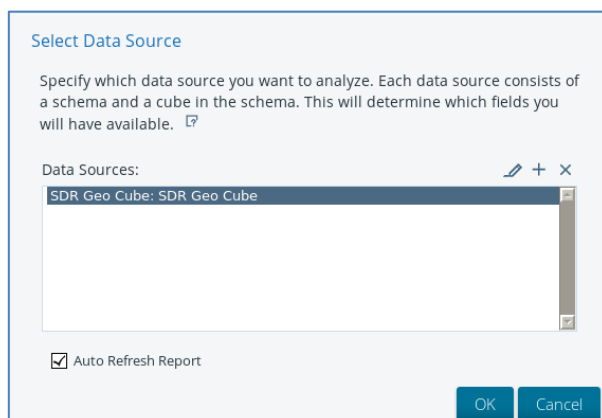
In this exercise, a telecom company is considering introducing a new VoIP service. You need to analyze the geolocation and calling data to determine customers' calling patterns to identify the best markets to launch a VoIP pilot service.

In Part 1 you will analyze data to determine where calls originate: from the house, the neighborhood, in town, or during travel. This information helps to determine which calling product has the most potential for this market. Later you will plot the calls on a map based on customers' source area codes to determine the highest-volume geographical markets to target for a new VoIP service.

### Part 1: Analyze data for a new VoIP pilot service

In this exercise you will create a table and column-line combo chart to aggregate call volume and average distance from home by location categories such as: at home, in the neighborhood, in town, during travel, etc.

1. Log in to Pentaho User Console, using "demouser" for both the user name and password.
2. Click on the **Create New | Analysis Report buttons**.
3. Select the SDR GEO CUBE data source from the **Data Sources** list.

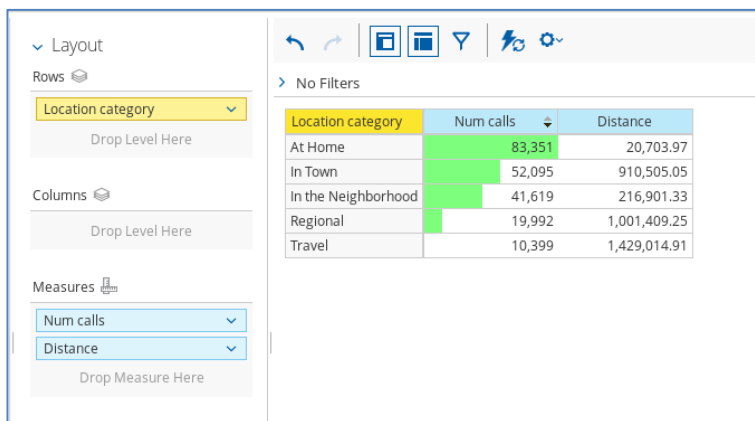


- This selection will launch you into a new **Analysis Report** view.
- From the **Available fields** section on the left, double-click or select and drag **Location Category**, **Num calls**, and **Distance** to the canvas.




Notice how the measures automatically aggregate the records in Analyzer.

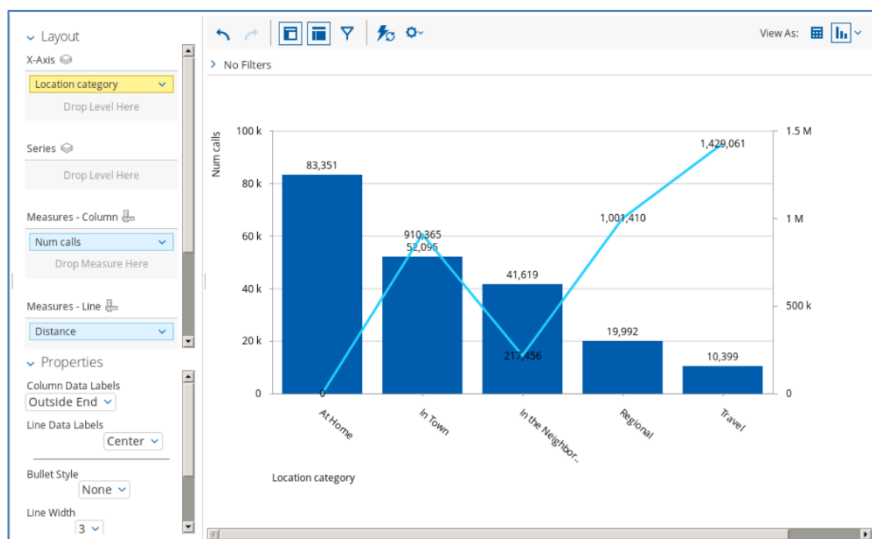
- Sort the **Num calls** field in descending order by right-clicking the **Num calls** column header and selecting **Sort Values High->Low**.
- Add conditional formatting to the **Num calls** column by right-clicking the **Num calls** column header and selecting **Conditional Formatting | Data Bar: Green**.



Note that most calls are made **At Home** or **In Town**, and that there is a drop in calls made **In the Neighborhood**. Given the high number of calls made at home, you've verified that a new VoIP calling service for homes may make sense.

- To visualize the data, click the chart drop-down list in the top right of your screen  and choose **Column-Line Combo**.
- In the **Layout** section, drag **Distance** from the **Measures – Column** drop zone down to the **Measures – Line** drop zone.

10. In the **Properties** section, change **Column Data Labels** to Outside End, **Line Data Labels** to Center, **Bullet Style** to None, and **Line Width** to 3. The resulting chart should match the following image:

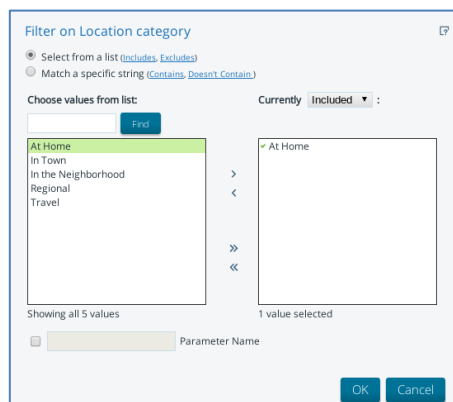


11. Click the **Save** icon to save the view as SDR Analyzer Exercise 1 in the default /home/demouser directory.

## Part 2: Analyze and map calling data by geography

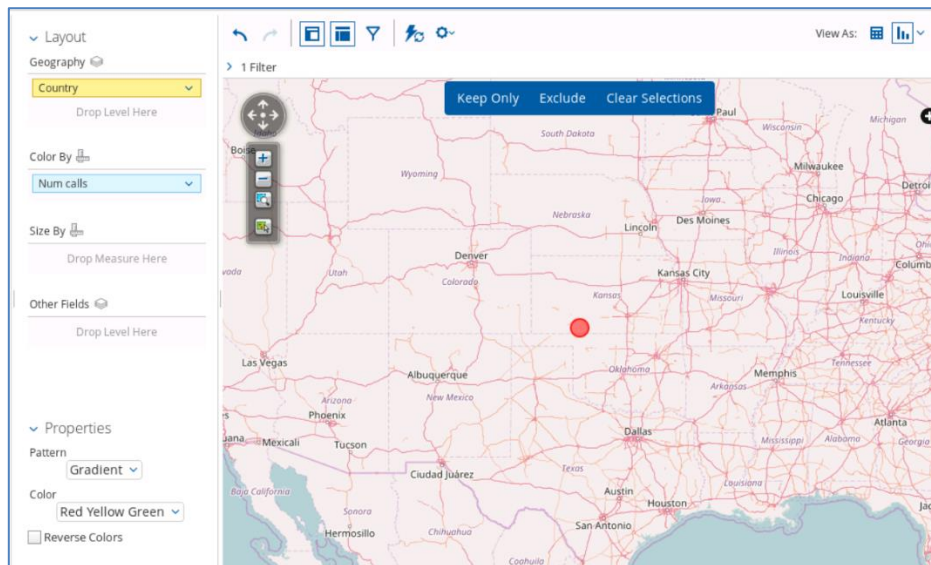
Now that you have verified that a VoIP calling plan makes sense, let's determine the geographic region where this service would make the most sense. In this exercise, you filter on calls made from home and plot call volume by geography in an interactive map.

12. Click on the **Create New | Analysis Report** buttons.
13. Select the SDR Geo Cube data source at the bottom of the **Data Sources** list.
14. In the **Available Fields** section, right-click on Location category and choose **Filter** to filter the view where Location Category equals At Home.

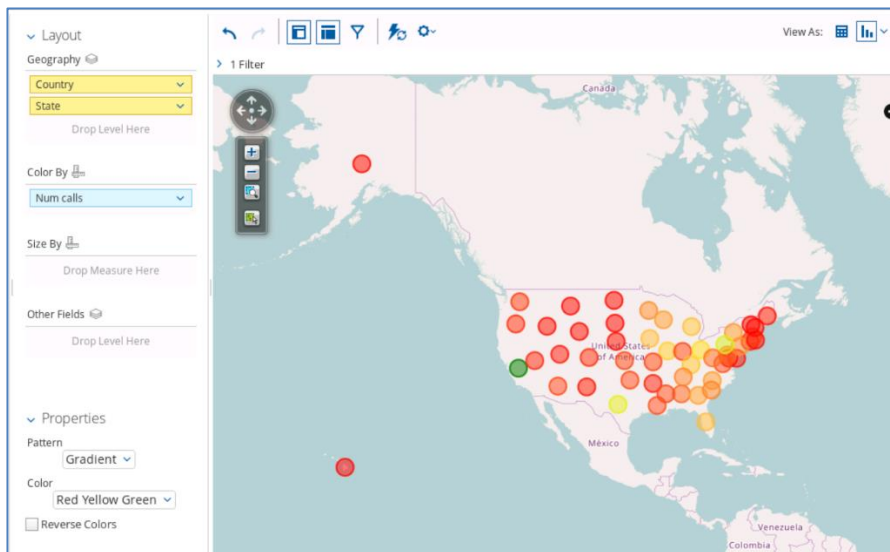


15. Add **Country** and the measure **Num calls** to the canvas.

16. Click the chart drop-down list and select **Geo Map**.



17. Double-click on the country circle and you will drill down to state.



**!** Note that California has the most calls at home, as denoted by the dark green circle.

You can see that California is the state to focus on. Congratulations! You have now completed the SDR use case.

# Use Case 3: Self-Service Data Preparation

## What is it?

Analytics users spend a significant amount of time preparing data for analysis or waiting for other people to prepare data for them. Self-service data preparation is a process for exploring, combining, cleaning, and transforming raw data into curated datasets for business intelligence and analytics.

## Why do it?

Existing data integration approaches are time-consuming and complex. Data analysts have a difficult time keeping up with the needs of the business. The growing volumes and variety of data are increasing the demand for easy-to-use data preparation tools. Companies have increased pressure to be responsive to the data needs of the business, and the need to quickly enrich and blend more data sources has become increasingly important. These challenges have made data preparation one of the primary blockers to delivering on the promise of analytics.

## Value of Pentaho

Pentaho's end-to-end platform offers several features that help organizations add value across their entire enterprise. These advantages include the following:

- Data agnostic – access to any data source
- Data exploration and profiling – a visual environment to explore and profile data
- Data transformation, blending, cleansing, filtering, modeling
- Collaboration – iterative, agile development environment with ability to publish and share models
- Data curation and governance – data encryption, security, and data lineage
- Integrated analytics and machine learning – use of analytics to improve data preparation

## What you will accomplish

In this use case, a marketing company offers drivers a service to wrap their cars with an advertisement to make extra money. An analyst for the marketing company needs to start targeting certain markets to attract new drivers and needs to determine which state and

cities to focus on first. To figure this out, you will need to combine call detail records (CDR) that the company has purchased and blend them with IoT data produced by cell towers.

By using these two data sources you can determine someone's cell phone usage and measure the distance they are from home. The further their distance from home, the more "exposure" that person provides for the advertisement and hence would be the best target market.

You will complete **one exercise** to create a transformation that merges data from two files and then categorizes the drivers into three buckets: low exposure, medium exposure, and large exposure. You'll correct the data along the way and create a geographical hierarchy to view the data on an interactive map.

## Self-Service Data Preparation Exercise 1: Use Pentaho to prepare data for analytics

1. In Spoon, create a new **Transformation**.
2. From the **Input** folder, select and drag the **CSV Input** step.



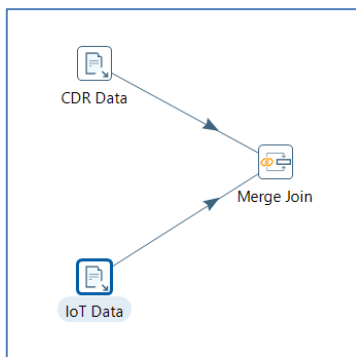
Note: You'll need two of these steps, so drag it to the canvas twice.

3. Double-click on the first **CSV Input** step to edit the step properties. Set the properties as follows:
  - a. Step Name: CDR Data
  - b. Filename: Browse to the following file to select it, and then click the **Open** button:
    - i. `file:///pentaho/evaluation/03_Self_service_data_prep/data/call_detail_records.csv`
  - c. Click **Get Fields** and enter "0" for the sample size to get the data types by scanning all rows of the file. Note that this scan may take a few seconds.
  - d. Click **Preview** to preview the data. In the sample size, type 500. You should see data coming back.
  - e. Click **OK**.

4. Now, let's update the properties of the second **CSV Input** step. Double-click it to update the properties.
  - a. Rename the step to **IoT Data**.
  - b. Filename: Browse to the following file to select it, and then click the **Open** button:  
`file:///pentaho/evaluation/03_Self_service_data_prep/data/call_detail_geolocation_output.csv`
  - c. Click **Get Fields** and enter "0" for the sample size to get the data types by scanning all rows of the file. Note that this scan may take a few seconds.
  - d. Click **Preview** to ensure that you can see the data.
  - e. Click **OK**.
5. Next, you will need to blend the data from these two sources. From the **Joins** folder, select and drag the **Merge Join** step onto the canvas.

**!** The **Merge Join** step allows you to blend data from disparate data sources based on one common object. The type of join will drive the result set. You only want data that appears in both data sets so you will choose **Inner Join** for your transform.

6. Create a hop between the **CDR Data** and **Merge Join** steps and select **Main output of step**.
7. Create a hop between the **IoT Data** and **Merge Join** steps and select **Main output of step**.
8. The result should look like the following image:



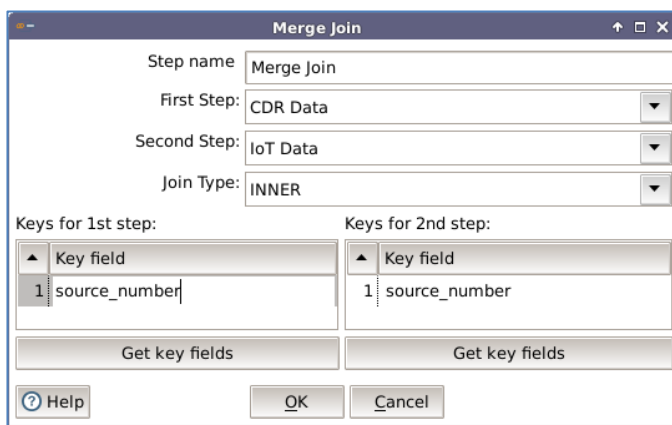
9. Double-click the **Merge Join** step to update its properties.

10. Set the properties as follows:

- a. **Step Name:** Merge Join
- b. **First Step:** CDR Data
- c. **Second Step:** IoT Data
- d. **Join Type:** Inner

11. Click on **Get key fields** for **1st step** and for **2nd step**.

12. You will blend data using `source_number` as the **key field**. Remove all other fields. Note that when you select a line and right click you can select **Keep only selected lines**. Your **Merge Join** properties should resemble the image below:



13. Click **OK** and then click **I Understand** in the subsequent prompt.



Sometimes, you need to add calculated fields to your data to provide additional analytics. You will create a new field called **Exposure** to categorize the data for analysis. The **Number Range** step in PDI allows you to categorize data based on thresholds.

14. From the **Transform** folder, select and drag the **Number Range** step onto the canvas. Double-click the step to update its properties.

15. Create a Hop between **Merge Join** and **Number Range**.

16. Rename the step to `Exposure Category`. Update the properties as follows:

- a. **Input Field:** distance
- b. **Output Field:** Exposure



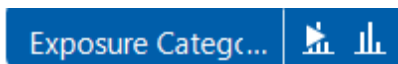
c. **Default Value:** Unknown

17. In the **Ranges** window, specify the following ranges:

▲	Lower Bound	Upper Bound	Value
1	0.0	10.0	Low Exposure
2	10.0	25.0	Medium Exposure
3	25.0		Large Exposure

Click **OK**.

18. Validate the data up to this point by clicking the **Data Explorer** icon that runs the transformation (the play icon).



19. You will be prompted to save the transformation. Save the transformation with the following name and location:

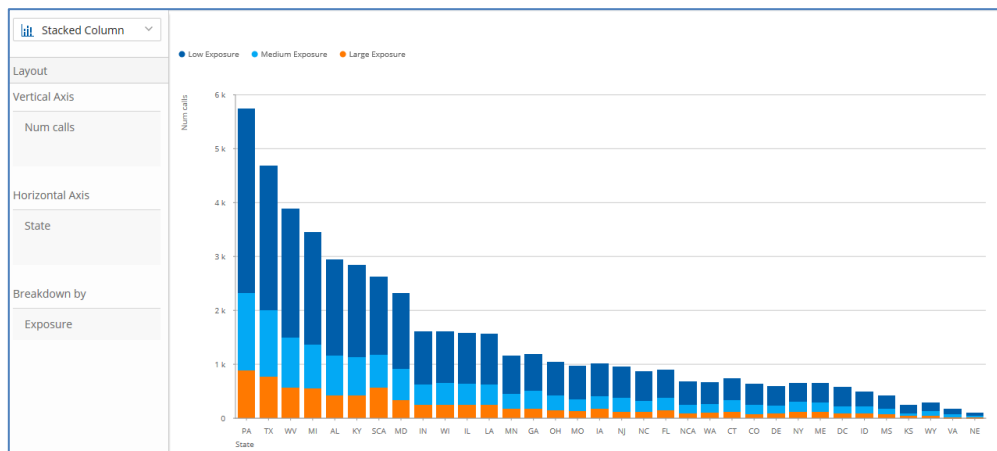
```
file:///pentaho/evaluation/03_Self_service_data_prep/studentfiles/SelfServiceDataPrep.ktr
```

20. Once saved you'll be taken to the Data Explorer interface. From the drop-down selector click on the **Stacked Column** visualization.

21. Add **State** to the **Horizontal Axis** and **Num calls** to the **Vertical Axis**.

22. Add **Exposure** to **Breakdown by**.

The graph should look like the following:





Note that California (CA) has been divided into Southern California (SCA) and Northern California (NCA) in the source data. You need to combine these to understand the exposure for California.

23. Return to the PDI canvas.

- From the **Transform** folder, select and drag the **Value Mapper** step onto the canvas.
- Insert the step between the **Merge Join** and the **Exposure Category** steps.
- Double-click the **Value Mapper** step to update its properties.
- Set the **Step Name:** Combine Northern and Southern CA
- Set **Fieldname to use** to `State`.
- Add a **Source value** of SCA with a **Target value** set to CA.
- Add another **Source value** of NCA with a **Target value** set to CA.
- The result should look like the following:

Value Mapper

Step name : Combine Northern and Southern California

Fieldname to use : state

Target field name (empty=overwrite) :

Default upon non-matching :

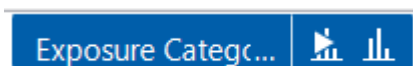
Field values:

#	Source value	Target value
1	NCA	CA
2	SCA	CA

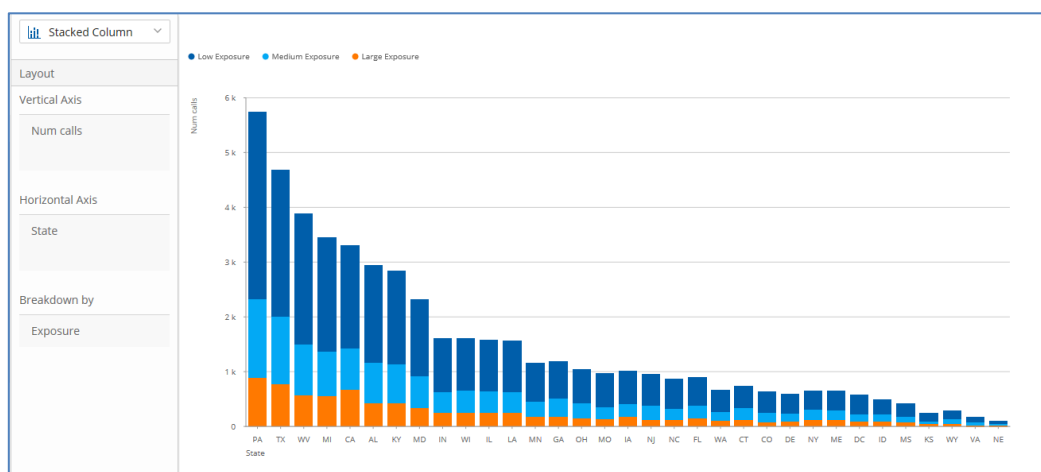
Buttons: Help, OK, Cancel

Click **OK**.

24. Click the play icon again.



25. You should now see only `CA` rather than separate values for the state.



26. Now you can filter out low and medium exposure and only look at the states with the largest distances travelled, providing the most exposure.

27. From the **Flow** folder, select and drag the **Filter rows** step onto the canvas.

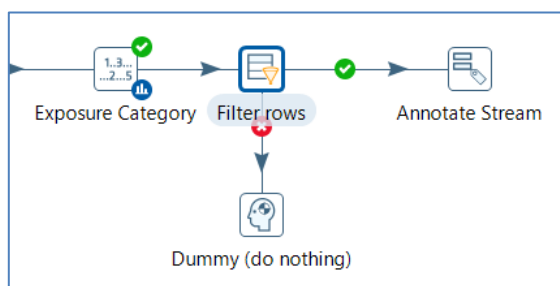
28. From the **Flow** folder, select and drag the **Annotate Stream** step onto the canvas.

29. From the **Flow** folder, select and drag the **Dummy (do nothing)** step onto the canvas.

30. Create a hop from **Exposure Category** to **Filter rows**.

31. Create a hop from **Filter Rows** to **Annotate Stream** and select **Result Is True**.

32. Create a hop from **Filter Rows** to **Dummy (do nothing)** and select **Result Is False**. The result should look like the following:



33. Double-click **Filter Rows** and enter the following:

- Set the condition where the <field> is `Exposure`, function is `Contains`, and the value is `Large`.
- The result should look like the following:

Filter rows

Step name:

Send 'true' data to step:

Send 'false' data to step:

The condition:

CONTAINS  (String)

34. Now you want create a geographical hierarchy to drill down to primary cities of each state.
35. Double-click the **Annotate Steam** step and change the **Step Name** to Add Geo Hierarchy.
  - a. Click **Select Fields** and select `country` from the **available fields** and move it to the **selected fields**. Click **OK**.
    - i. Double-click on **country** and select **Create Attribute** from the **Actions** drop-down list.
    - ii. For Geo Type select Country.
    - iii. For **Dimension** enter `Geo`.
    - iv. For **Hierarchy** enter `Geo`.
    - v. Click **OK**.
    - vi. The **Annotate Stream** step should now look like the following:

Step Name:  
Add Geo Hierarchy

☒ Local  
☐ Shared

Description:

Annotations:

Field	Model Action	Summary
country	Create Attribute	country is top level in hierarchy Geo

Add Calculated Measure... Select Fields...

Help Apply OK Cancel

- b. Click **Select Fields** and select `state` from the **available fields** and move it to the **selected fields**. Click **OK**.
  - i. Double-click on `state` and select **Create Attribute** from the **Actions** drop-down list.
  - ii. For **Geo Type** select **State**.
  - iii. For **Parent Attribute** select **Country**.
  - iv. For **Dimension** select **Geo**.
  - v. For **Hierarchy** select **Geo**.
  - vi. Click **OK**.
- c. Click **Select Fields** and select `primary_city` from the **available fields** and move it to the **selected fields**. Click **OK**.
  - i. Double-click on `primary_city` and select **Create Attribute** from the **Actions** drop-down list.
  - ii. For **Geo Type** select **City**.
  - iii. For **Parent Attribute** select **State**.

- iv. For **Dimension** select **Geo**.
  - v. For **Hierarchy** select **Geo**.
  - vi. Click **OK**.
- d. Click **Select Fields** and select `source_number` from the **available fields** and move it to the **selected fields**. Click **OK**.
- i. Double-click on `source_number` and select **Create Attribute** from the **Actions** drop-down list.
  - ii. For **Geo Type** select **Location**.
  - iii. For **Latitude** select **home\_latitude**.
  - iv. For **Longitude** select **home\_longitude**.
  - v. For **Parent Attribute** select **primary\_city**.
  - vi. For **Dimension** select **Geo**.
  - vii. For **Hierarchy** select **Geo**.
  - viii. Click **OK**.
  - ix. The **Annotate Stream** step should now look like the following:

Step Name:


☒ Local  
☐ Shared

Description:

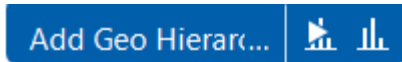
Annotations:

Field	Model Action	Summary
country	Create Attribute	country is top level in hierarchy Geo
state	Create Attribute	state participates in hierarchy Geo with parent country
primary_city	Create Attribute	primary_city participates in hierarchy Geo with parent state
source_number	Create Attribute	source_number participates in hierarchy Geo with parent primary_city

36. Click **OK**.

37. Click  to run the transformation and click **Yes** to save the transformation.

38. Click the play icon while the **Annotate Stream** step is selected on the canvas.

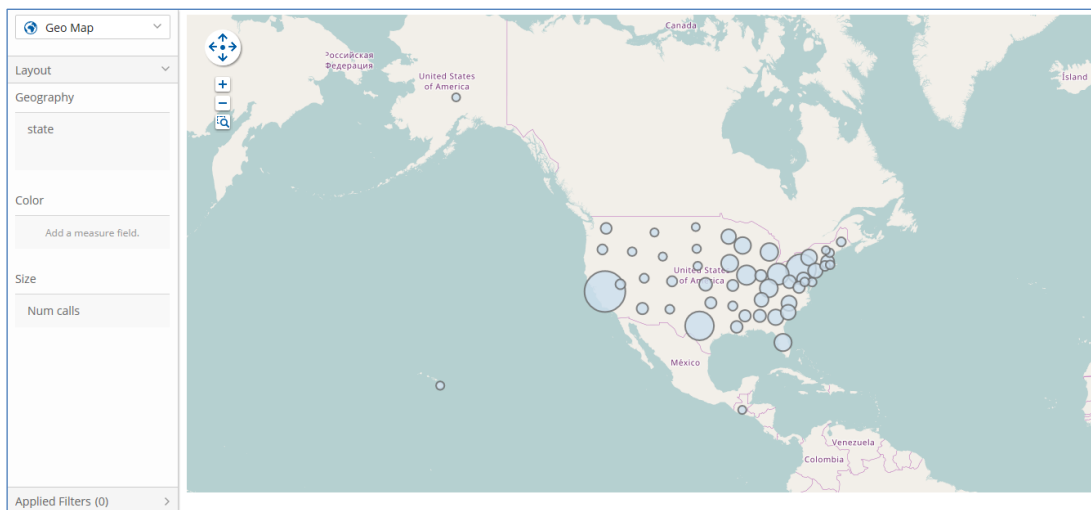


39. From within Data Explorer select **Geo Map**.

40. Scroll down the list of attributes to find the **Geo** hierarchy. Select `State` and drag it under **Geography**.

41. Scroll up to the **Measures** attribute and select `Num calls` and drag it under **Size**.

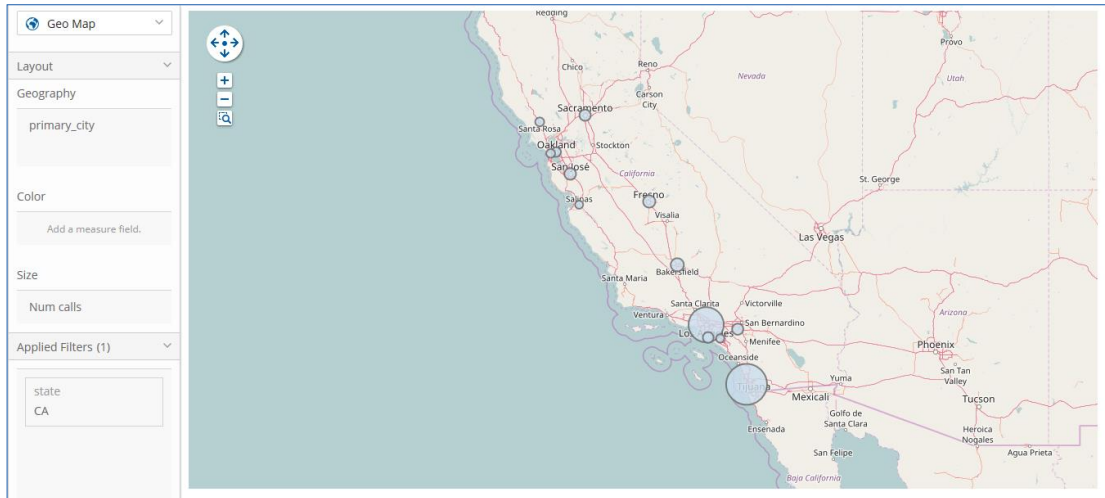
42. Your map should look like the following:



43. Click the **+** to zoom in.

44. You can see that California has the largest circle of all the states. Double-click on the circle over California to drill down to the cities.

45. Your map now looks like the following:



46. You can see the primary cities in California where you'll get the most exposure for your campaign. Hover over those circles and you can see the number of calls for each city.



# Use Case 4: Self-Service Analytics

## What is it?

Self-service analytics enables users to visualize business metrics quickly and easily to improve decision-making based on accurate, up-to-date, and governed data. Users need a reliable system for delivering consistent business metrics at the right time and in the right format. Self-service analytics includes reporting, ad hoc analysis, dashboards, and advanced visualization.

## Why do it?

- Make better decisions based on a comprehensive view of the business across the organization
- Empower business users with the information they need to make the best and most timely decisions
- To replace an existing BI solution that no longer serves customer needs or to make the information more reliable and consistent

## Value of Pentaho

Pentaho offers an end-to-end platform for data integration and business intelligence with the following key benefits.

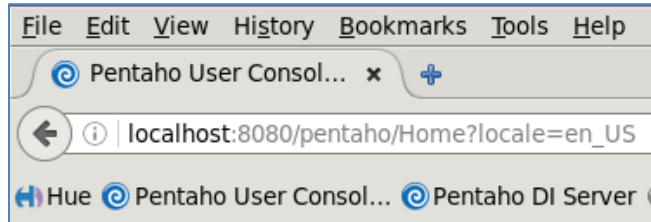
- Better together: Improve analytics and lower costs by combining business intelligence and data integration in a single platform
- Simplified analytics: Drag-and-drop interfaces for building analytic applications with the right data, in the right format, at the right time for better decision-making across your organization
- Time to value and low TCO: Pentaho provides a complete analytics offering that is easy to deploy to the broadest set of users (typically deploys in less than four weeks)

## What you will accomplish

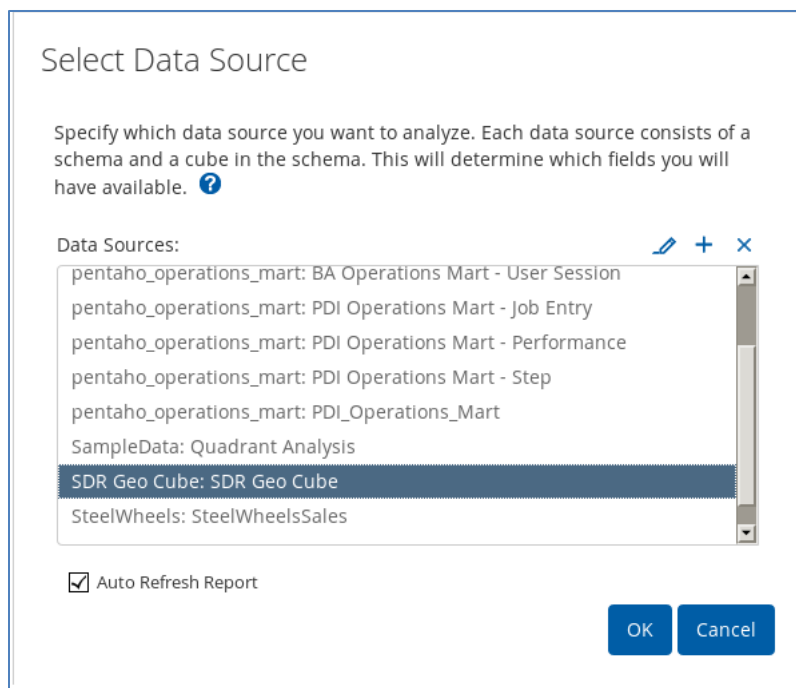
You will complete one exercise to perform interactive query and analysis on data from previous exercises.

## Self-Service Analytics Exercise 1: Use Pentaho to visualize data

1. From a web browser, connect to the Pentaho User Console by clicking the **Bookmarks** tab.



2. You'll need to login to the Pentaho User Console using demouser / demouser for the **User name** and **Password**.
3. From the Pentaho User Console select **Create New**. Select **Analysis Report**.
4. Select the **SDR Geo Cube: SDR Geo Cube** data source.



5. Create the following visualization by dragging **Country**, **State**, and the measure **Num calls** into the report.

The screenshot shows the 'Analysis Report' interface. On the left, the 'Available fields (20) for: SDR Geo Cube' are listed. The 'Layout' section shows 'Country' and 'State' in the Rows area, and 'Num calls' in the Measures area. The main table displays the following data:

Country	State	Num calls
	AK	85
	AL	3,665
	AR	916
	AZ	2,467
	CA	29,267
	CNMI	5
	CO	1,695
	CT	1,296
	DC	578
	DE	590
	FL	7,132
	GA	6,105
	GU	143
	HI	190
	IA	6,941
	ID	492
	IL	8,721
	IN	2,524
	KS	2,385
	KY	7,875
	LA	2,802

6. Then select the **Switch to Chart Format** pull-down menu.  Next select **Geo Map**.

The screenshot shows the 'View As' dropdown menu. The 'Geo Map' option is selected, indicated by a checkmark.

7. Save the visualization under the **/home/demouser** folder. Name this analysis **GEO\_MAP**.

### Save

Filename:

Location:

Name	Type	Date Modified
------	------	---------------

Save

Cancel

8. Create a new visualization using the same data source by clicking the icon with a **Plus Sign** at the top, selecting **New Analysis Report**, and then selecting the data source **SDR Geo Cube: SDR Geo Cube**.
9. You'll look at calls by area code. Add **State**, **Area code**, and the measure **Num calls** to the report as depicted below:

GEO\_MAP x Analyzer Report x

Available fields (22) for: SDR Geo Cube

Find: View

Location category

Measures

Area code

Call month

Call year

Day of week

Distance

Key

Num calls

Num calls

Source number

Source number

Time zone

Time zone

Weekday

Weekday

Layout

Rows

State

Area code

Drop Level Here

Columns

Drop Level Here

Measures

Num calls

Drop Measure Here

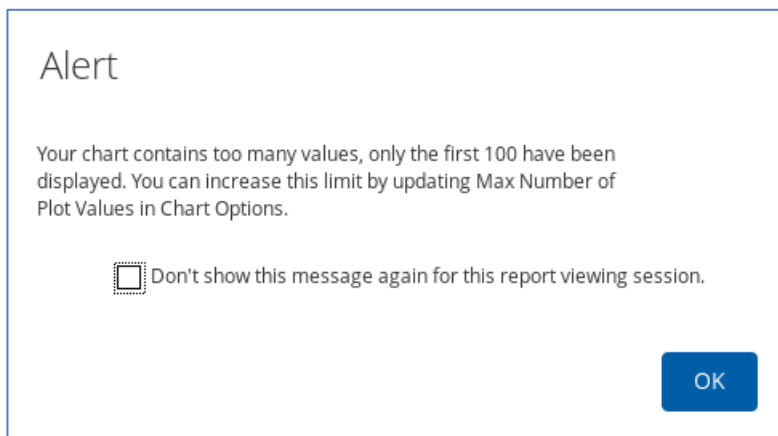
Properties

Report Options...

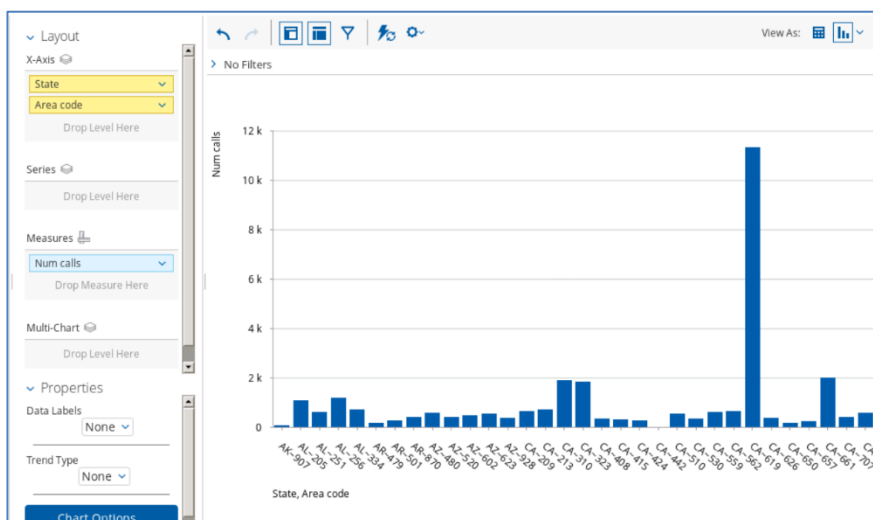
No Filters

State	Area code	Num calls
AK	907	85
	205	1,111
AL	251	623
	256	1,214
	334	717
AR	479	182
	501	294
	870	440
	480	592
	520	414
AZ	602	501
	623	558
	928	402
	209	676
	213	734
	310	1,902
	323	1,863
	408	375
	415	310
	424	285
	442	1
	510	549
	530	349

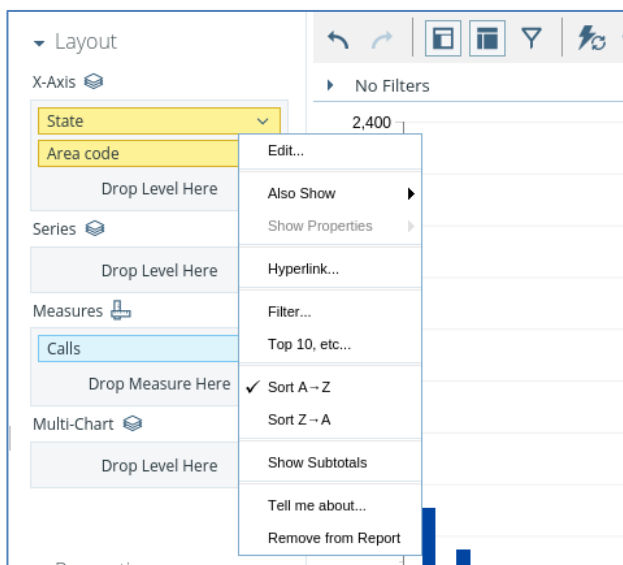
10. Change the **View As** to **Column**. Click **OK** if you see a warning about too many records.
11. You'll see an alert that the chart contains too many values and only the first 100 are displayed. Click the option to not display this alert again during this report viewing session and click **OK**



12. Your visualization should now look like this:

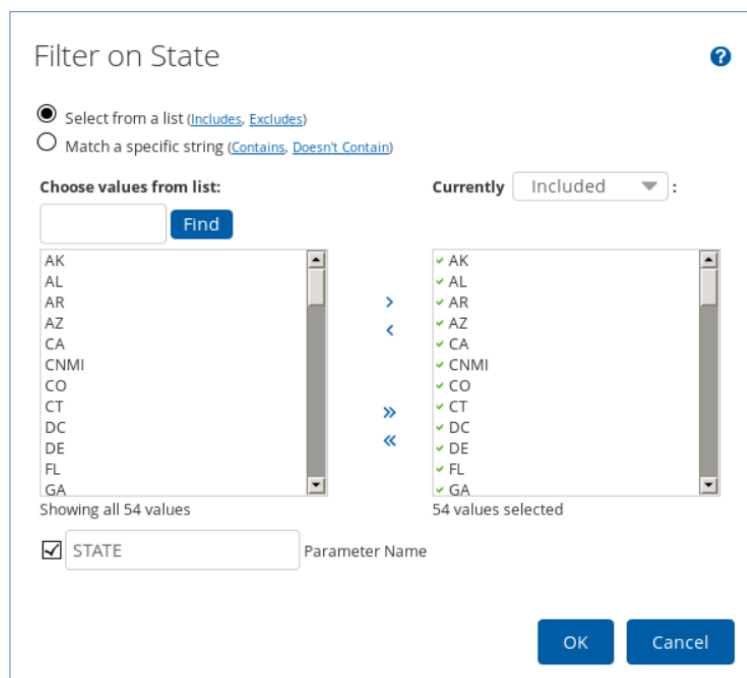


13. Right-click on **State** and add a filter on that field:



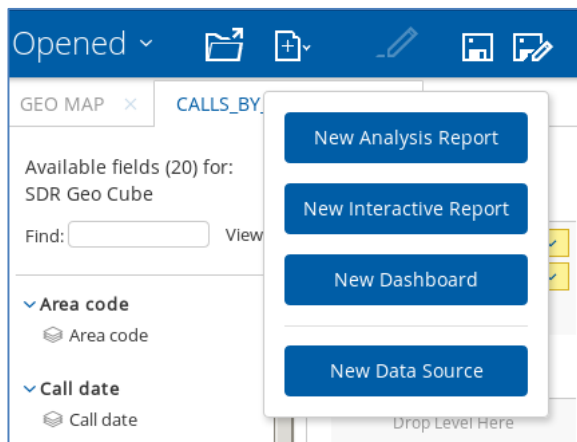
14. On the **Filter Options** select:

- Select from a list
- Include all states
- Add a parameter called STATE

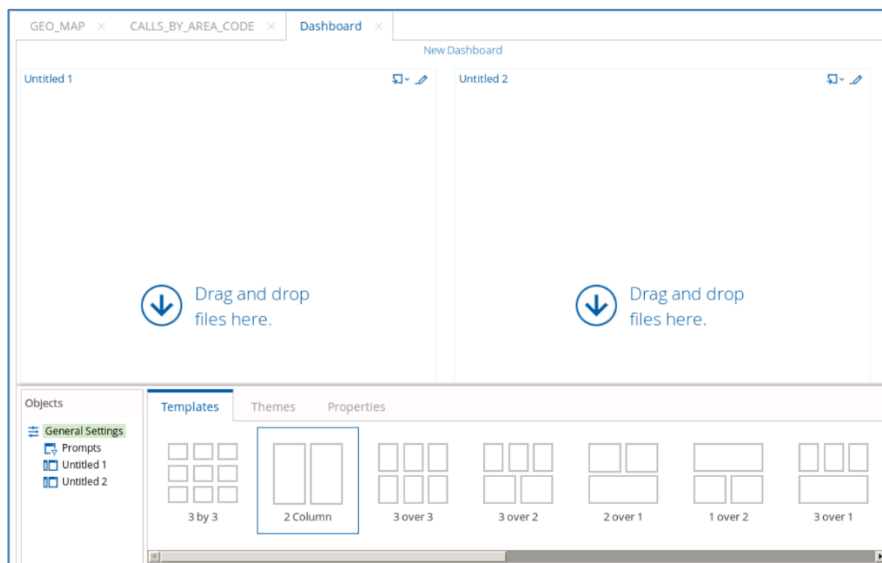


15. Save this analysis in **/home/demouser** and name this analysis **CALLS\_BY\_AREA\_CODE**.

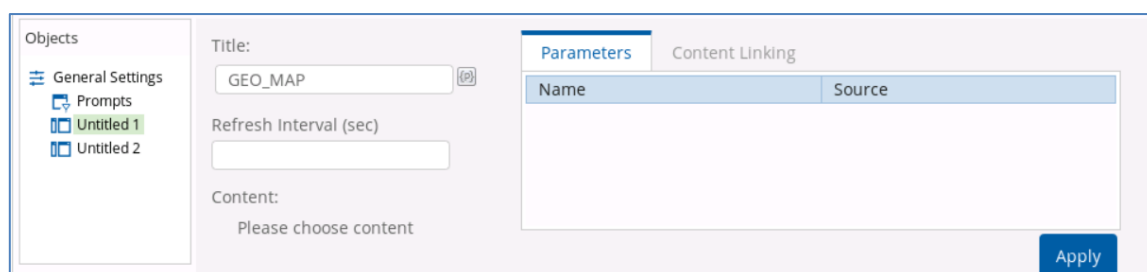
16. From the top tab, click on the icon with a **Plus Sign**. Select **New Dashboard**.



17. Select the 2-column template:



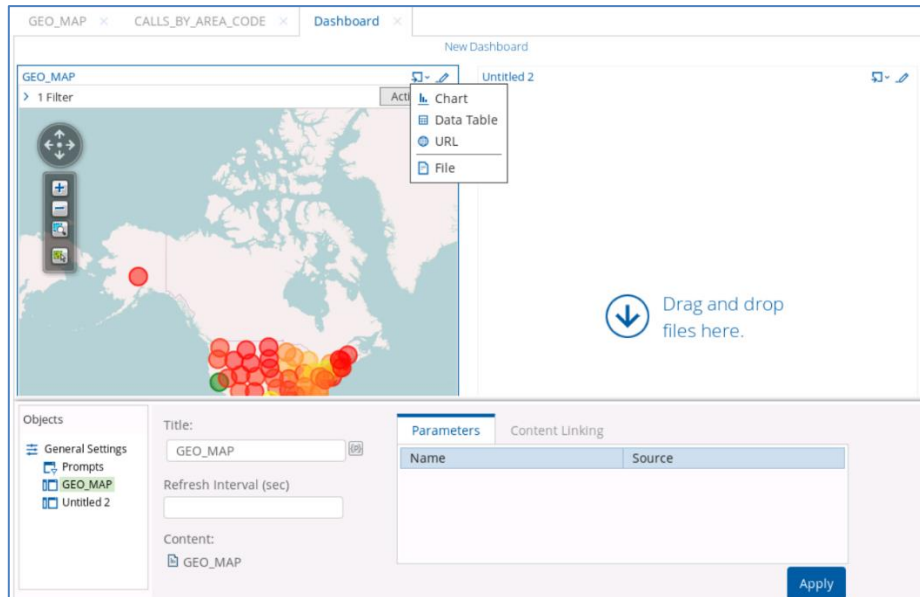
18. Add a title to the dashboard components. You can use the corresponding report names as titles.



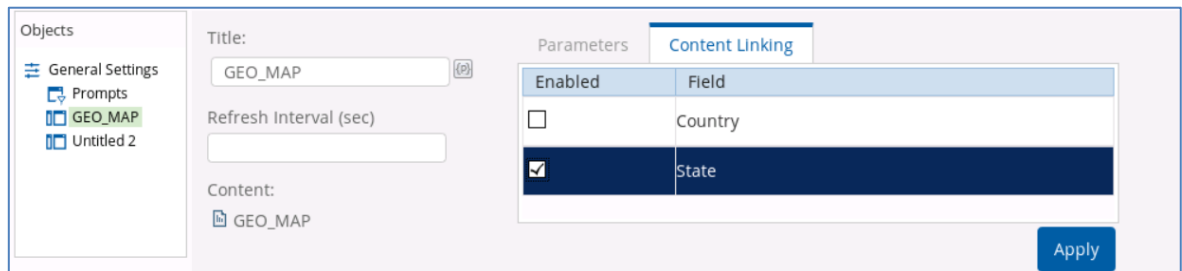
19. Click **Apply**.

20. From the left column of the dashboard select the icon to **Insert Content** and select **File**.

21. Navigate to **/home/demouser** and select GEO\_MAP and click **Select**.



22. Select the **Content Linking** tab and enable **State**:



23. Click **Apply**.

24. From the right column of the dashboard select the icon to **Insert Content** and select **File**.

25. Navigate to **/home/demouser** and select CALLS\_BY\_AREA\_CODE and click **Select**.



26. You'll see an alert that the chart contains too many values and only the first 100 are displayed. Click the option to not display this alert again during this report viewing session and click **OK**

27. Select the second column under objects named **Untitled 2**, and change the Title to `CALLS_BY_AREA_CODE`.

28. Click **Apply**

29. Select the `CALLS_BY_AREA_CODE` component and then on the State Parameter select **GEO\_MAP - State**.

Objects

- General Settings
- Prompts
- GEO\_MAP
- CALLS\_BY\_AREA\_CODE

Title: `CALLS_BY_AREA_CODE`

Refresh Interval (sec)

Content: `CALLS_BY_AREA_CODE`

Name	Source
STATE	GEO_MAP - State

Apply

30. On the **Content Linking** tab enable **State**:

Objects

- General Settings
- Prompts
- GEO\_MAP
- CALLS\_BY\_AREA\_CODE

Title: `CALLS_BY_AREA_CODE`


Refresh Interval (sec)

Content: `CALLS_BY_AREA_CODE`

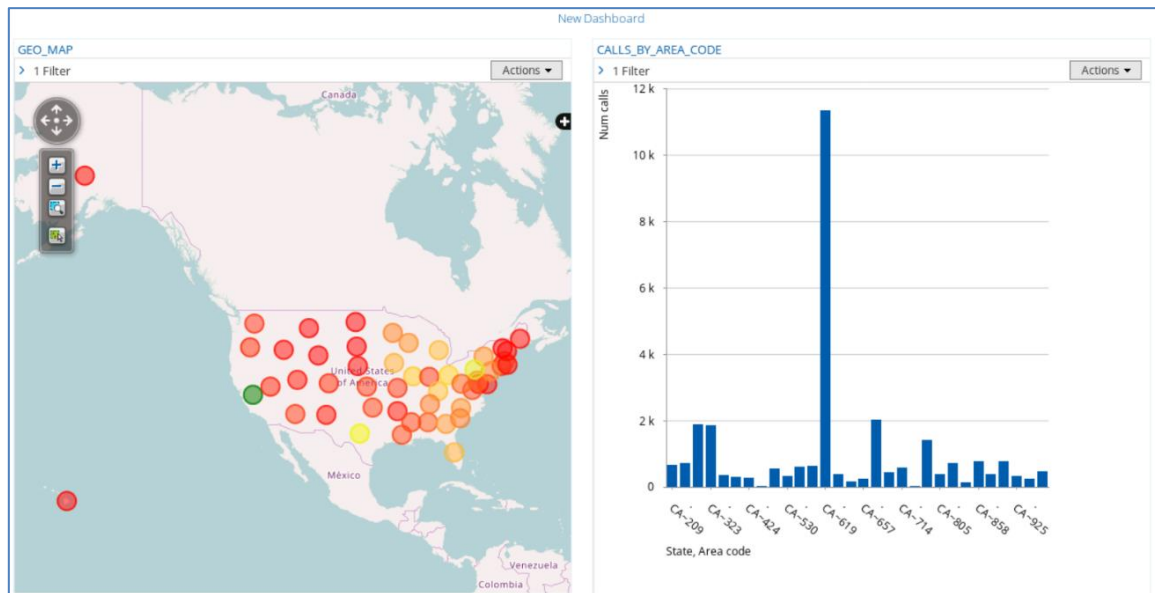
Enabled	Field
<input checked="" type="checkbox"/>	State
<input type="checkbox"/>	Area code

Apply

31. Click **Apply**

32. Save the dashboard and get out of the edit mode by hitting the pencil icon (  ).

33. Now, if you select California (double-click on the green circle in California), you should see the **Time Zone** component of the dashboard reflect only California area codes:



**Congratulations! You have now completed all use cases for the Big Data Sandbox.**