# A logo of a city Description automatically generated[https://avatars2.githubusercontent.com/u/4156894?v=3&s=100](http://www.calstatela.edu/centers/hipic) CIS5200 Term Project Tutorial

#### Authors: [Jae Hoon Lee](https://www.linkedin.com/in/jaehoonlee01/);  [Demetrios Daniel](https://www.linkedin.com/in/demetrius-daniel-a37422329/)

#### Instructor: [Jongwook Woo](https://www.linkedin.com/in/jongwook-woo-7081a85)

#### Date: 11/12/2024

**Lab Tutorial**

Jae Hoon Lee ([jlee464@calstatela.edu](mailto:jlee464@calstatela.edu))

11/12/2024

**Exploring Electric Vehicle Adoption Through Hadoop-Driven Urban Analytics**

**Objectives**

**List what your objectives are.** In this hands-on lab, you will learn how to:

* Remotely connect to a Hadoop Cluster and manage data uploads and downloads.
* Clean and preprocess the California EV registration dataset using Python to resolve line break issues in the Vehicle ID field.
* Create external Hive tables for EV datasets from various states.
* Perform SQL queries to analyze EV registration trends across states.
* Consolidate and summarize EV data into unified tables using SQL commands.
* Export summarized data from HDFS to a local environment.
* Visualize EV data trends using Tableau, including creating heatmaps, bar charts, and dashboards.

**Platform Spec**

* IBM Bluemix BigInsights
* CPU Speed: ?
* # of CPU cores: ?
* # of nodes: ?
* Total Memory Size: ?

**Step #1: Remotely connect to Hadoop Cluster and download files**

1. Open 3 Gitbash

1. Upload and download for your desktop.
2. Bash
3. Beeline

ssh **jlee464**@129.146.230.230 (change username)

Use beeline

-bash-4.2$ beeline

Make sure use your own database.

-bash-4.2$ use **jlee464**;

2. Download files from the below links

<https://www.energy.ca.gov/filebrowser/download/6539?fid=6539#block-symsoft-page-title>

<https://geodata.bts.gov/datasets/usdot::alternative-fueling-stations/about>

<https://afdc.energy.gov/vehicle-registration>

<https://www.atlasevhub.com/materials/state-ev-registration-data/>

(download all States’ files)

A screenshot of a computer

Description automatically generated

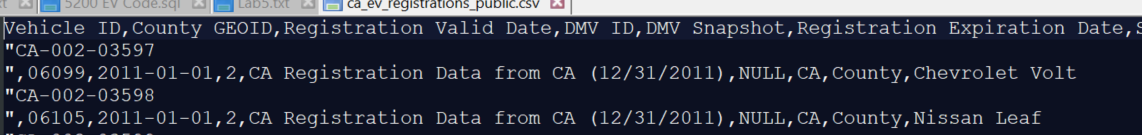
**Step #2: Create 5 ev folders for external table due to data format.**

1. California data cleaning with Python

California EV Dataset Issue

When loading the California EV dataset in Beeline, the Vehicle ID field caused issues due to unintentional line breaks, leading to row splitting and NULL values.

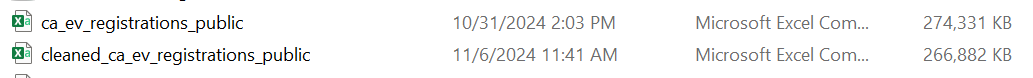
This Python script removes line breaks within Vehicle ID to ensure each record stays on a single line for correct parsing in Hive.



|  |
| --- |
| #csv file cleasning with python.  import csv  import re  # Define file paths  input\_file = r'F:\CSULA\CIS 5200\Project\EV\ca\_ev\_registrations\_public.csv'  output\_file = r'F:\CSULA\CIS 5200\Project\EV\cleaned\_ca\_ev\_registrations\_public.csv'  # Open input and output files  with open(input\_file, 'r', encoding='utf-8') as infile, open(output\_file, 'w', newline='', encoding='utf-8') as outfile:  # Initialize CSV writer  writer = csv.writer(outfile)    # Buffer to accumulate lines for each record  buffer = []    for line in infile:  # If line starts with "CA-", it is a new record, so process the buffer  if re.match(r'^"CA-\d{3}-\d{5}', line):  if buffer:  # Write the previous buffered line as a single record  writer.writerow(csv.reader([''.join(buffer)]).\_\_next\_\_())  # Start new buffer  buffer = [line.strip()]  else:  # Continuation of the previous line; add to buffer  buffer.append(line.strip())    # Write the last buffered record  if buffer:  writer.writerow(csv.reader([''.join(buffer)]).\_\_next\_\_())  print(f"Cleaned file saved to {output\_file}") |

A black background with yellow and green text

Description automatically generated



|  |
| --- |
| DROP TABLE IF EXISTS ev1;  CREATE EXTERNAL TABLE IF NOT EXISTS ev1 (  vehicle\_id STRING,  county\_geoid STRING,  registration\_valid\_date STRING,  dmv\_id INT,  dmv\_snapshot STRING,  registration\_expiration\_date STRING,  state\_abbreviation STRING,  geography STRING,  vehicle\_name STRING  )  ROW FORMAT DELIMITED  FIELDS TERMINATED BY ','  STORED AS TEXTFILE  LOCATION '/user/jlee464/ev1'  TBLPROPERTIES ("skip.header.line.count"="1"); |

2. fl External Table

|  |
| --- |
| CREATE EXTERNAL TABLE IF NOT EXISTS fl (  dmv\_id INT,  dmv\_snapshot STRING,  county string,  vehicle\_name STRING,  vin\_model\_year STRING,  registration\_valid\_date STRING,  registration\_expiration\_date STRING,  technology string  )  ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'  WITH SERDEPROPERTIES (  "separatorChar" = ",",  "quoteChar" = "\""  )  STORED AS TEXTFILE  LOCATION '/user/jlee464/fl'  TBLPROPERTIES ("skip.header.line.count"="1"); |

3. wa External Table

|  |
| --- |
| Drop table if exists wa;  CREATE EXTERNAL TABLE IF NOT EXISTS wa (  dmv\_id INT,  dmv\_snapshot STRING,  zipcode STRING,  state STRING,  registration\_valid\_date STRING,  dmv\_is\_complete STRING,  registration\_expiration\_date STRING,  vin\_prefix STRING,  vin\_model\_year STRING,  vehicle\_name STRING,  technology STRING  )  ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'  WITH SERDEPROPERTIES (  "separatorChar" = ",",  "quoteChar" = "\""  )  STORED AS TEXTFILE  LOCATION '/user/jlee464/wa'  TBLPROPERTIES ("skip.header.line.count"="1"); |

4. wi EXTERNAL TABLE

|  |
| --- |
| 4. wi EXTERNAL TABLE  drop table if exitsts wi;  CREATE EXTERNAL TABLE IF NOT EXISTS wi (  dmv\_id INT,  dmv\_snapshot STRING,  zipcode STRING,  vin\_prefix STRING,  vin\_model\_year STRING,  registration\_valid\_date STRING,  registration\_expiration\_date STRING,  make STRING,  model STRING,  model\_year STRING  )  ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'  WITH SERDEPROPERTIES (  "separatorChar" = ",",  "quoteChar" = "\""  )  STORED AS TEXTFILE  LOCATION '/user/jlee464/wi'  TBLPROPERTIES ("skip.header.line.count"="1"); |

5. All other State External Table

|  |
| --- |
| 5. All other State External Table  CREATE EXTERNAL TABLE IF NOT EXISTS ev3 (  state STRING,  county STRING,  registration\_date STRING,  vehicle\_make STRING,  vehicle\_model STRING,  vehicle\_model\_year INT,  drivetrain\_type STRING,  vehicle\_gvwr\_class STRING,  vehicle\_gvwr\_category STRING,  vehicle\_count INT,  dmv\_snapshot\_id INT,  dmv\_snapshot STRING,  latest\_dmv\_snapshot\_flag BOOLEAN  )  ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'  WITH SERDEPROPERTIES (  "separatorChar" = ",",  "quoteChar" = "\""  )  STORED AS TEXTFILE  LOCATION '/user/jlee464/ev3'  TBLPROPERTIES ("skip.header.line.count"="1"); |

|  |
| --- |
| Show Tables; |

+--------------------+

| tab\_name |

+--------------------+

| ev1 |

| ev3 |

| fl |

| wa |

| wi |

+--------------------+

**Step #3: Prepare data upload to Linux server**

create ev2 folder in your local computer



move file into ev2 folder with wi\_ev\_registrations\_public, wa\_ev\_registrations\_public and fl\_ev\_registrations\_public.

A screenshot of a computer

Description automatically generated

create ev3 folder in your local computer

move file in to ev3 folder with all register files except ca, wi, wa, fl data files.

A screenshot of a computer

Description automatically generated

Create a zip file with ev3 folder



scp "F:\CSULA\CIS 5200\Project\EV\cleaned\_ca\_ev\_registrations\_public.csv" jlee464@129.146.230.230:/home/jlee464/

scp "F:\CSULA\CIS 5200\Project\EV\ev2\fl\_ev\_registrations\_public.csv" jlee464@129.146.230.230:/home/jlee464/

scp "F:\CSULA\CIS 5200\Project\EV\ev2\wa\_ev\_registrations\_public.csv" jlee464@129.146.230.230:/home/jlee464/

scp "F:\CSULA\CIS 5200\Project\EV\ev2\wi\_ev\_registrations\_public.csv" jlee464@129.146.230.230:/home/jlee464/

scp "F:\CSULA\CIS 5200\Project\EV\ev3.zip" jlee464@129.146.230.230:/home/jlee464/

-bash-4.2$ ls

A screen shot of a computer code

Description automatically generated

Create folders and move files

-bash-4.2$ mkdir ev1

-bash-4.2$ mv cleaned\_ca\_ev\_registrations\_public.csv/

-bash-4.2$ unzip ev3.zip

**Step #4: Create HDSF folders and put files**

-bash-4.2$ hdfs dfs -mkdir ev1

-bash-4.2$ hdfs dfs -mkdir fl

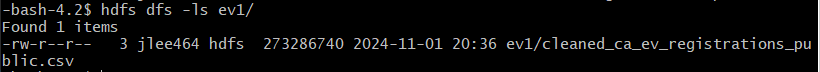
-bash-4.2$ hdfs dfs -mkdir wa

-bash-4.2$ hdfs dfs -mkdir wi

-bash-4.2$ hdfs dfs -mkdir ev3

-bash-4.2$ hdfs dfs -put ev1/\*.\* ev1/

-bash-4.2$ hdfs dfs -ls ev1/



-bash-4.2$ hdfs dfs -put fl\_ev\_registrations\_public.csv fl/

-bash-4.2$ hdfs dfs -ls fl/



-bash-4.2$ hdfs dfs -put wa\_ev\_registrations\_public.csv wa/

-bash-4.2$ hdfs dfs -ls wa/



-bash-4.2$ hdfs dfs -put wi\_ev\_registrations\_public.csv wi/

-bash-4.2$ hdfs dfs -ls wi/



-bash-4.2$ hdfs dfs -put ev3/\*.\* ev3/

-bash-4.2$ hdfs dfs -ls ev3/

A screen shot of a computer screen

Description automatically generated

**Step #5: test external tables**

select \* from ev1 limit 5;

A black screen with white text

Description automatically generated

select \* from fl limit 5;

select \* from wa limit 5;

select \* from wi limit 5;

select \* from ev3 limit 5;

**Step #6: Create summary Tables and test**

1. Create ev1\_sum table and test

|  |
| --- |
| DROP table if exists ev1\_sum;  CREATE TABLE IF NOT EXISTS ev1\_sum AS  SELECT  r\_year,  state,  make,  model,  COUNT(\*) AS vehicle\_count  FROM (  SELECT  SUBSTRING(registration\_valid\_date, 1, 4) AS r\_year,  state\_abbreviation AS state,  REGEXP\_EXTRACT(vehicle\_name, '^([^ ]+)', 1) AS make,  REGEXP\_EXTRACT(vehicle\_name, '^[^ ]+ (.+)$', 1) AS model  FROM ev1  ) AS extracted\_data  GROUP BY  r\_year,  state,  make,  model  ORDER BY  r\_year, state, make, model;  select \* from ev1\_sum limit 5; |

A screenshot of a computer screen

Description automatically generated

1. create tabel wi\_sum

|  |
| --- |
| DROP table if exists wi\_sum;  CREATE TABLE IF NOT EXISTS wi\_sum AS  SELECT  CASE  WHEN registration\_valid\_date IS NOT NULL AND registration\_valid\_date != ''  THEN SUBSTRING(registration\_valid\_date, -4)  ELSE NULL  END AS r\_year,  'WI' AS state,  make,  model,  COUNT(dmv\_id) AS vehicle\_count  FROM wi  GROUP BY  CASE  WHEN registration\_valid\_date IS NOT NULL AND registration\_valid\_date != ''  THEN SUBSTRING(registration\_valid\_date, -4)  ELSE NULL  END,  make,  model;  select \* from wi\_sum limit 5; |

1. create fl\_sum table

|  |
| --- |
| DROP table if exists fl\_sum;  CREATE TABLE IF NOT EXISTS fl\_sum AS  SELECT  CASE  WHEN registration\_valid\_date IS NOT NULL AND registration\_valid\_date != ''  THEN SUBSTRING(registration\_valid\_date, -4)  ELSE NULL  END AS r\_year, -- Extracts the year from MM/DD/YYYY format  'FL' AS state, -- Sets state as FL  REGEXP\_EXTRACT(vehicle\_name, '^([^ ]+)', 1) AS make, -- Extracts make from Vehicle Name  REGEXP\_EXTRACT(vehicle\_name, '^[^ ]+ (.+)$', 1) AS model, -- Extracts model from Vehicle Name  COUNT(\*) AS vehicle\_count  FROM fl  GROUP BY  CASE  WHEN registration\_valid\_date IS NOT NULL AND registration\_valid\_date != ''  THEN SUBSTRING(registration\_valid\_date, -4)  ELSE NULL  END,  REGEXP\_EXTRACT(vehicle\_name, '^([^ ]+)', 1),  REGEXP\_EXTRACT(vehicle\_name, '^[^ ]+ (.+)$', 1)  ORDER BY  r\_year, state, make, model;  select \* from fl\_sum limit 5; |

1. create wa\_sum table

|  |
| --- |
| DROP TABLE IF EXISTS wa\_sum;  CREATE TABLE IF NOT EXISTS wa\_sum AS  SELECT  CASE  WHEN registration\_valid\_date IS NOT NULL AND registration\_valid\_date != ''  THEN SUBSTRING(registration\_valid\_date, -4)  ELSE NULL  END AS r\_year, -- Extracts the year from MM/DD/YYYY format  state,  REGEXP\_EXTRACT(vehicle\_name, '^([^ ]+)', 1) AS make, -- Extracts make from Vehicle Name  REGEXP\_EXTRACT(vehicle\_name, '^[^ ]+ (.+)$', 1) AS model, -- Extracts model from Vehicle Name  COUNT(\*) AS vehicle\_count  FROM wa  GROUP BY  CASE  WHEN registration\_valid\_date IS NOT NULL AND registration\_valid\_date != ''  THEN SUBSTRING(registration\_valid\_date, -4)  ELSE NULL  END,  state,  REGEXP\_EXTRACT(vehicle\_name, '^([^ ]+)', 1),  REGEXP\_EXTRACT(vehicle\_name, '^[^ ]+ (.+)$', 1)  ORDER BY  r\_year, state, make, model;  select \* from wa\_sum limit 5; |

1. ev3\_sum table

|  |
| --- |
| DROP TABLE IF EXISTS ev3\_sum;  CREATE TABLE IF NOT EXISTS ev3\_sum AS  SELECT  CASE  WHEN registration\_date IS NOT NULL AND registration\_date != ''  THEN SUBSTRING(registration\_date, -4)  ELSE NULL  END AS r\_year, -- Extracts the year from MM/DD/YYYY format  state,  vehicle\_make AS make,  vehicle\_model AS model,  SUM(vehicle\_count) AS vehicle\_count  FROM ev3  GROUP BY  CASE  WHEN registration\_date IS NOT NULL AND registration\_date != ''  THEN SUBSTRING(registration\_date, -4)  ELSE NULL  END,  state,  vehicle\_make,  vehicle\_model  ORDER BY  r\_year, state, make, model;  SELECT \* FROM ev3\_sum limit 5; |

1. Union tables. Upper case make and model

|  |
| --- |
| DROP TABLE IF EXISTS all\_vehicles\_sum;  CREATE TABLE all\_vehicles\_sum AS  SELECT \* FROM ev1\_sum  UNION ALL  SELECT \* FROM wa\_sum  UNION ALL  SELECT \* FROM wi\_sum  UNION ALL  SELECT \* FROM fl\_sum  UNION ALL  SELECT \* FROM ev3\_sum;  SELECT \* FROM all\_vehicles\_sum LIMIT 5;  DROP TABLE IF EXISTS all\_veh;  CREATE TABLE all\_veh AS  SELECT r\_year, state, UPPER(make), UPPER(model), vehicle\_count  FROM all\_vehicles\_sum; |

1. Export all\_vel data in to Linux

|  |
| --- |
| INSERT OVERWRITE DIRECTORY '/user/jlee464/all\_veh'  ROW FORMAT DELIMITED  FIELDS TERMINATED BY ',' -- Use comma as the delimiter  SELECT \* FROM all\_veh; |

**Step #7: Download file from HDFS to Linux then download into your local computer**

-bash-4.2$ hdfs dfs -ls /user/jlee464/all\_veh

-bash-4.2$ hdfs dfs -get /user/jlee464/all\_veh/\*

-bash-4.2$ ls

A black background with white text

Description automatically generated

Use your username and local computer path.

scp **jlee464**@129.146.230.230:/home/jlee464/000000\_0 "F:\CSULA\CIS 5200\Project/EV/all\_veh.csv"



**Step #8: Use Tableau for charts**

1. Go to File > Open or Connect > Text File to load the following files:

* all\_veh.csv
* Alternative\_Fueling\_Stations\_3077267324180258876.csv
* Vehicle\_Population\_Last\_updated\_04-30-2024\_ada.csv
* New\_ZEV\_Sales\_Last\_updated\_08-06-2024\_ada.csv

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Create charts as following data source and drop variable as the pictures.

1. Filer up to 2020

Drag the date field (e.g., Registration Date) to the Filters shelf.

Set ranges like: Up to 2020

A screenshot of a computer

Description automatically generated

1. Filter up to 2020 and NY

A screenshot of a computer

Description automatically generated

1. Filter up to 2020 and CA

A screenshot of a computer

Description automatically generated

1. Filter up to 2020

Drag County to Rows and ZEV Count to Columns.

Change the Marks Type to Heatmap.

Filter data to focus on California.

A screenshot of a computer

Description automatically generated

1. Filter up to 2023 and NY with Table Chart

Drag Year, State, and Vehicle Make to Rows.

Drag Vehicle Count to Text on the Marks card.

Filter by years (e.g., 2021-2023).

A screenshot of a computer

Description automatically generated

1. Filter up to 2023 and CA with Table Chart

Drag Year, State, and Vehicle Make to Rows.

Drag Vehicle Count to Text on the Marks card.

Filter by years (e.g., 2021-2023).

A screenshot of a chat

Description automatically generated

1. Use Heatmap for EV Charger by State

Drag the State field to Rows.

Drag the Charger Count field to Columns.

Change the Marks type to "Square":

In the Marks card, select Square from the dropdown menu.

Drag Charger Count to the Color shelf in the Marks card.

Adjust the Color settings:

Click on the Color legend, choose a gradient (e.g., light to dark), and ensure the range is clear.

A screenshot of a computer

Description automatically generated

1. Use Heatmap for ZEV CA by county

Drag the State field to the Filters shelf.

Select California (or CA) as the filter value.

Drag the County field to Rows.

Drag the Vehicle Count field to Columns.

Change the Marks type to Square:

In the Marks card, select Square from the dropdown menu.

Drag the Vehicle Count field to the Color shelf in the Marks card.

A screenshot of a computer

Description automatically generated

1. Use Bar chart and filter Electric

Click on the "Sheet" tab at the bottom to open a new worksheet.

Rename the sheet to "CA ZEV 2024"

Drag the Vehicle Type field to the Filters shelf.

Select Electric as the filter value.

Apply the filter.

Drag the Year field to the Columns shelf to display time progression.

Drag the Vehicle Count field to the Rows shelf to show counts.

Drag State or Make to the Color or Detail shelf (optional):

This will split the bars by state or vehicle make for additional insight.

A screenshot of a computer

Description automatically generated

1. CA Fuel Type 2021-2023 Filter year between 2021 and 2023 use Table chart

Click on the "Sheet" tab at the bottom to open a new worksheet.

Rename the sheet to "CA Fuel Type 2021-2023"

Drag the Year field to the Filters shelf.

Set the filter condition:

Select Range of Dates or Years depending on the field type.

Specify the range from 2021 to 2023.

Apply the filter.

Drag the following fields to the Rows shelf:

* Fuel Type

Drag Vehicle Count to the Text area in the Marks card to display the counts.

A screenshot of a chat

Description automatically generated

1. Use Table chart and filter year between 2021 and 2023

Click on the "Sheet" tab at the bottom to open a new worksheet.

Rename the sheet to "2021 – 2023 CA ZEV Sales Make” for clarity.

Drag the Year field to the Filters shelf.

Select the Range of Years or Relative Dates filter option based on your data structure.

Specify the range as 2021 to 2023 and apply the filter.

Drag the following fields to the Rows shelf:

* Year
* State
* Vehicle Make

Drag Vehicle Count to the Text area in the Marks card to display the counts.

A screenshot of a computer

Description automatically generated

1. Use Table chart and filter year between 2021 and 2023

Click on the "Sheet" tab at the bottom to open a new worksheet.

Rename the sheet to "2021 – 2023 CA ZEV Sales Model" for clarity.

Drag the Year field to the Filters shelf.

Select the Range of Years or Relative Dates filter option based on your data structure.

Specify the range as 2021 to 2023 and apply the filter.

Drag the following fields to the Rows shelf:

* Year
* State
* Make
* Model

Drag Vehicle Count to the Text area in the Marks card to display the counts.

A screenshot of a computer

Description automatically generated

1. Create Dashboard

Click on the New Dashboard tab.

Drag individual sheets (e.g., heatmap, bar chart, table) onto the canvas.

Arrange and resize components for clarity.

Add interactive filters to the dashboard (e.g., by state or year).

A screenshot of a computer

Description automatically generated

1. Crate Dashboard

A screenshot of a computer

Description automatically generated

1. Create Dashboard

A screenshot of a computer

Description automatically generated

1. Create Storyboard

A map of the united states

Description automatically generated