//make a directory on HDFS and put input data for hive there

1. $hdfs dfs -mkdir HiveMonthlyCovidCases
2. $hdfs dfs -put OrganizeMonths/part-r-00000 HiveMonthlyCovidCases
3. $hdfs dfs -ls HiveMonthlyCovidCases //make sure file is there

//connect to hive

4. Beeline
5. !connect jdbc:hive2://babar.es.its.nyu.edu:10000/
6. use lam923

//we need to make a temporary table with the filler colum (the filler column is there to resolve an error that happens with MapR, where there is space between the key and the value that is not able to be removed by using trim() in hive. This space prevents from casting the column of the total monthly cases as an int.

7. create external table tempmonthdata (month int, state string, filler string,cases bigint) row format delimited fields terminated by ',' location '/user/lam923/HiveMonthlyCovidCases/';

//check if table is on hive

8. show tables;

//check that columns are correct

9. describe tempmonthdata;
10. select * from tempmonthdata;

//create the table without the filler column

11. create external table monthlystatecases (month int,state string,cases int);

//check that the table was created

12. show tables;
13. describe monthlystatecases;

//insert everything by the filler column from temp month data

14. INSERT INTO monthlystatecases SELECT month,state,cases FROM tempmonthdata;

//check that table is correct

15. select * from monthlystatecases;

//we need to fix the formatting of the months, create a new table to do so

16. create external table monthlycasesoutput (month string, state string, cases int);

//check that table is there

17. show tables;

//insert from monthly state cases into monthly state output, concatenating 2020- with the month. We do this step so that the team all has the months formatted the same

18. INSERT INTO monthlycasesoutput SELECT CONCAT('2020-',month),state,cases FROM monthlystatecases

//check that the table is correct

19. select * from monthlycasesoutput;

//write table to HDFS

20. INSERT OVERWRITE DIRECTORY '/user/lam923/hive_output' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY "\n" SELECT * FROM monthlycasesoutput;

//check that table is there on HDFS

21. $hdfs dfs -ls /user/lam923/hive_output
22. $hdfs dfs -cat /user/lam923/hive_output/000000_0

//Use Andrew's net ID to get the emissions table

23. use ae1586;

//check that it is correct

24. show tables;
25. select * from allstates;

//put Andrew's table into my HDFS directory so I can use it in my Hive databas

26. INSERT OVERWRITE DIRECTORY '/user/lam923/EmissionDatahive' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY "\n" SELECT * FROM allstates;

//check that it is in HDFS

27. $hdfs dfs -ls '/user/lam923/EmissionDataHive'
28. $hdfs dfs -cat '/user/lam923/EmissionDataHive/000000_0'

//go back to my Hive database

29. use lam923;

//create a Hive table from the emissions table

30. create external table tempcarbondata (state string, date string ,averageco2 float,medianco2 float,minco2 float,maxco2 float, totalrecords int) row format delimited fields terminated by ',' location '/user/lam923/EmissionDataHive/';

//check that table is correct

31. select * from tempcarbondata;

//we need to create a table that we will use to join the emissions and carbon data

32. create external table carbontableforjoin (identifier string,state string,month string,average float);

//insert the date, month, and average CO2 into the table for the join. We will use the average CO2 for our analytic

33. INSERT INTO carbontableforjoin SELECT CONCAT(date,'-',state),state,date,averageco2 FROM tempcarbondata;

//check that table is correct

34. select * from carbontableforjoin;

//create a covid cases table that w will use from the join

35. create external table covidcasestableforjoin (identifier string,month string,state string,cases int);

//insert the trimmed month, state and cases from the monthly cases outpiut

36. INSERT INTO covidcasestableforjoin SELECT CONCAT(trim(month),'-',trim(state)), month,state,cases FROM monthlycasesoutput;

//check that table is correct

37. select * from covidcasestableforjoin;

//create the table that will combine the emissions and covid table

38. create external table covidandemissions (date string, state string, cases int, co2average float);

//join the covid and emissions join table by the date and the state

39. INSERT INTO covidandemissions SELECT c.MONTH, c.STATE, c.CASES, e.AVERAGE FROM COVIDCASESTABLEFORJOIN c JOIN CARBONTABLEFORJOIN e ON (c.IDENTIFIER=e.IDENTIFIER);

//check that table is correct

40. select * from covidandemissions;

//insert table into HDFS

41. INSERT OVERWRITE DIRECTORY '/user/lam923/covidandemissions_output' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY "\n" SELECT * FROM covidandemissions;

//check that table is there

42. $ hdfs dfs -ls /user/lam923/covidandemissions_output
43. $ hdfs dfs -cat /user/lam923/covidandemissions_output/000000_0
44. $ hdfs dfs -get /user/lam923/covidandemissions_output
45. $ls

//move table to local (incase Hive is down and we need to access it)

46. scp lam923@dumbo.es.its.nyu.edu:/home/lam923/covidandemissions_output/000000_0

//create a table that will be used to calculate the correlation coefficient . In this table we will store the variables we need for the correlation coefficent (number of cases, the co2 average, the number of cases*the co2 average, the number of cases ^2 and the co2 average ^2) for each state and each month

47. create external table step1_correlation_data (date string, state string, cases int, co2average float, product double,cases_squared bigint,co2_squared float);

//insert the table from the joined cases and emissions table into the table that will be used for the correlation step

48. INSERT INTO step1_correlation_data SELECT date, state, cases,co2average, cases*co2average,CAST(cases AS BIGINT)*CAST(cases AS BIGINT),co2average*co2average FROM covidandemissions;

//check that the table is correct

49. select * from step1_correlation_data;

//create a table for the correlation coefficient

50. create external table correlation_results (state string, correlation double);

//for each state, calculate the correlation coefficient using the formula ((9*sum(product))-(sum(cases)*sum(co2average)))/(sqrt(((9*sum(cases_squared))-(sum(cases)*sum(cases)))*((9*sum(co2_squared))-(sum(co2average)*sum(co2average))))) from step1_correlation_data group by state (we use 9 because we are calculating the coefficient from data from 9 months).

51. INSERT INTO correlation_results select state ((9*sum(product))-(sum(cases)*sum(co2average)))/(sqrt(((9*sum(cases_squared))-(sum(cases)*sum(cases)))*((9*sum(co2_squared))-(sum(co2average)*sum(co2average))))) from step1_correlation_data group by state;

//check that data is correct, view in different orders to get an idea of the results

52. select * from correlation_results;

53. select * from correlation_results order by correlation DESC;
54. select * from correlation_results order by correlation ASC;

//create the table we will use for the output

55. create table correlation_output (state string, correlation double);

//insert the correlation coefficient in ascending order

56. INSERT INTO correlation_output select state, correlation from correlation_results order by correlation ASC;

//check that the table is correct

57. select * from correlation_output;

//move to HDFS

58. INSERT OVERWRITE DIRECTORY '/user/lam923/correlation_output' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY "\n" SELECT * FROM correlation_output;

//check is correct

59. $hdfs dfs -ls '/user/lam923/correlation_output'
60. $hdfs dfs -cat '/user/lam923/correlation_output/000000_0'