Problem Set E Submission Form

Overview

Your Name	Hendi Kushta
Your SU Email	hkushta@syr.edu

Instructions

Put your name and SU email at the top. Answer these questions all from the lab. When asked to include screenshots, please follow the screen shot guidelines from the first homework.

Remember as you complete the homework it is not only about getting it right / correct. We will discuss the answers in class so it's important to articulate anything you would like to contribute to the discussion in your answer:

- If you feel the question is vague, include any assumptions you've made.
- If you feel the answer requires interpretation or justification provide it.
- If you do not know the answer to the question, articulate what you tried and how you are stuck.
- Highlight any doubts or questions you would like me to review.

This how you receive credit for answering questions which might not be correct. In addition, you must complete the reflection portion of the homework assignment for full credit. Since most answers will be similar this is an important part of your individual submission.

Complete Part II of this document first, then go back and complete the Reflection in Part I.

Part I - Reflection

Use this section to reflect on your learning. To achieve the highest grade on the assignment you must be as descriptive and personal as possible with your reflection.

1. As you completed this assignment, identify what you learned.

Queried data from a variety of sources with Spark SQL. Handled Structured and Semi-structured data with Drill. Recognized the differences between Drill SQL and Spark SQL

2. What barriers or challenges did you encounter while completing this assignment?

3. How prepared were you to complete this assignment? What can you do to be better prepared?

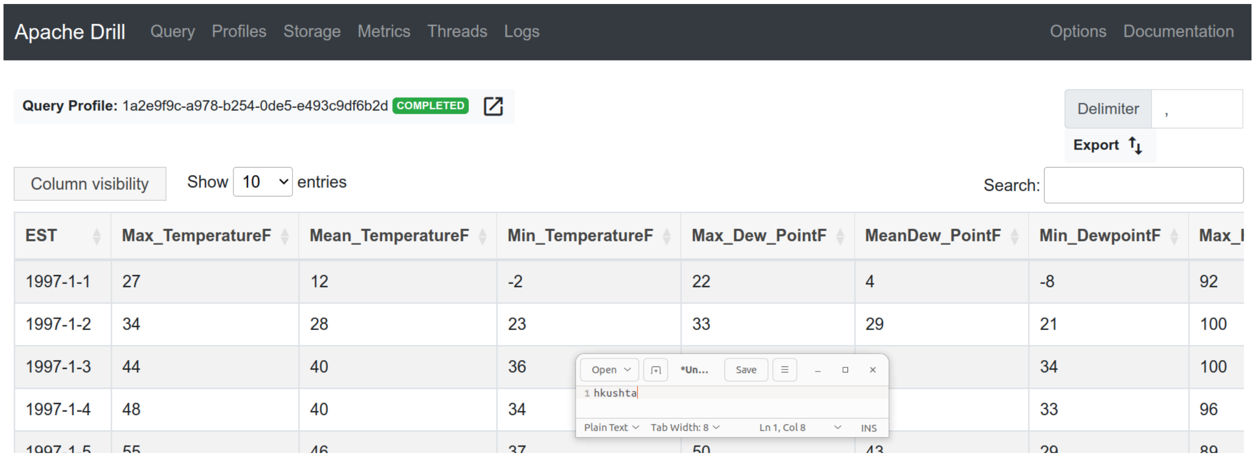
4. Rate your comfort level with this week’s material. Use the rubric provided.

4 ==> I understand this material and can explain it to others.
3 ==> I understand this material.
2 ==> I somewhat understand the material but sometimes need guidance from others.
1 ==> I understand very little of this material and need extra help.

Part II – Questions

For each question, include a copy of the code required to complete the question along with a screenshot of the code and a screenshot of the output.

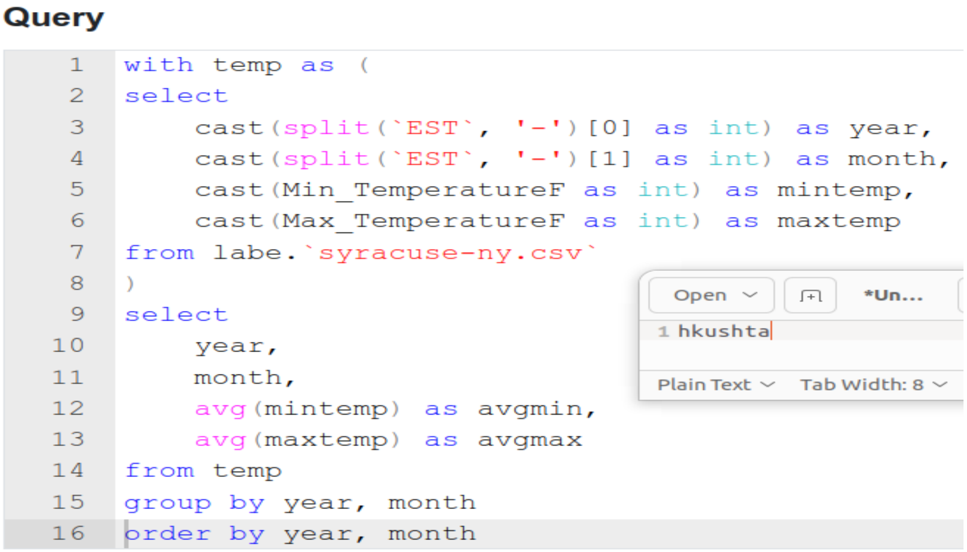
1. Configure a Drill storage plugin for the Minio **labe** bucket. Then write a drill query for **syracuse-ny.csv** to demonstrate you can read the file with headers.



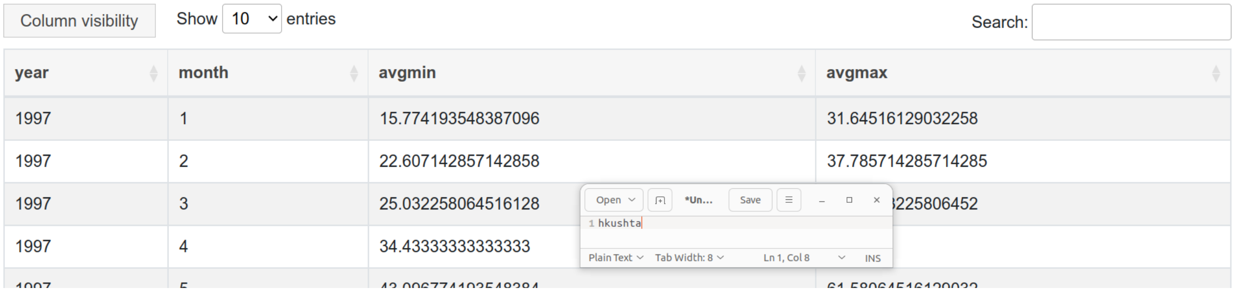
Query Profile: 1a2e9f9c-a978-b254-0de5-e493c9df6b2d

EST	Max_TemperatureF	Mean_TemperatureF	Min_TemperatureF	Max_Dew_PointF	MeanDew_PointF	Min_DewpointF	Max_I
1997-1-1	27	12	-2	22	4	-8	92
1997-1-2	34	28	23	33	29	21	100
1997-1-3	44	40	36			34	100
1997-1-4	48	40	34			33	96
1997-1-5	55	46	37	50	43	29	89

2. Write a Drill SQL Query to get the overall average min and max temperatures by year and month. Use drill's SPLIT() function to separate Year, Month. You might need to use cast() to ensure the min and max temperatures are numeric types. You output should include 4 columns: Year, Month, the average minimum temperature for that month, and the average maximum temperature for that month.

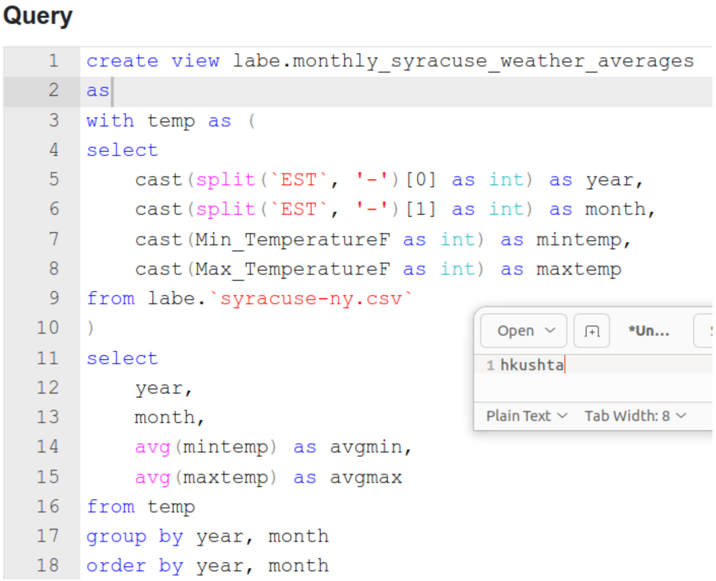


```sql
with temp as (
select
    cast(split(`EST`, '-')[0] as int) as year,
    cast(split(`EST`, '-')[1] as int) as month,
    cast(Min_TemperatureF as int) as mintemp,
    cast(Max_TemperatureF as int) as maxtemp
from labe.`syracuse-ny.csv`
)
select
    year,
    month,
    avg(mintemp) as avgmin,
    avg(maxtemp) as avgmax
from temp
group by year, month
order by year, month
```

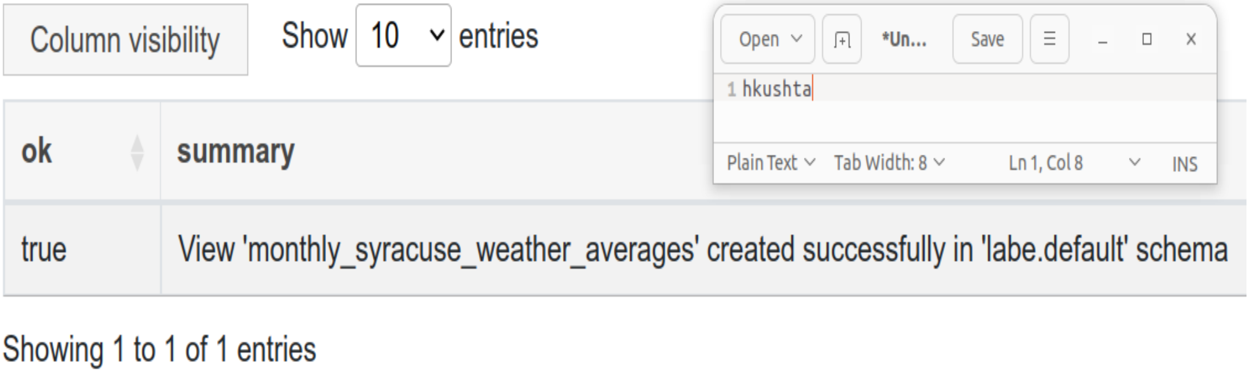


year	month	avgmin	avgmax
1997	1	15.774193548387096	31.64516129032258
1997	2	22.607142857142858	37.785714285714285
1997	3	25.032258064516128	225806452
1997	4	34.43333333333333	
1997	5	43.096774193548384	61.58064516129032

3. Create a view called **monthly_syracuse_weather_averages** from the query you wrote in question 2 and store it back on the **labe** bucket. (If you cannot get question 2 working, use a similar query). Provide your drill SQL code and a screenshot showing the view file is on the Minio bucket.
   NOTE: If you get an error about an immutable object, you need to change your storage config so you can write to the storage location.

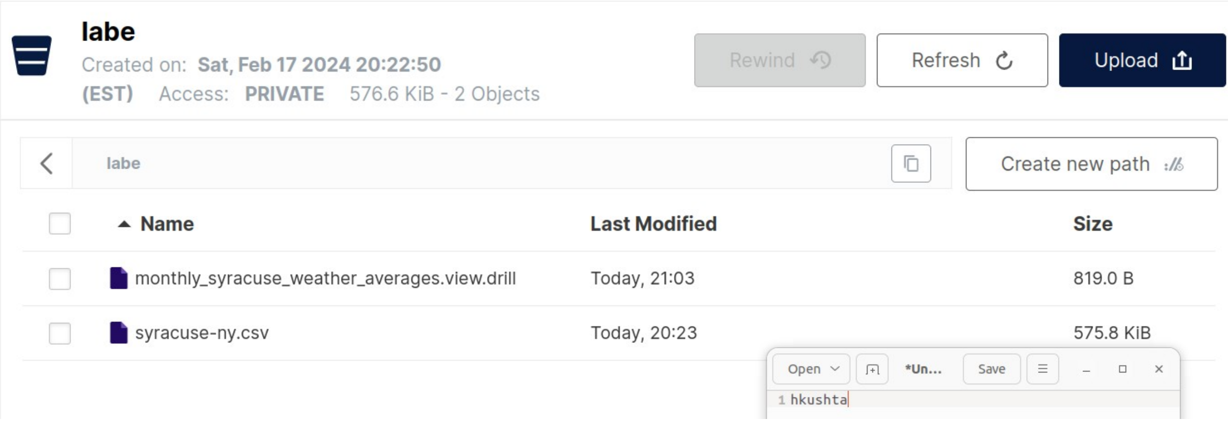


```
create view labe.monthly_syracuse_weather_averages
as
with temp as (
select
    cast(split(`EST`, '-')[0] as int) as year,
    cast(split(`EST`, '-')[1] as int) as month,
    cast(Min_TemperatureF as int) as mintemp,
    cast(Max_TemperatureF as int) as maxtemp
from labe.`syracuse-ny.csv`
)
select
    year,
    month,
    avg(mintemp) as avgmin,
    avg(maxtemp) as avgmax
from temp
group by year, month
order by year, month
```



ok	summary
true	View 'monthly_syracuse_weather_averages' created successfully in 'labe.default' schema

Showing 1 to 1 of 1 entries

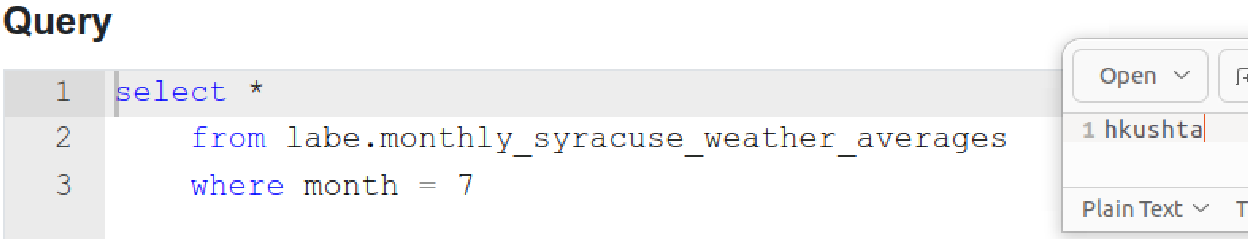


labe

Created on: Sat, Feb 17 2024 20:22:50 (EST) Access: PRIVATE 576.6 KiB - 2 Objects

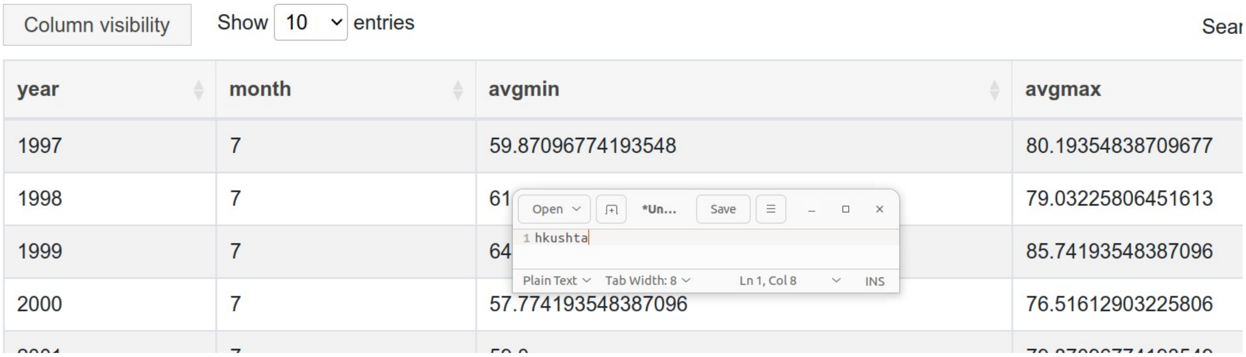
Name	Last Modified	Size
monthly_syracuse_weather_averages.view.drill	Today, 21:03	819.0 B
syracuse-ny.csv	Today, 20:23	575.8 KiB

4. Use the view you created in question 3 to show the weather data only the month of July.



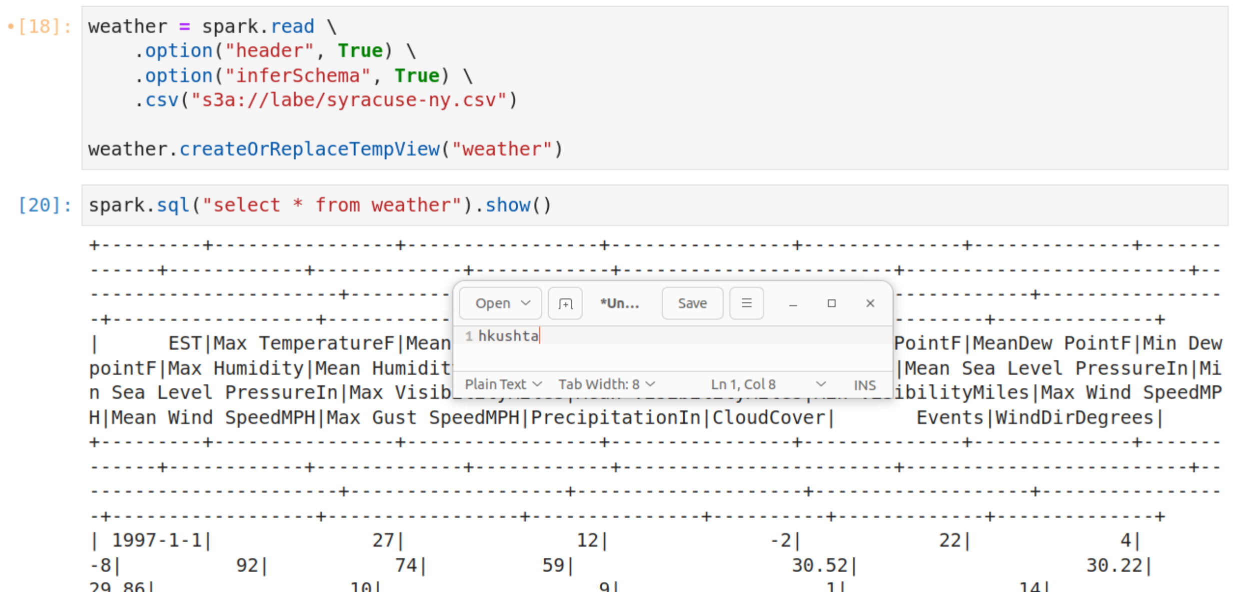
Query

```
select *
     from labe.monthly_syracuse_weather_averages
     where month = 7
```



year	month	avgmin	avgmax
1997	7	59.87096774193548	80.19354838709677
1998	7	61	79.03225806451613
1999	7	64	85.74193548387096
2000	7	57.774193548387096	76.51612903225806

5. Configure spark to read from Minio **labe** bucket, then load **syracuse-ny.csv** into a DataFrame as register it as the table **weather**

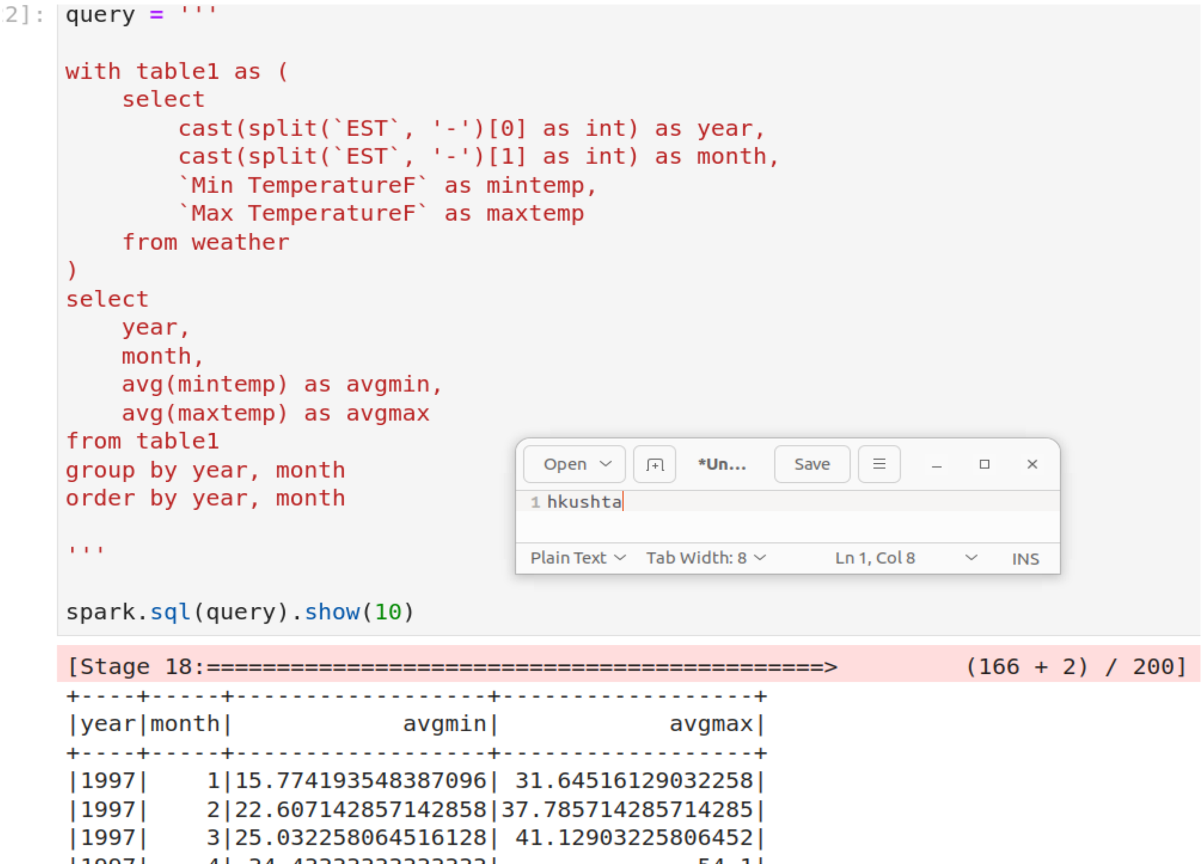


```
•[18]: weather = spark.read \
           .option("header", True) \
           .option("inferSchema", True) \
           .csv("s3a://labe/syracuse-ny.csv")

       weather.createOrReplaceTempView("weather")
```

```
 [20]: spark.sql("select * from weather").show()
```

```
+---------+----------------+-----------------+----------------+--------------+--------------+-------
------+------------+-------------+----------------+-------------------------+-------------------------+--
---------------------+--------                                   ----------+----------------
-+-----------------+----------                                   --------+--------------+
|      EST|Max TemperatureF|Mean                                   PointF|MeanDew PointF|Min Dew
pointF|Max Humidity|Mean Humidit                                   |Mean Sea Level PressureIn|Mi
n Sea Level PressureIn|Max Visib                                   ibilityMiles|Max Wind SpeedMP
H|Mean Wind SpeedMPH|Max Gust SpeedMPH|PrecipitationIn|CloudCover|       Events|WindDirDegrees|
+---------+----------------+-----------------+----------------+--------------+--------------+-------
------+------------+-------------+----------------+-------------------------+-------------------------+--
---------------------+--------------------+-------------------+---------------------+----------------
-+-----------------+-----------------+---------------+----------+-------------+--------------+
| 1997-1-1|              27|               12|              -2|            22|             4|
-8|          92|           74|          59|                      30.52|                    30.22|
29.86|                   10|                 9|                 1|                14|
```

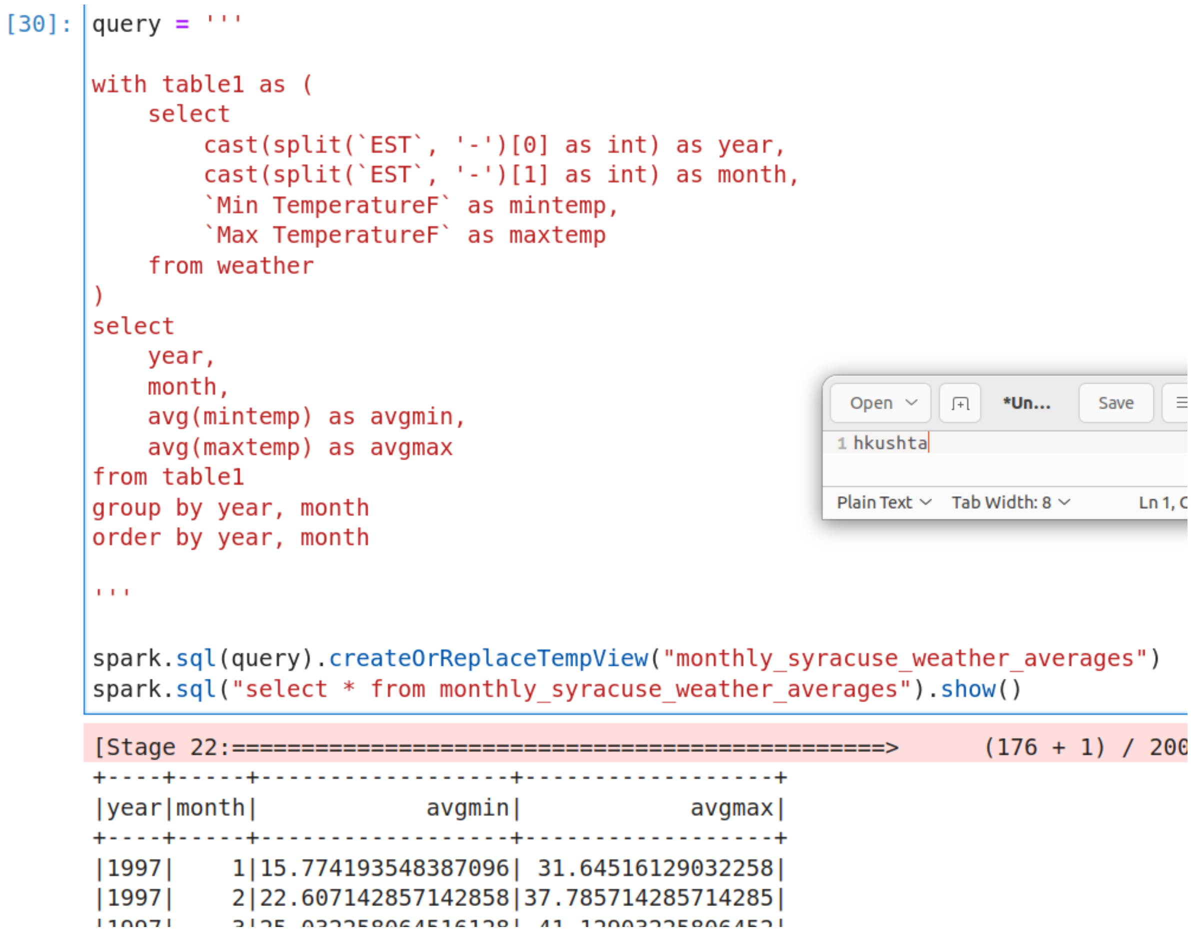
6. Rewrite question 2 using pure Spark SQL and the **weather** temp view. NOTE: There will be some subtle differences with how you must write the code, so be sure to **printSchema()** so you can see what the columns are.



```
2]: query = '''

    with table1 as (
        select
            cast(split(`EST`, '-')[0] as int) as year,
            cast(split(`EST`, '-')[1] as int) as month,
            `Min TemperatureF` as mintemp,
            `Max TemperatureF` as maxtemp
        from weather
    )
    select
        year,
        month,
        avg(mintemp) as avgmin,
        avg(maxtemp) as avgmax
    from table1
    group by year, month
    order by year, month

    '''

    spark.sql(query).show(10)
```

```
[Stage 18:============================================>          (166 + 2) / 200]
+----+-----+------------------+------------------+
|year|month|            avgmin|            avgmax|
+----+-----+------------------+------------------+
|1997|    1|15.774193548387096| 31.64516129032258|
|1997|    2|22.607142857142858|37.785714285714285|
|1997|    3|25.032258064516128| 41.12903225806452|
|1997|    4| 24.43333333333333|              54.1|
```

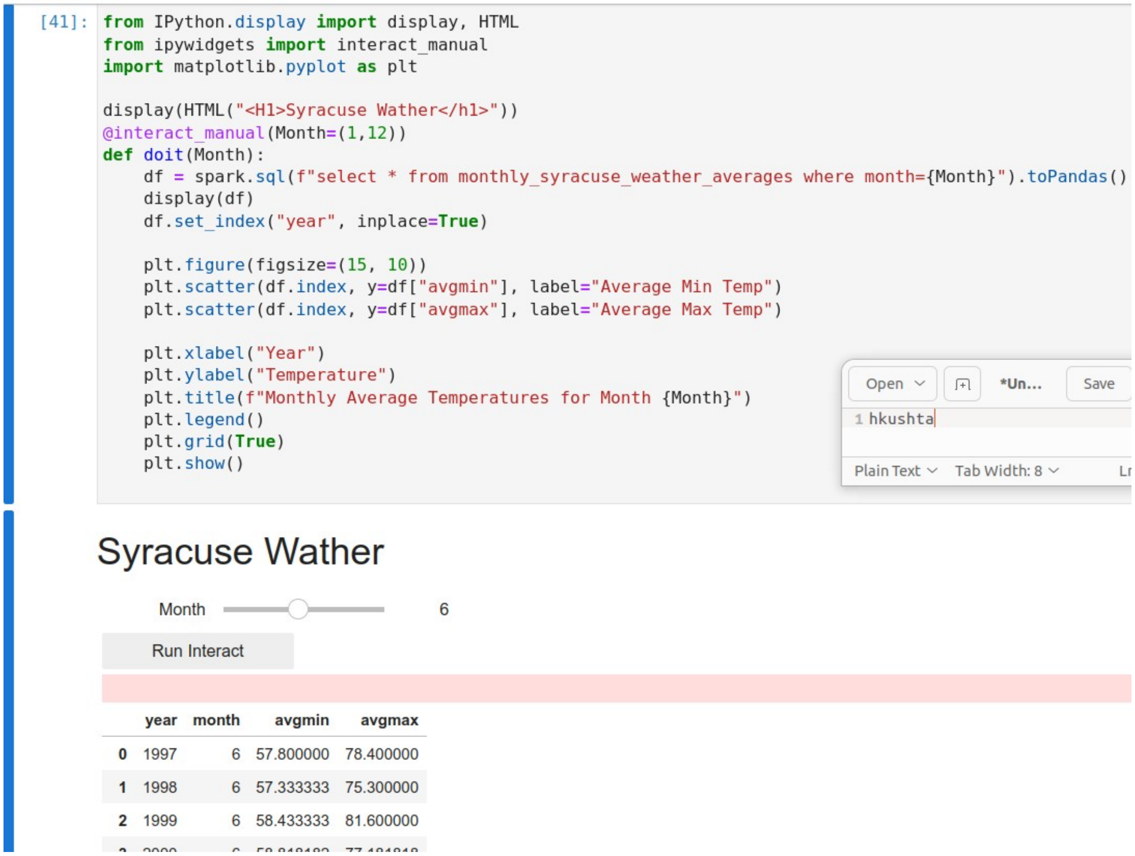
7. Save the output from the DataFrame in question 6 to the temp view **monthly_syracuse_weather_averages**. Prove the view is there by querying it.



```
[30]: query = '''

      with table1 as (
          select
              cast(split(`EST`, '-')[0] as int) as year,
              cast(split(`EST`, '-')[1] as int) as month,
              `Min TemperatureF` as mintemp,
              `Max TemperatureF` as maxtemp
          from weather
      )
      select
          year,
          month,
          avg(mintemp) as avgmin,
          avg(maxtemp) as avgmax
      from table1
      group by year, month
      order by year, month

      '''

      spark.sql(query).createOrReplaceTempView("monthly_syracuse_weather_averages")
      spark.sql("select * from monthly_syracuse_weather_averages").show()
```

```
[Stage 22:===============================================>      (176 + 1) / 200
+----+-----+------------------+------------------+
|year|month|            avgmin|            avgmax|
+----+-----+------------------+------------------+
|1997|    1|15.774193548387096| 31.64516129032258|
|1997|    2|22.607142857142858|37.785714285714285|
|1997|    3|25.032258064516128| 41.12903225806452|
```

8. CHALLENGE YOURSELF! At the bottom of the **work/content/E-Drill-Spark.ipynb** file there is a section Called “Big Data to Small Data”. Try to write a complete program that:
    a. Inputs a month 1 – 12 at run-time.
    b. Displays a scatter plot of min/max average monthly temperatures, where year is on the X-Axis.



[41]:

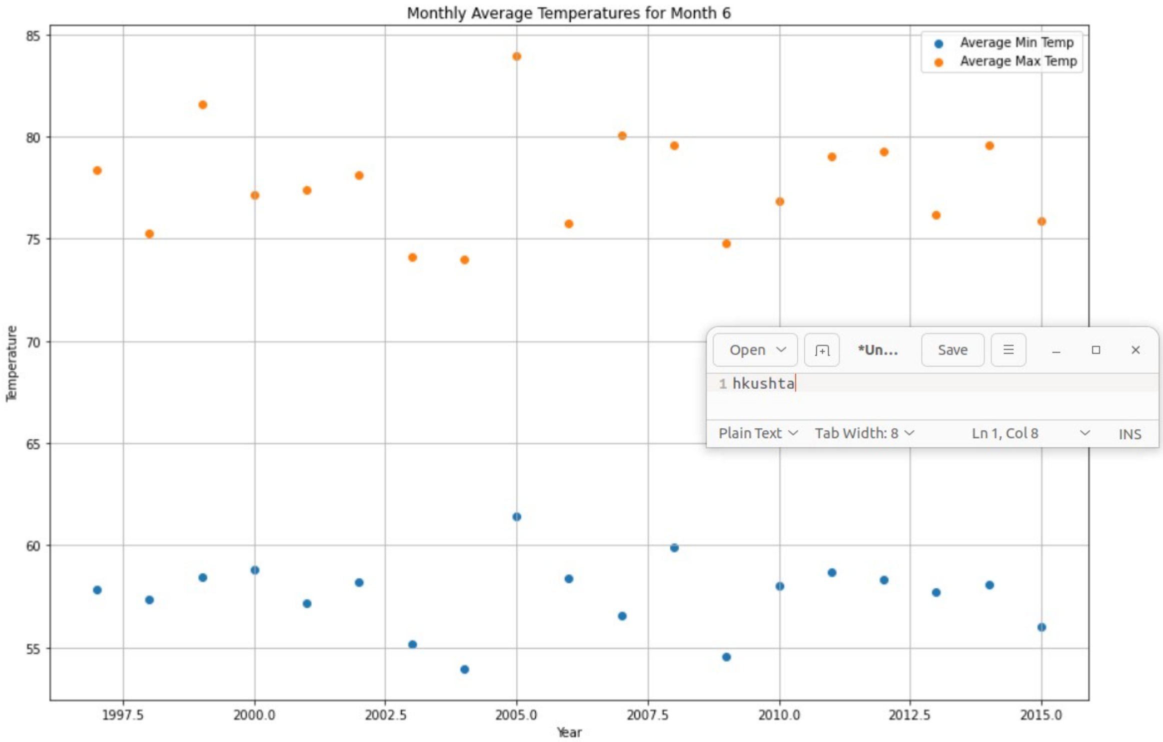
```
from IPython.display import display, HTML
from ipywidgets import interact_manual
import matplotlib.pyplot as plt

display(HTML("<H1>Syracuse Wather</h1>"))
@interact_manual(Month=(1,12))
def doit(Month):
    df = spark.sql(f"select * from monthly_syracuse_weather_averages where month={Month}").toPandas()
    display(df)
    df.set_index("year", inplace=True)

    plt.figure(figsize=(15, 10))
    plt.scatter(df.index, y=df["avgmin"], label="Average Min Temp")
    plt.scatter(df.index, y=df["avgmax"], label="Average Max Temp")

    plt.xlabel("Year")
    plt.ylabel("Temperature")
    plt.title(f"Monthly Average Temperatures for Month {Month}")
    plt.legend()
    plt.grid(True)
    plt.show()
```

hkushta

Syracuse Wather

Month 6

	year	month	avgmin	avgmax
0	1997	6	57.800000	78.400000
1	1998	6	57.333333	75.300000
2	1999	6	58.433333	81.600000
3	2000	6	58.818182	77.181818



hkushta