Athena Lab

Team 1 – Patty, Andrea, Bryce, Miguel, Patrick

6/22/2024

**Concise Narrative Explaining Lab**

As we saw in class with the Ohio voter data, Athena is a powerful tool for analyzing data housed in Amazon S3 using SQL. This lab allowed us to do just that, running SQL queries on the NYC taxi dataset using our traditional experience in R, and contrasting it with running the queries using Athena.

Our initial step in this lab was to import the NYC taxi cab data into R studio for a single month of 2018. This was a lengthy process and provided great context for the value of Athena from an efficiency perspective.

We then created a new user and configured Athena, creating a new table with the NYC taxi data. We ran a test query on the table within Athena, finding the query to run *significantly* faster than this process ran in R.

Lastly, we combined the two approaches and connected R to Athena within the R Studio environment on a virtual machine from the Mendoza Analytics Virtual Computer Lab. This allowed us to use ggplot to create the data visualization, but leverage the speed and efficiency of Athena. After some initial roadblocks connecting R and Athena, the query was a success and we were able to very quickly analyze the entire year of taxi data before visualizing the data using ggplot. It was very cool to see the two integrated, and incredible how quickly the query ran vs. the original query we ran in R at the beginning of the lab.

**Question Responses:**

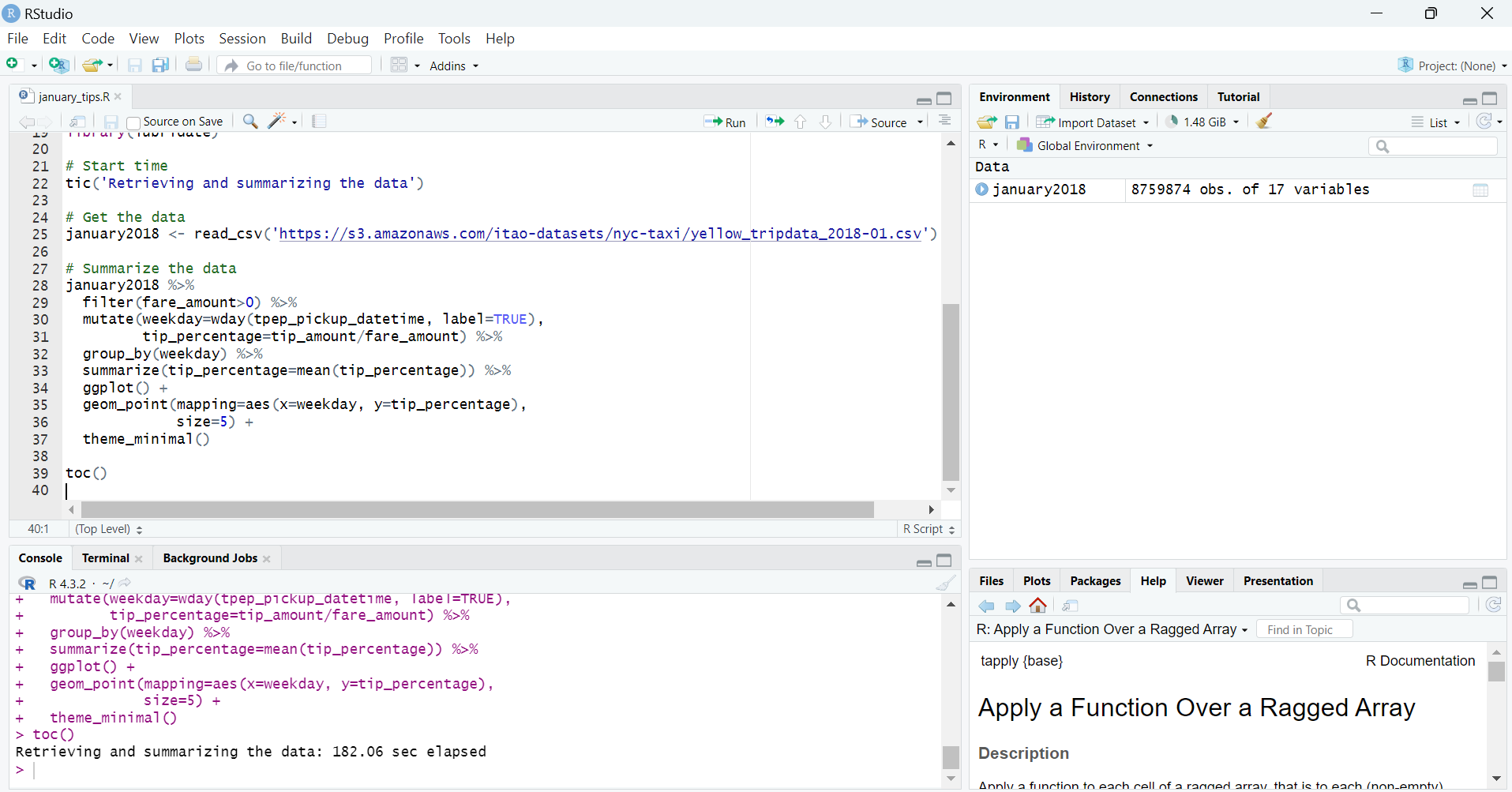
A – SQL query I write is as follows:

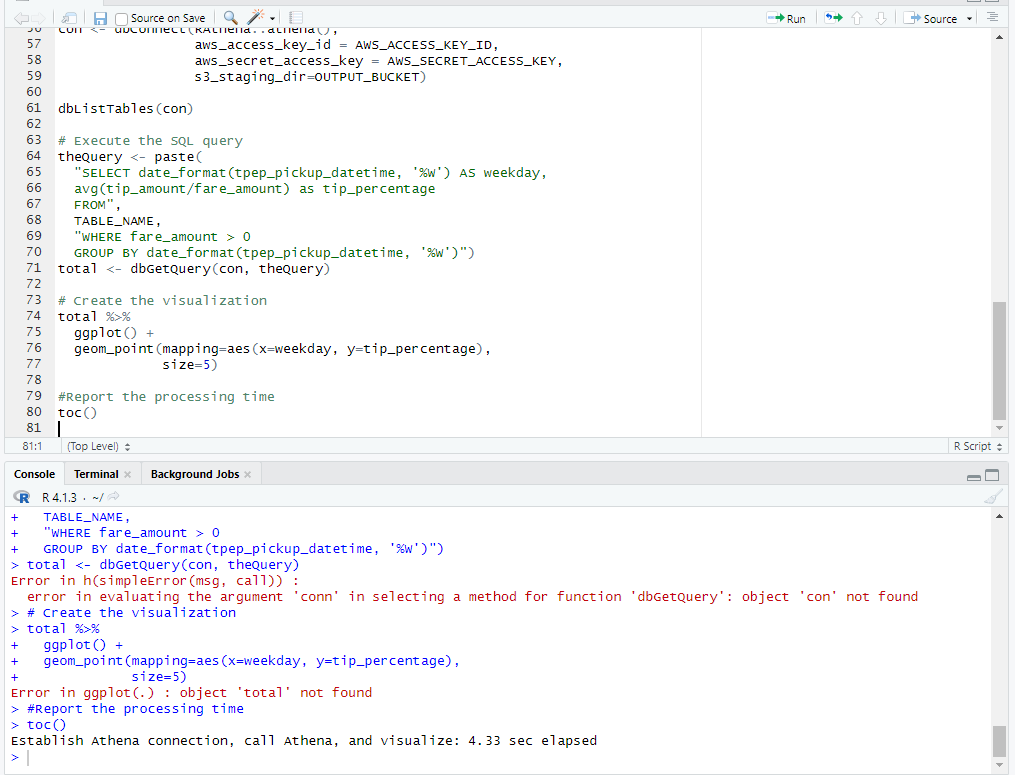
SELECT COUNT(\*) AS total\_rows

FROM ride\_data\_pgrodach;

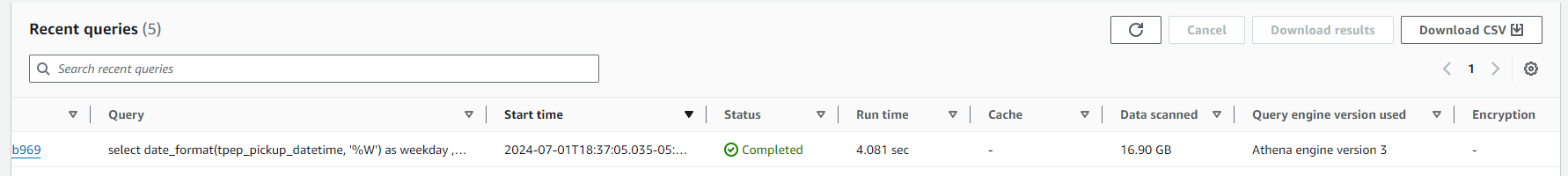
The total number of rows 2,056,13,140

B -- To estimate how long it would take running a similar script on the entire dataset, we would consider that the entire dataset contains 12 files (one for each month). If one month takes 182.06 seconds, the estimated time for the entire dataset would be 12 x 182.06 = 2184.72 = 36 minutes. It only took 4.081 sec to run a query across the entire dataset using Athena. It took roughly 4.33 seconds to create the visualization using Athena and R together.

* Script for one month of data w/o using Athena = 182.06 seconds!
* It took only 4.33 seconds to run the query while connecting Athena in R.



C – To estimate the cost of executing an Athena query, the amount of data scanned by Athena has to be considered.. Amazon Athena charges based on the amount of data scanned, at $5 per TB. Approximately 16.90 GB of data was scanned, so 16.9/1024(GB per 1 TB) \* $5 = $0.08.



D – In Section I, the data transfer involves reading a month's worth of data directly into RStudio using a CSV file, which means the entire content of the January 2018 CSV file is transferred. In Section VI, the data transfer occurs by querying Athena from RStudio, which returns a summarized dataset (tip percentages by weekday), resulting in the transfer of only the query result, which is much smaller in size compared to the entire CSV file. The key difference is that in Section I, the entire CSV file is transferred, representing a large data volume, whereas in Section VI, only the query results are transferred, representing a smaller data volume.

E – Running queries locally in RStudio is advantageous for smaller datasets or when quick, iterative analysis is needed, as it incurs no additional cloud costs and is faster for small data processing and visualization. Example queries suited for this environment include data cleaning and transformation, exploratory data analysis on small datasets, and local visualizations and quick plots. On the other hand, Amazon Athena is designed to handle large datasets efficiently without the need to download them locally, offering scalability and integration with other AWS services. Example queries for Athena include:

* aggregations and summaries over large datasets
* complex joins and data transformations
* pre-processing data for machine learning or advanced analytics.

By considering data volume, processing requirements, and cost, you can decide whether to run queries locally or on a service like Athena. For example, executing a script on a single month of data takes 9.97 seconds, while running a similar script on the entire dataset is estimated to take around 2 minutes. Athena's cost is based on the data scanned, priced at $5 per TB.

Data transfer differences include transferring the entire CSV file versus only the summarized query results. Therefore, local execution is preferable for small data, while Athena is suitable for large-scale processing. Providing exact execution times from the Athena console and RStudio, along with the data scanned, would allow for more precise answers.

F— If we were running an analytics *team*, we would choose RStudio Server. We may have had a different answer at the beginning of the Cloud course, but seeing the benefit of additional computing power, resource management tools, and collaboration benefits to a cloud-based environment make it a better tool for a team. For us individually, R Studio Desktop has been sufficient for coursework, but not in a team environment.

RStudio Desktop Pros – Easy to set up and use, no internet connection required. Great for

use on airplanes, etc.

RStudio Desktop Cons—Computing power is limited. We saw this very clearly in the lab, and

in the Ohio Voter example in class. Additionally, the lack of a professional version-control feature like the ones available in the cloud environments and difficulty with regulating the activity of users are negative with respect to R Desktop.

RStudio Server Pros—Ability to leverage additional compute resources allows for quicker queries than the Desktop version. Version-control, security benefits, ability to monitor users, and easier collaboration with the cloud-based model are also beneficial for a team environment.

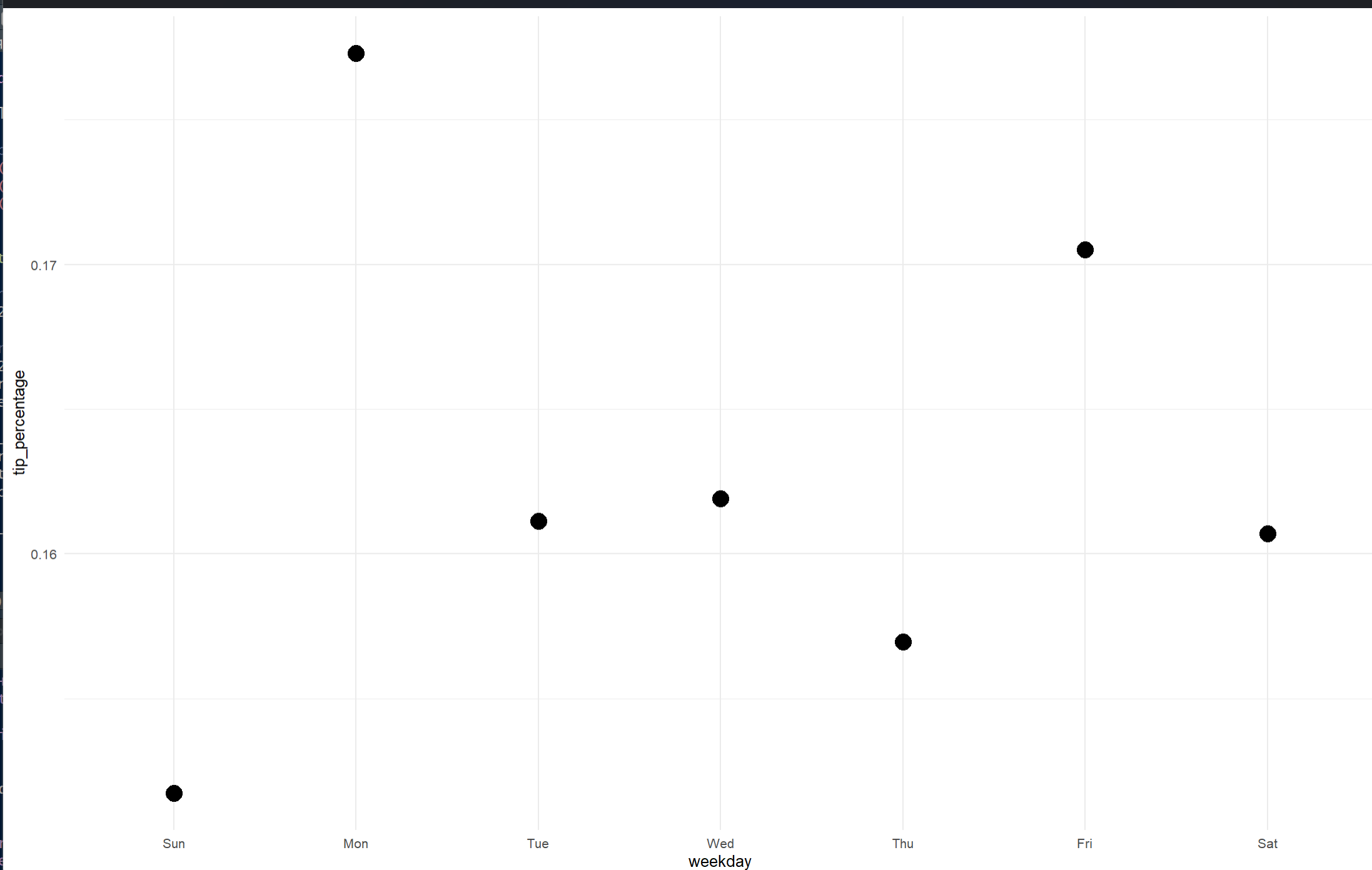
RStudio Server Cons—Its not as easy to get started as it is with RStudio desktop (i.e., you can’t simply download and start coding DPLYR like we did in this program). Its also a more expensive option and requires internet connectivity.

G— Three interesting conclusions from Todd Schneider’s work:

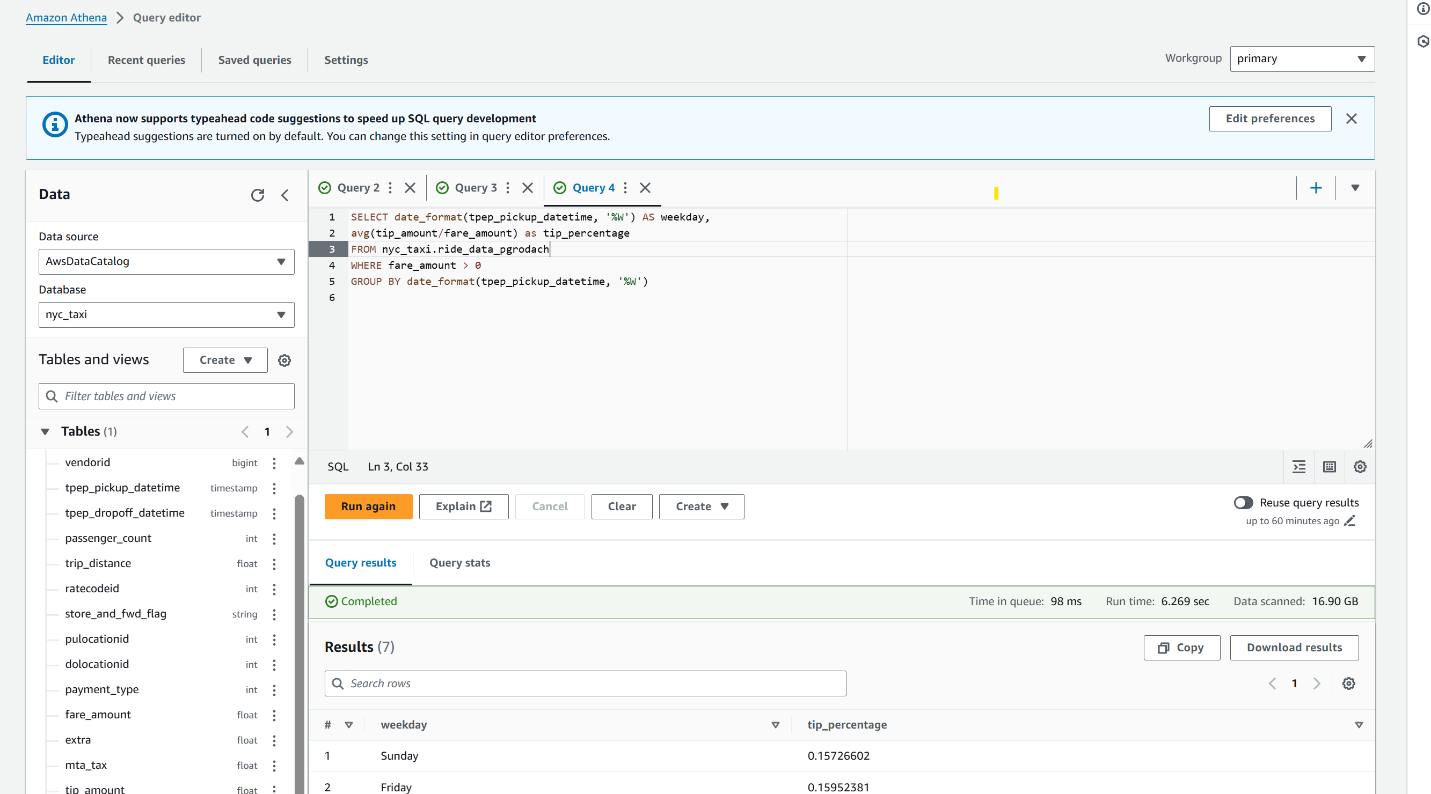
* It is incredible how telling the maps are in telling the story of how pickup frequency in certain locales (i.e., Manhattan) differs from drop-offs. Its really cool to compare these two maps and see a story that you wouldn’t be able to see without the visualization tool.
* Could Bruce Willis and Samuel L. Jackson have made it from the Upper West Side to Wall Street in 30 minutes? Apparently they *very easily* could do just that. This would have been a good assignment for the movie makers to work through prior to planning this sequence for the film. Seems like they had a relatively pedestrian trip during the film.
* The comparison of the median drop off times at Goldman vs. CITI was fascinating. Goldman employees are dropped off at a median time of 7:59AM, while CITI employees are dropped off at 7:51AM. The 8 minute difference felt material to me, but I also with fascinated at how close the Goldman employees are to nailing the perfect 8:00AM arrival time, given the variability associated with relying on a taxi to get to work.

**Lab Screenshots:**

* Plot of original R script from Section 1. Note that it took 293 seconds for the R script to run. Assuming all files are of equal size and the file is the bottleneck, I would estimate that the entire year of 2018 would take (293 seconds \* 12 files) = 3,516 seconds:



* Athena Test Query from Part V. This query took 6.29 seconds to run:



* Screenshot of R Plot and Runtime from section VI. This query ran in 17.66 seconds!:

