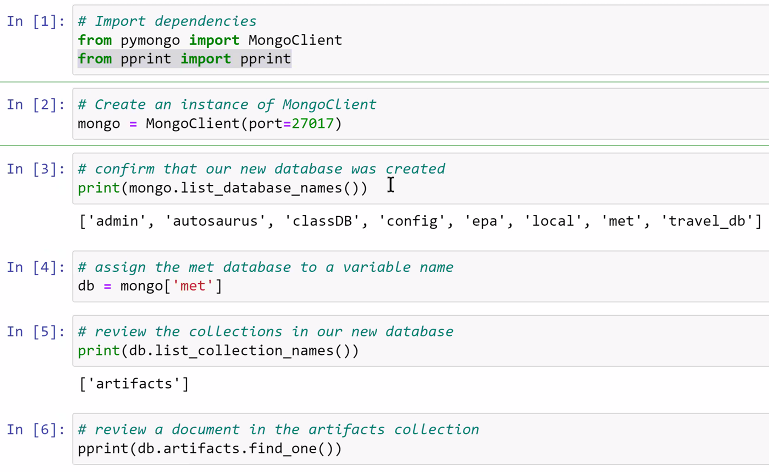
**12-5 notes**

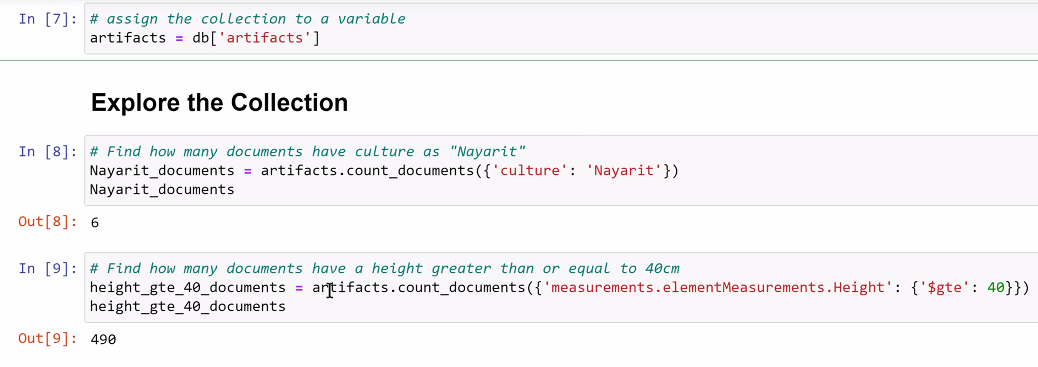
Aggregation, Analysis, and Integration with MongoDB

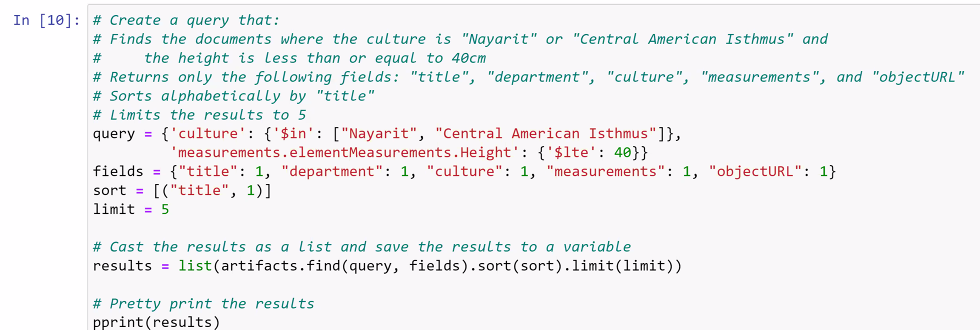
Goals:

* Use aggregation and aggregation pipelines with MongoDB to analyze a subset of a MongoDB collection.
* Convert a MongoDB result to a Pandas DataFrame.
* Import data from an API to save to MongoDB.
* Use data from a Mongo database to plot charts with Matplotlib.

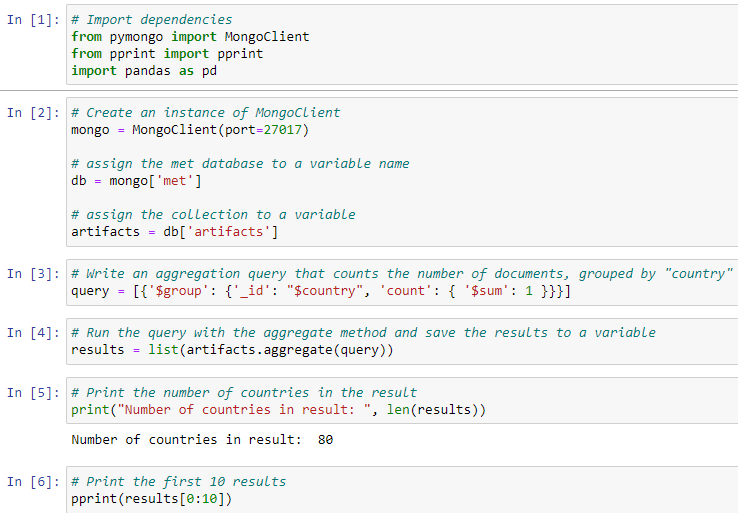
**MongoDB Warmup Solution:**

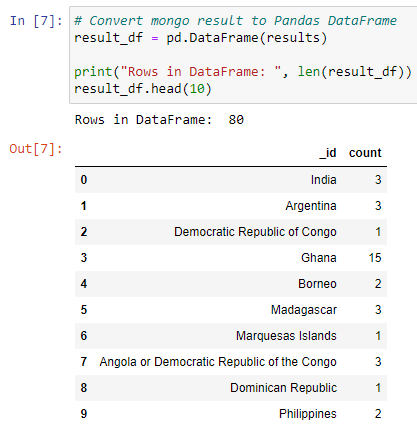






MongoDB Aggregation Solution:





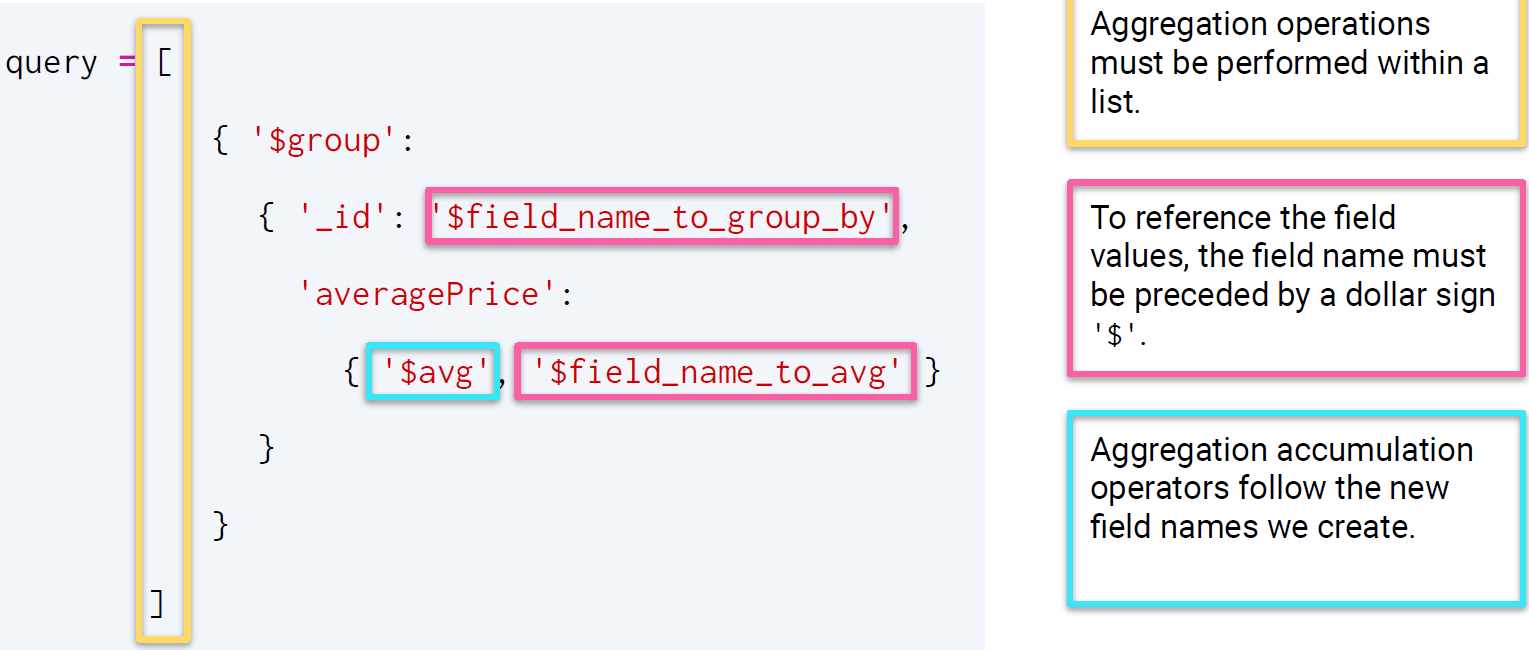
^ IN [4]- the column ***count*** didn’t exist before. You made it by calling it in this line.

Review 7:08pm in recording- More about the sum in IN[4]

<https://www.mongodb.com/docs/manual/reference/operator/aggregation/sum>



^ must use underscore before id.



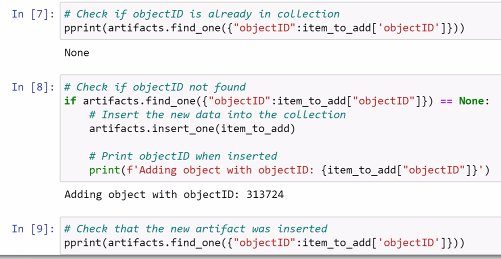
To reference field values, the field name must be preceded by a dollar sign. Ie: **$field\_name**

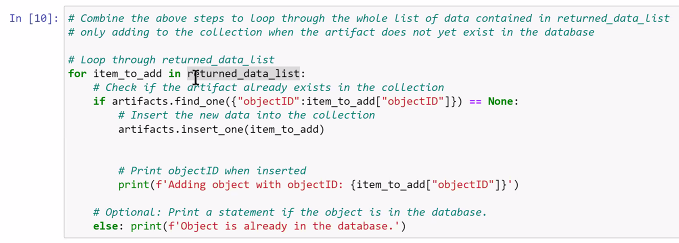
**Why build an aggregation pipeline in MongoDB?**

* When there are hundreds or thousands of documents in a collection, using an aggregation pipeline reduces the amount of information returned.
* e.g. If the database is stored in the cloud, when the operations are performed within the database cluster, this can save us time and significantly reduce the storage space needed on our local computers to process the data.

Review 7:20-7:35pm in the recording.- Aggregate pipeline activity

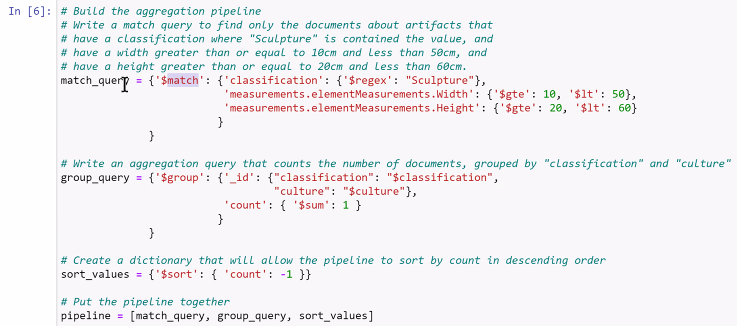
**Mini project! – Very useful for project due Tuesday**

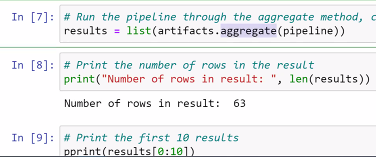


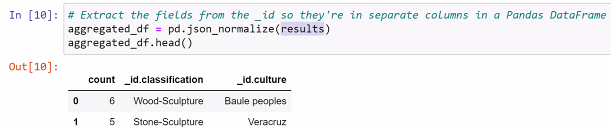


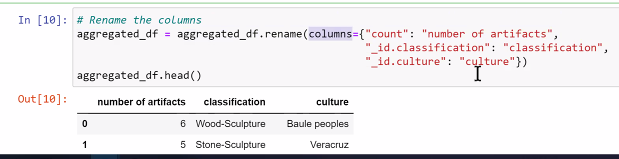
**Part 2!**

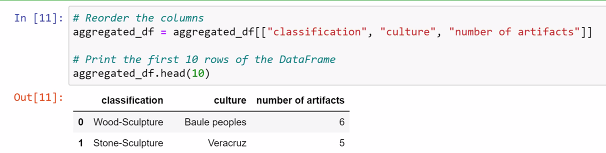












**Part 3!**





