MSDS610 Data Engineering

Week 5 Lab Supplemental – Part 2

**This is a supplemental document that supports the MSDS610 Week 5 lab.**

**Created by Dr. Bob Mason on 9/27/2019 in support of Dr. Nate George’s Week 5 lab video.**

**Based on this youtube video by Dr. George (Nate), please follow the steps shown below.**

**This part 2 begins at 12 minutes into the video.**

<https://www.youtube.com/watch?v=tRruqJhjgFQ&feature=youtu.be>

**We are still using Jupyter Notebooks and had finished adding and testing data frame info df.info() to our python program.**

***Our program looks like the following at this point in time:***

*import pandas as pd*

*import requests as req*

*response = req.get('https://www.metaweather.com/api/location/2391279/')*

*days = response.json()['consolidated\_weather']*

*df = pd.io.json.json\_normalize(days[0])*

*for day in days[1:]:*

*df = df.append(pd.io.json.json\_normalize(day))*

*df.info()*

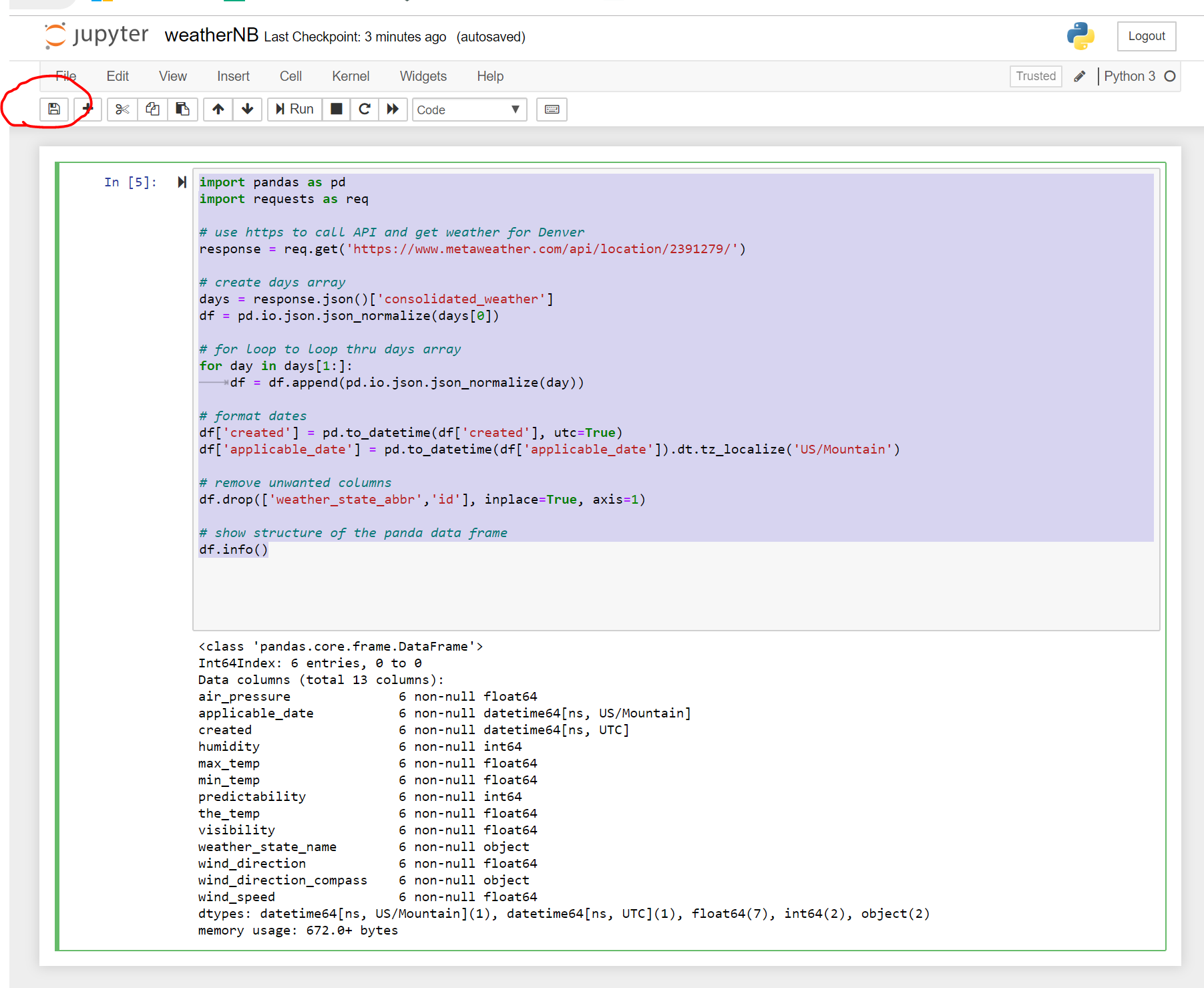


**Add three statements above the df.info() statement, as shown below to format dates and remove unwanted columns. Also, I added # comments and improved the format of the program with spaces before we go much further. Also, within Notebooks, you can click to save (disk symbol) the program for later use. A tip from an ‘old programmer’ is to save frequently.**

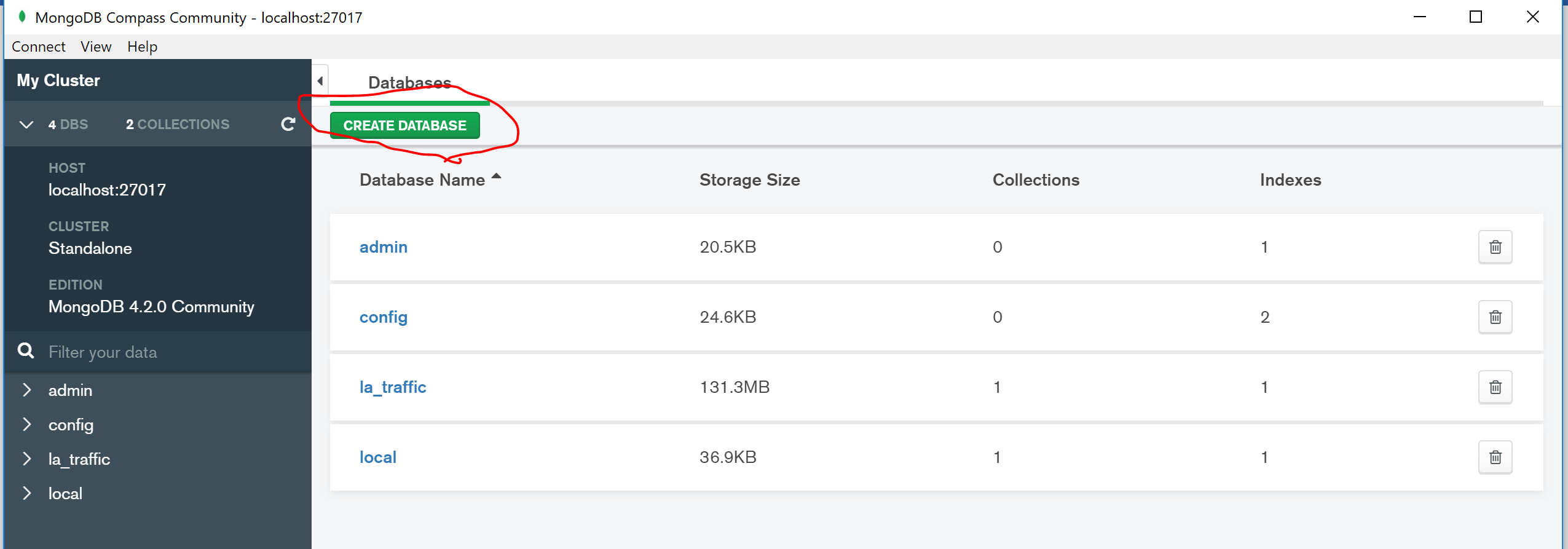
*df['created'] = pd.to\_datetime(df['created'], utc=True)*

*df['applicable\_date'] = pd.to\_datetime(df['applicable\_date']).dt.tz\_localize('US/Mountain')*

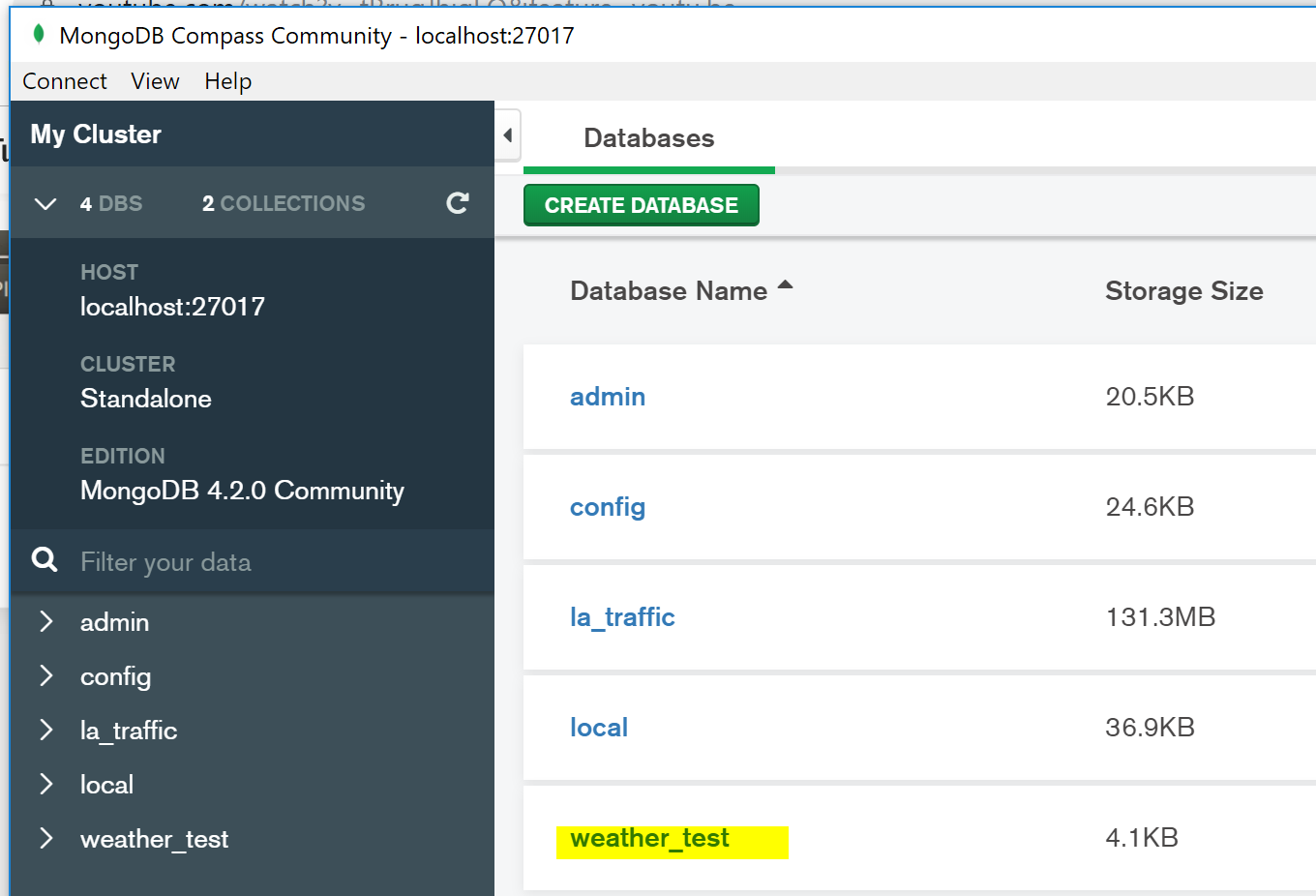
*df.drop(['weather\_state\_abbr','id'], inplace=True, axis=1)*



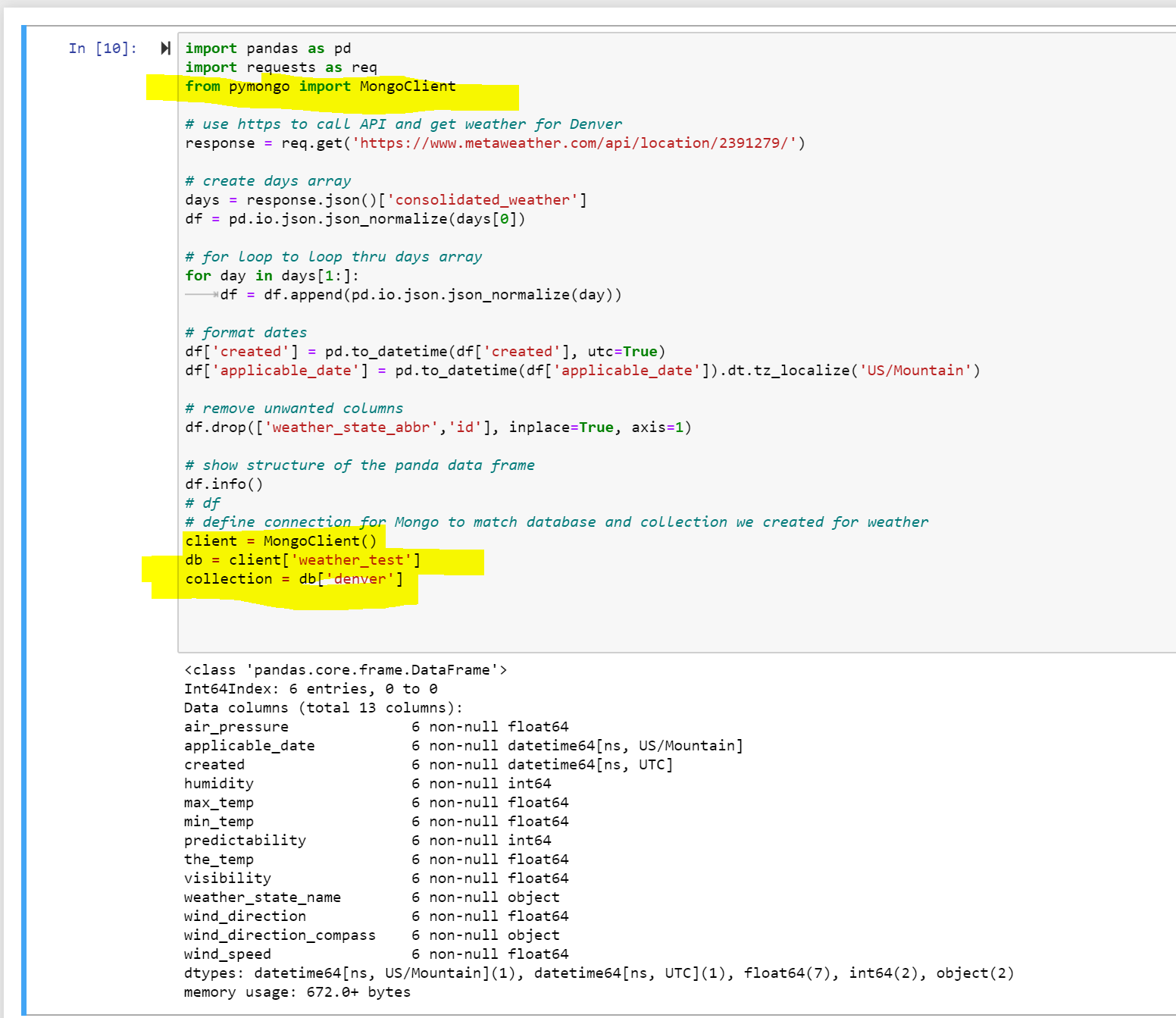
**Reference Lab 4 to refresh your memory about MongoDB Compass (if you installed on Windows). We are going to create a new database for the weather data.**



**After we create the new database, it should display in the list (below).**



**I added the import for the MongoClient at the top. Then, I added in the connection information to match the database and collection I created above using MongoDB.**



**After adding the code to insert into MongoDB, the current code is as follows:**

*import pandas as pd*

*import requests as req*

*from pymongo import MongoClient*

*# use https to call API and get weather for Denver*

*response = req.get('https://www.metaweather.com/api/location/2391279/')*

*# create days array and data frame*

*days = response.json()['consolidated\_weather']*

*df = pd.io.json.json\_normalize(days[0])*

*# for loop to loop thru days array*

*for day in days[1:]:*

*df = df.append(pd.io.json.json\_normalize(day))*

*# format dates*

*df['created'] = pd.to\_datetime(df['created'], utc=True)*

*df['applicable\_date'] = pd.to\_datetime(df['applicable\_date']).dt.tz\_localize('US/Mountain')*

*# remove unwanted columns*

*df.drop(['weather\_state\_abbr','id'], inplace=True, axis=1)*

*# show structure of the panda data frame*

*# df.info() -- I commented this out for later use*

*# df -- I commented this out for later use*

*# define connection for Mongo to match database and collection we created for weather*

*client = MongoClient()*

*db = client['weather\_test']*

*collection = db['denver']*

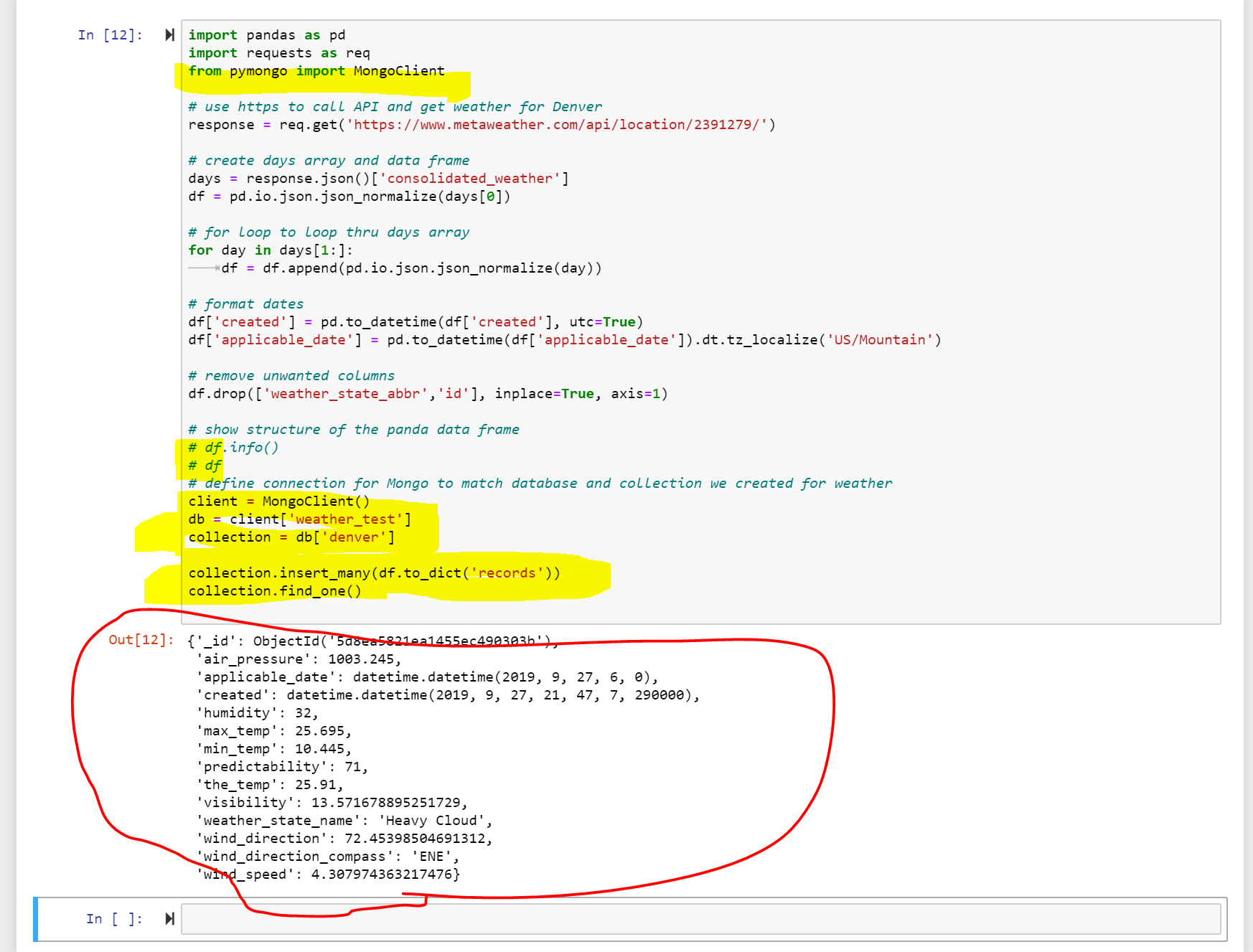
*# Insert data into our collection*

*collection.insert\_many(df.to\_dict('records'))*

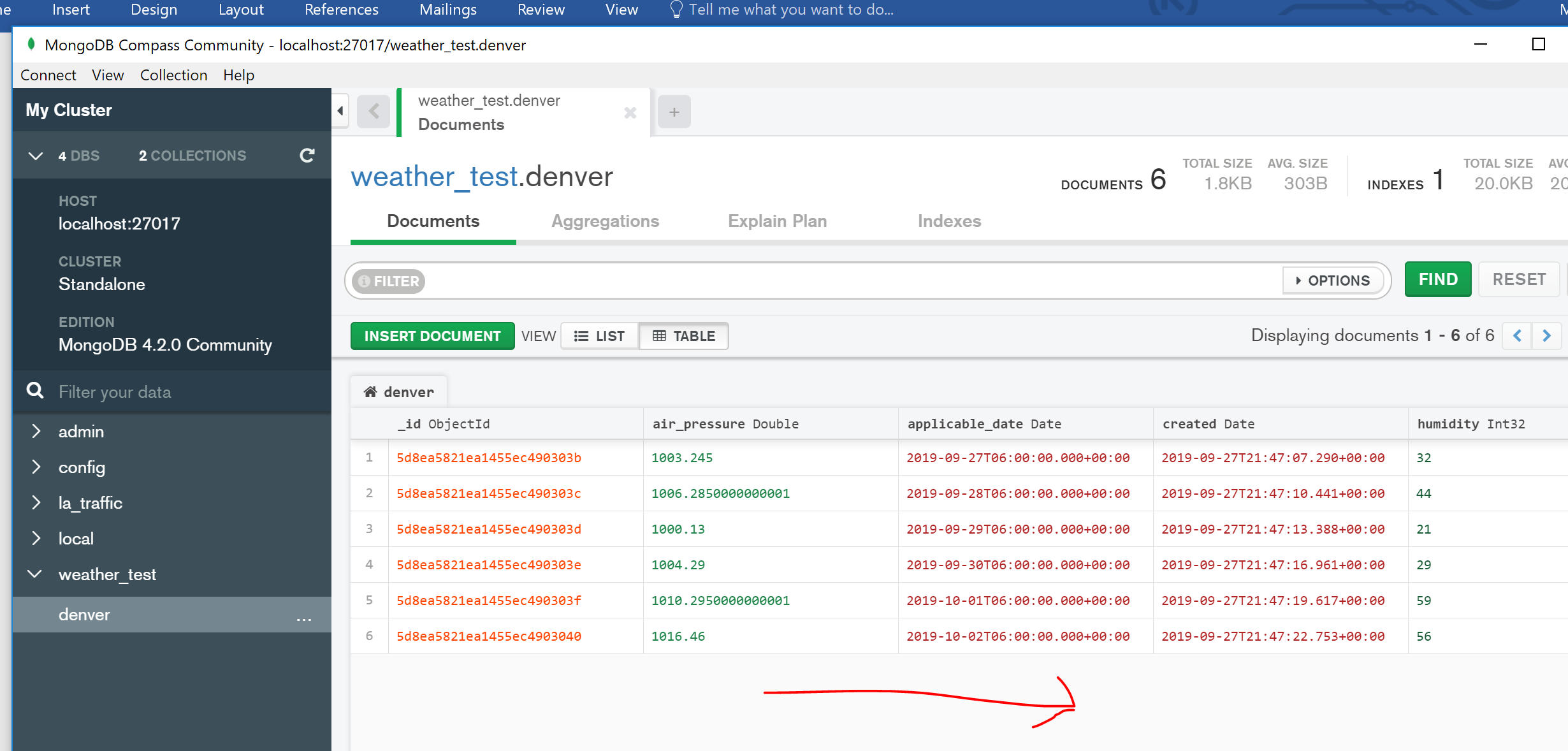
*# show an example*

*collection.find\_one()*

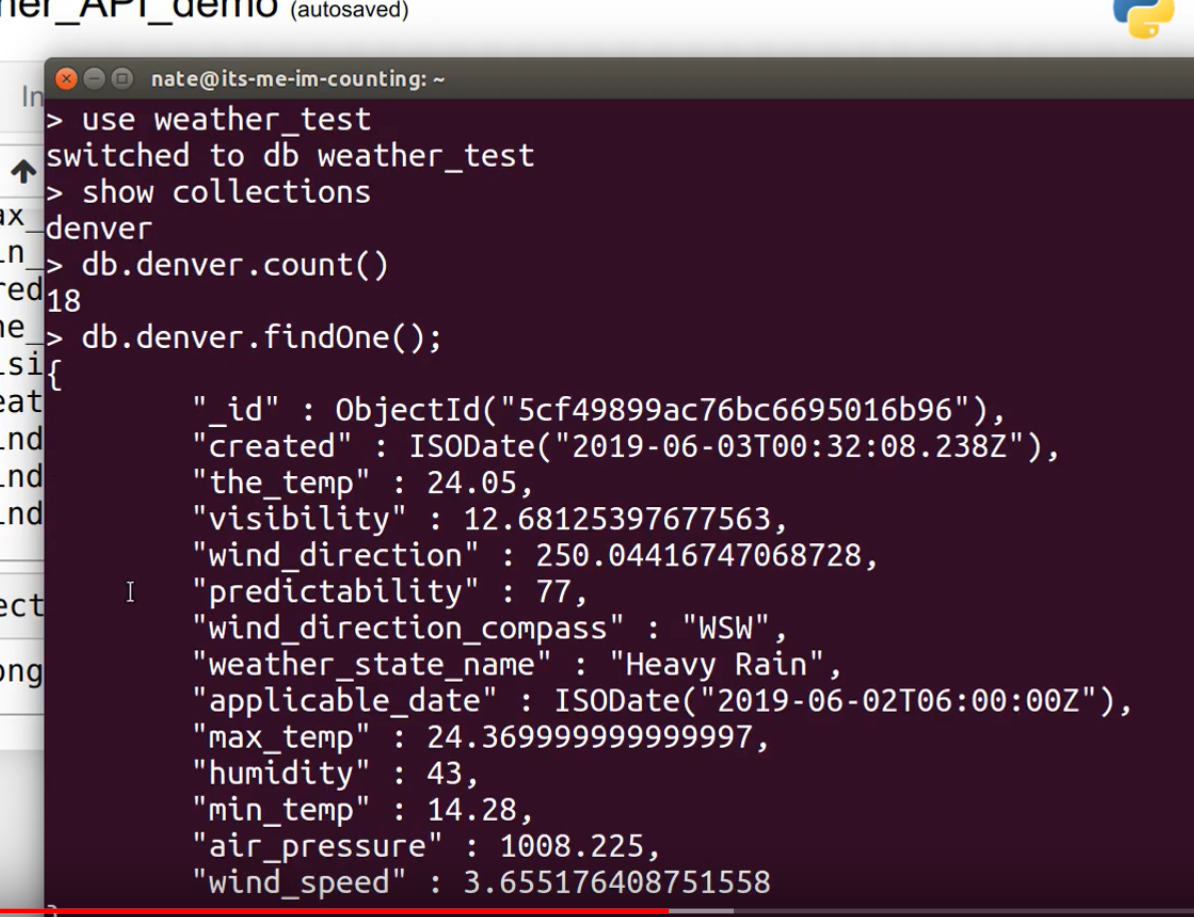
**Notice that I commented out df and df.info() to save for later in case I need to debug it. The find\_one() shows me a sample of the data I inserted into the collection. I only ran this one time, otherwise I’d have inserted duplicates. This is a good time to save your program.**



**I can easily use compass in Table view to check the data that I added. Or if I didn’t use Windows and Compass, I can write a query as shown below from Nate’s demo.**



**If you use the command line to verify, this is what it could look like. This is from Nate’s demo.**



**This is the last required portion of lab 5. There are optional portions of the lab, if you wish to experiment with Cron jobs, etc.**

**This is end of Lab 5 Part 2.**