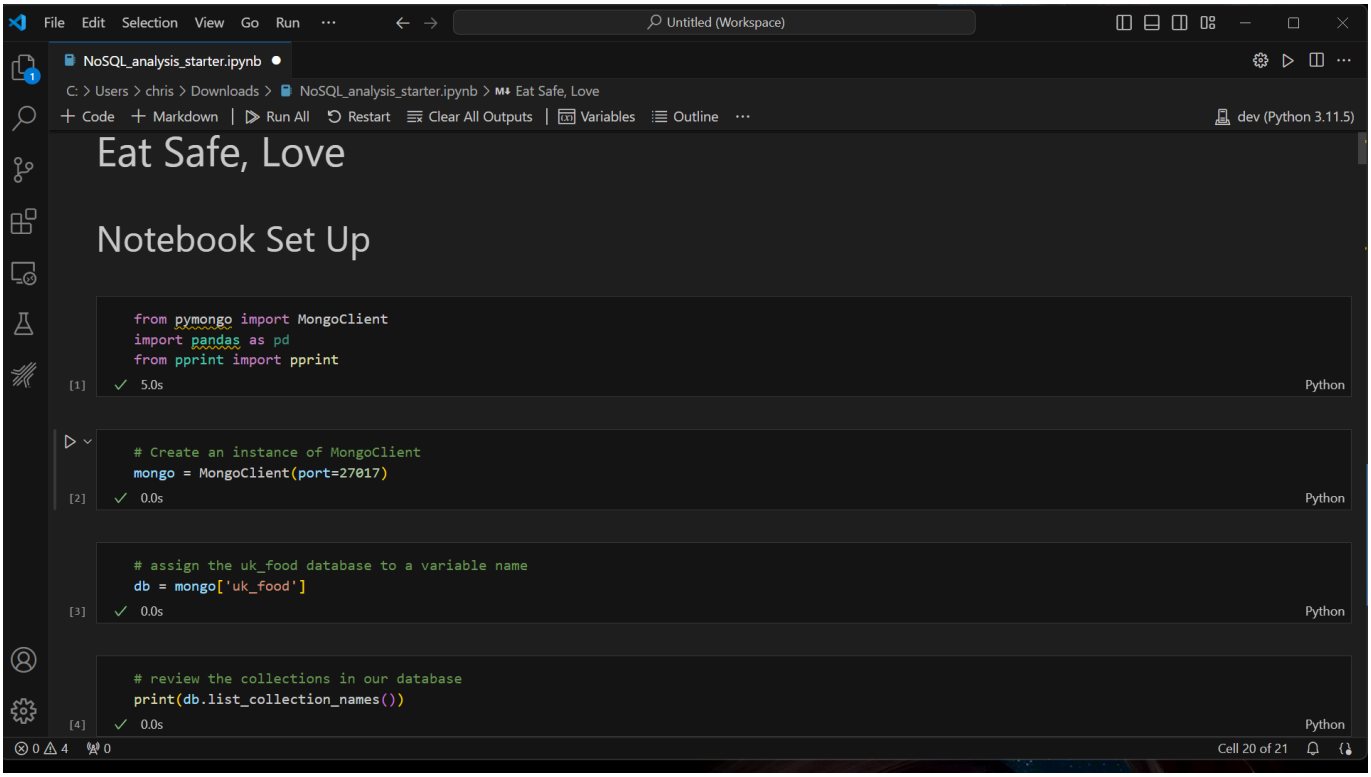
May 23, 2024

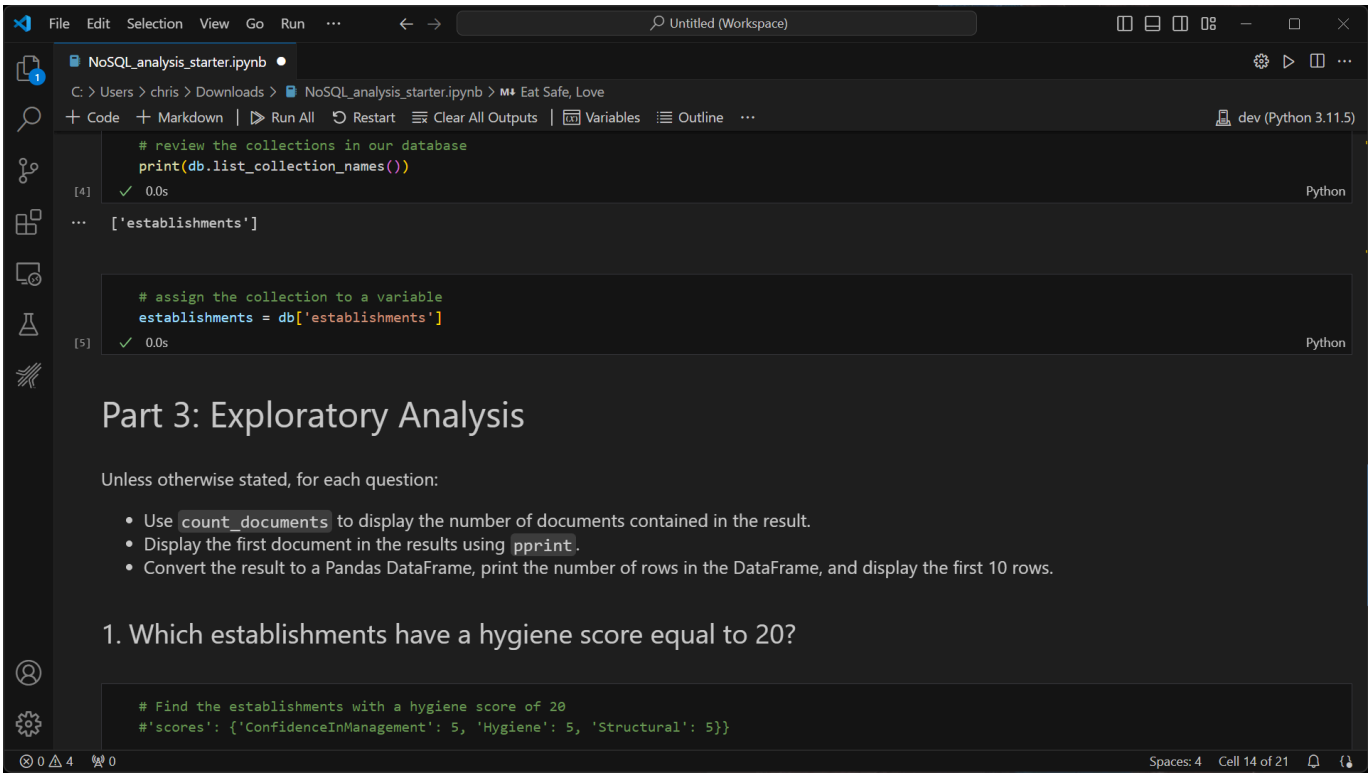
Dear grader,
Thank you for reviewing my Module 12 Challenge (nosql-challenge). Last night during office hours after class, I worked with my instructor and teaching assistant to attempt to address the issue that prevents me from using Mongosh to import data (and which causes me to have to use MongoDB Compass).

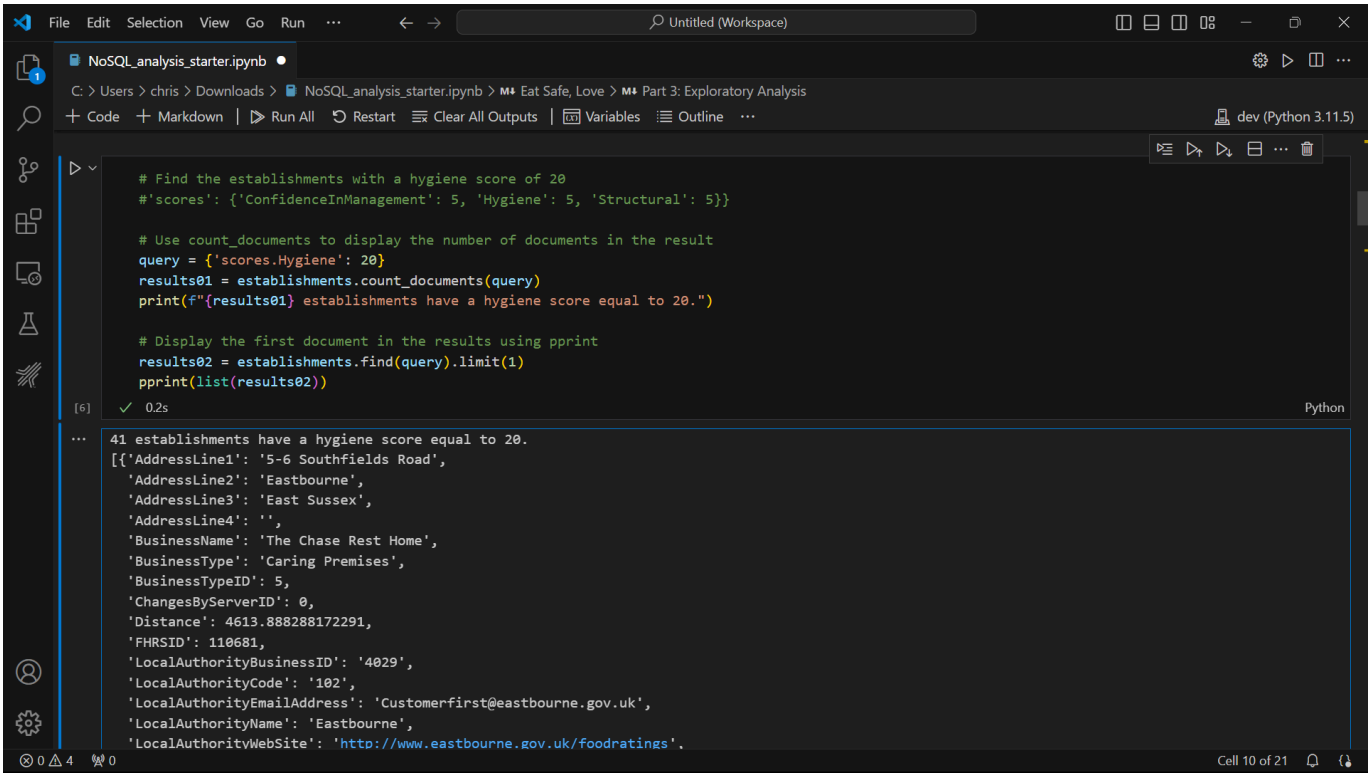
We pulled up my analysis notebook in VS Code and were able to reproduce the same outputs I had when I first completed the challenge.

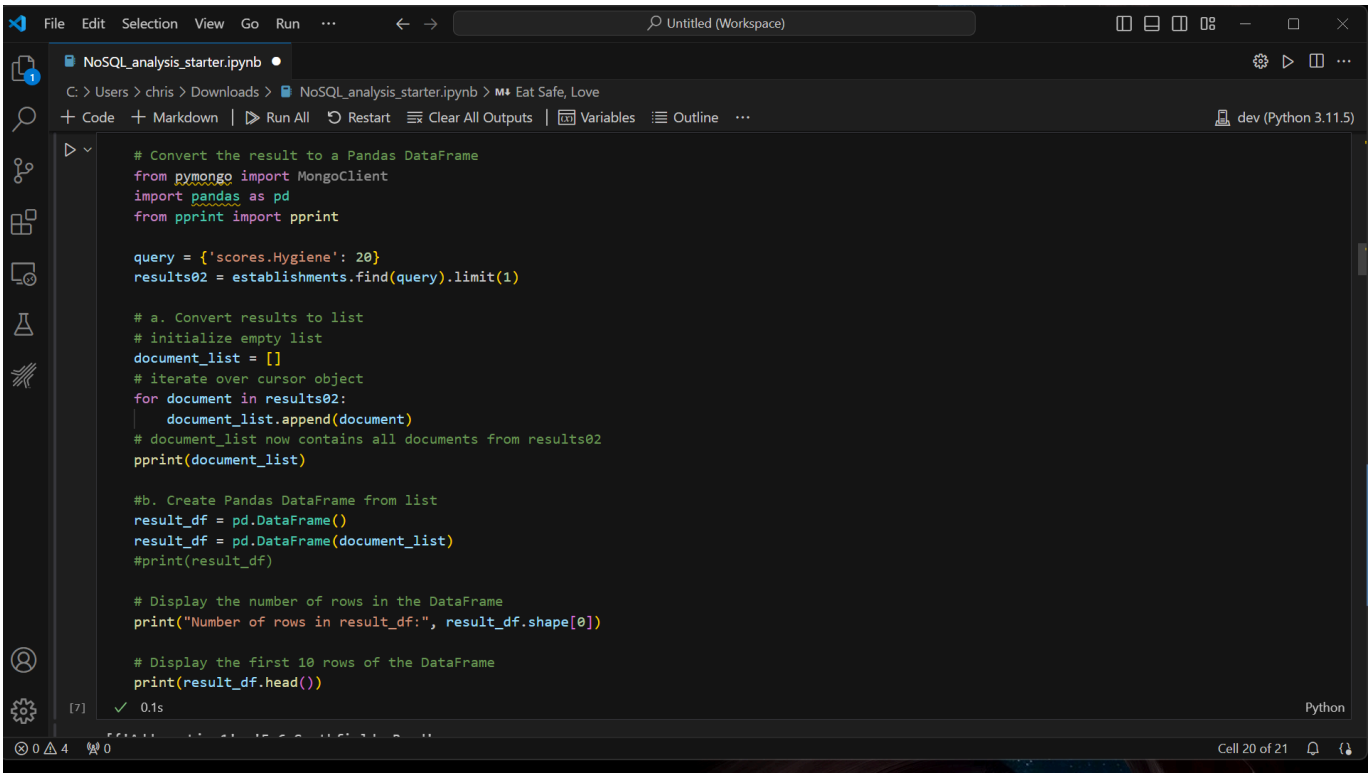
Both my instructor and teaching assistant assured me that not being able to run my notebook and receive outputs is unrelated to how I imported the data. They advised me not to spend time figuring out how to remedy the Mongosh issue (which I do have installed but can't use since it won't remain in my environment variables) and instead resubmit my github with a note and screenshots of the analysis output included.

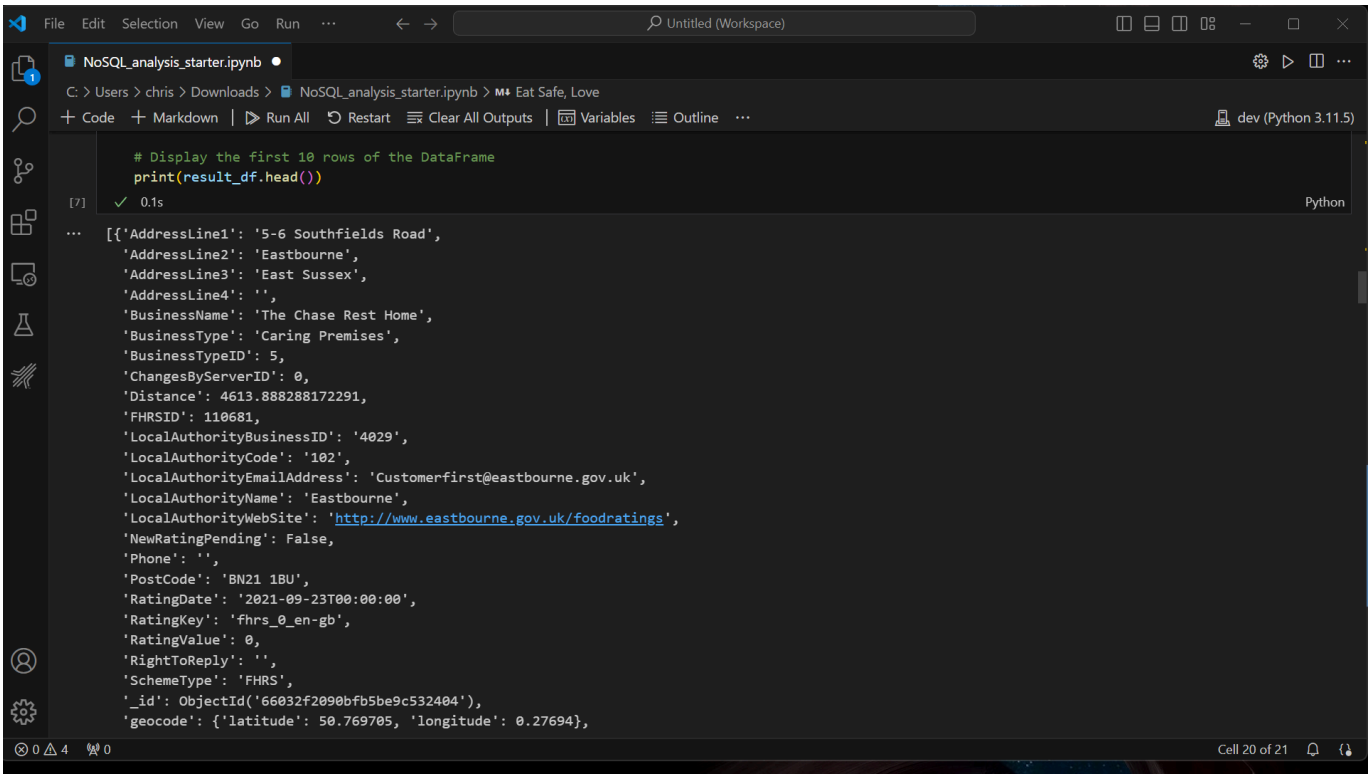
Would you please review these screenshots and revise my grade for this challenge?
Thank you very much,
Christine Jauregui

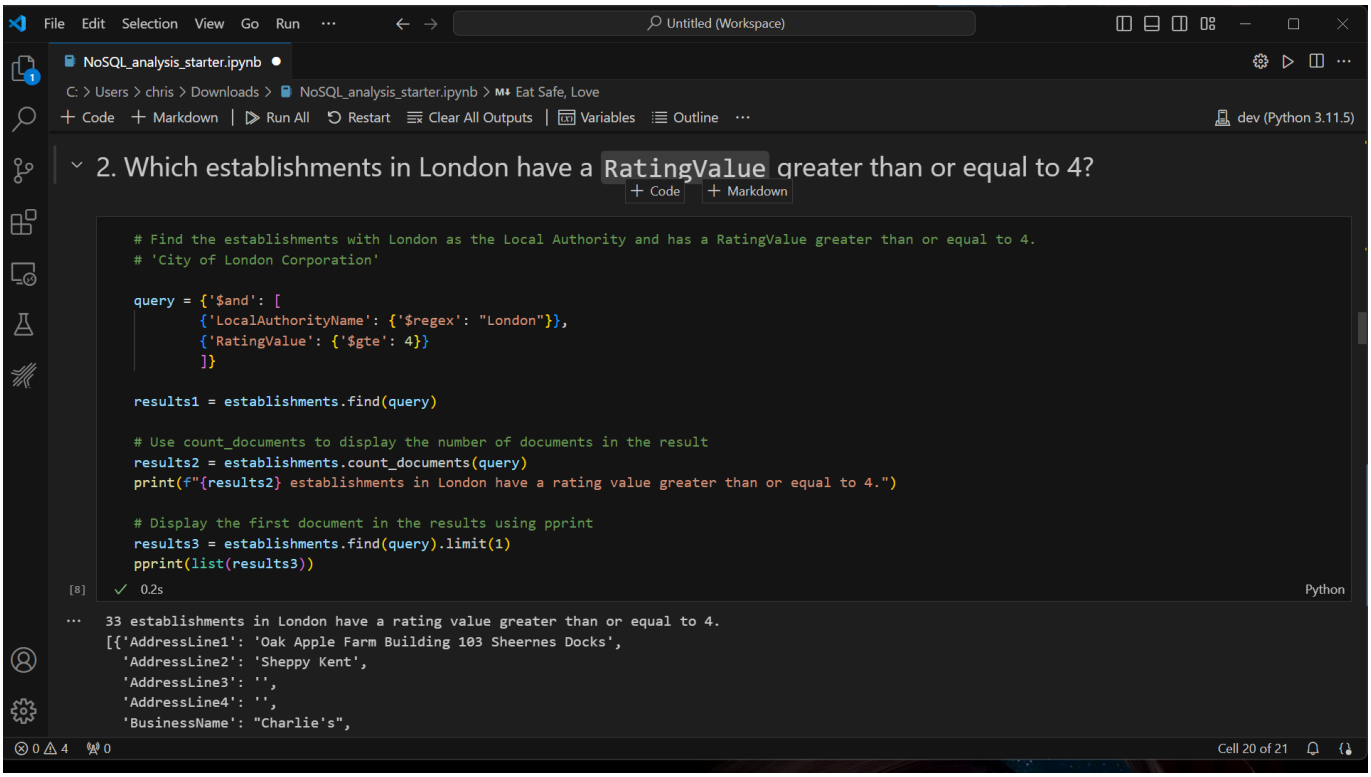


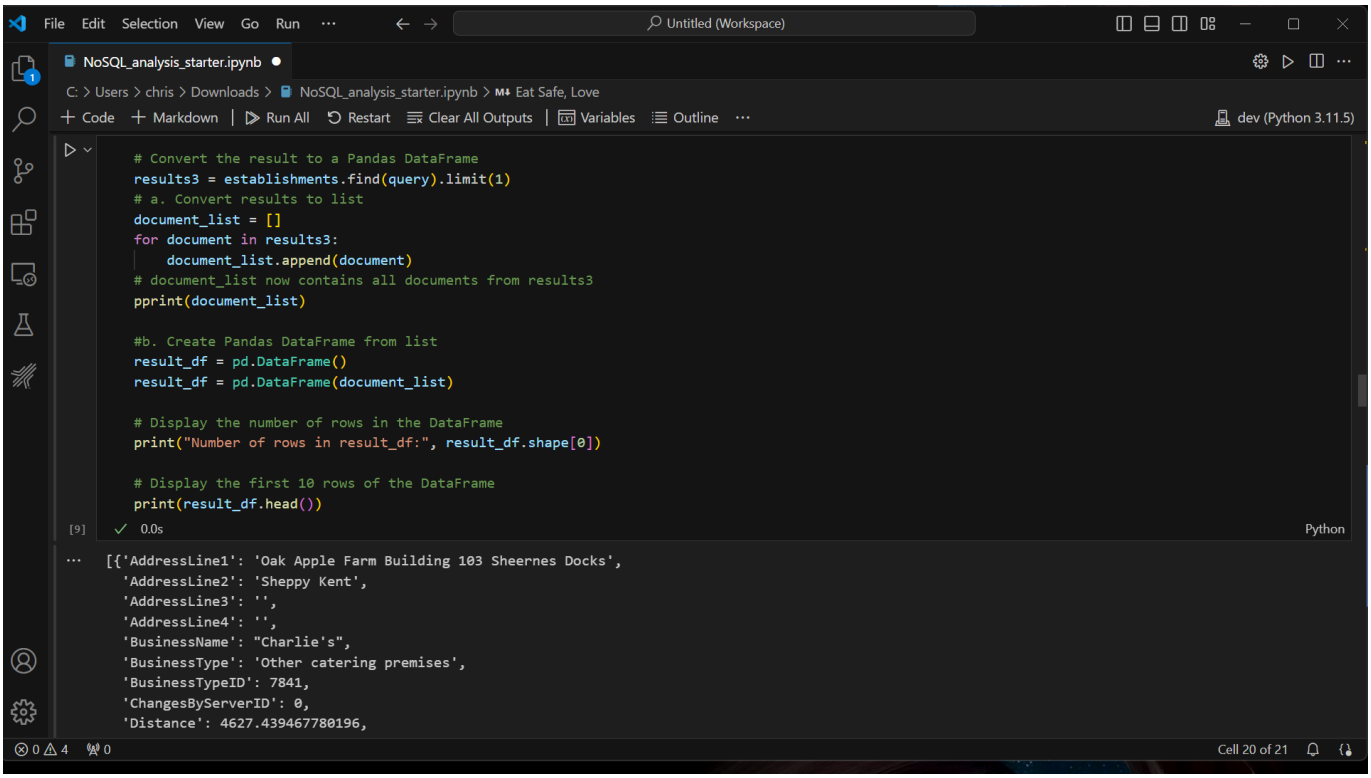

File Edit Selection View Go Run
Untitled (Workspace)
NoSQL_analysis_starter.ipynb
C: > Users > chris > Downloads > NoSQL_analysis_starter.ipynb > Eat Safe, Love
Code Markdown Run All Restart Clear All Outputs Variables Outline
dev (Python 3.11.5)
# review the collections in our database
print(db.list_collection_names())
[4] 0.0s Python
['establishments']
# assign the collection to a variable
establishments = db['establishments']
[5] 0.0s Python
Part 3: Exploratory Analysis
Unless otherwise stated, for each question:
Use count_documents to display the number of documents contained in the result.
Display the first document in the results using pprint.
Convert the result to a Pandas DataFrame, print the number of rows in the DataFrame, and display the first 10 rows.
1. Which establishments have a hygiene score equal to 20?
Find the establishments with a hygiene score of 20
#'scores': {'ConfidenceInManagement': 5, 'Hygiene': 5, 'Structural': 5}}
Spaces: 4 Cell 14 of 21

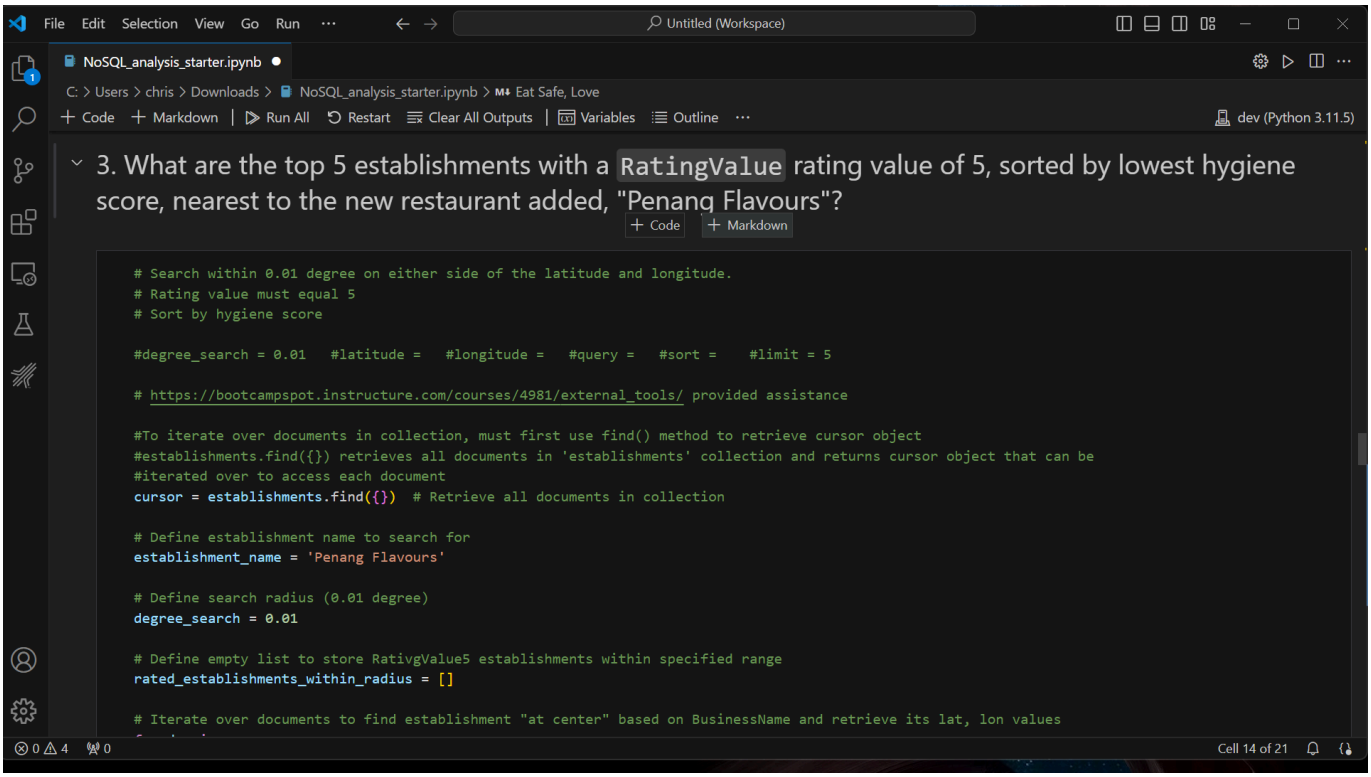

File Edit Selection View Go Run
Untitled (Workspace)
NoSQL_analysis_starter.ipynb
C: > Users > chris > Downloads > NoSQL_analysis_starter.ipynb > Eat Safe, Love > Part 3: Exploratory Analysis
Code Markdown Run All Restart Clear All Outputs Variables Outline
dev (Python 3.11.5)
Find the establishments with a hygiene score of 20
#'scores': {'ConfidenceInManagement': 5, 'Hygiene': 5, 'Structural': 5}}
# Use count_documents to display the number of documents in the result
query = {'scores.Hygiene': 20}
results01 = establishments.count_documents(query)
print(f"{results01} establishments have a hygiene score equal to 20.")
# Display the first document in the results using pprint
results02 = establishments.find(query).limit(1)
pprint(list(results02))
[6] 0.2s Python
41 establishments have a hygiene score equal to 20.
[{'AddressLine1': '5-6 Southfields Road',
'AddressLine2': 'Eastbourne',
'AddressLine3': 'East Sussex',
'AddressLine4': '',
'BusinessName': 'The Chase Rest Home',
'BusinessType': 'Caring Premises',
'BusinessTypeID': 5,
'ChangesByServerID': 0,
'Distance': 4613.888288172291,
'FHRSID': 110681,
'LocalAuthorityBusinessID': '4029',
'LocalAuthorityCode': '102',
'LocalAuthorityEmailAddress': 'Customerfirst@eastbourne.gov.uk',
'LocalAuthorityName': 'Eastbourne',
'LocalAuthorityWebSite': 'http://www.eastbourne.gov.uk/foodratings',
Cell 10 of 21

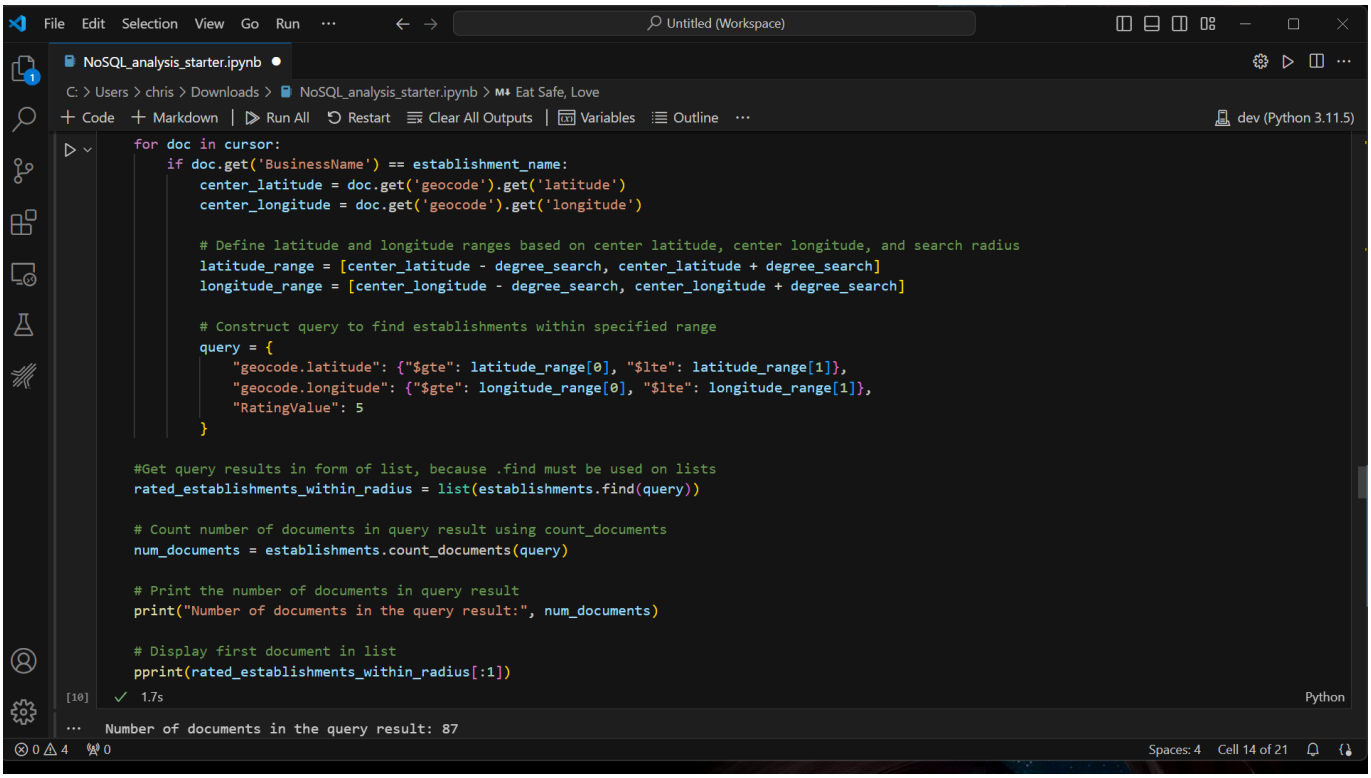

File Edit Selection View Go Run
Untitled (Workspace)
NoSQL_analysis_starter.ipynb
C: > Users > chris > Downloads > NoSQL_analysis_starter.ipynb > Eat Safe, Love
Code Markdown Run All Restart Clear All Outputs Variables Outline
dev (Python 3.11.5)
# Convert the result to a Pandas DataFrame
from pymongo import MongoClient
import pandas as pd
from pprint import pprint
query = {'scores.Hygiene': 20}
results02 = establishments.find(query).limit(1)
# a. Convert results to list
# initialize empty list
document_list = []
# iterate over cursor object
for document in results02:
document_list.append(document)
# document_list now contains all documents from results02
pprint(document_list)
#b. Create Pandas DataFrame from list
result_df = pd.DataFrame()
result_df = pd.DataFrame(document_list)
#print(result_df)
# Display the number of rows in the DataFrame
print("Number of rows in result_df:", result_df.shape[0])
# Display the first 10 rows of the DataFrame
print(result_df.head())
[7] 0.1s
Python
Cell 20 of 21

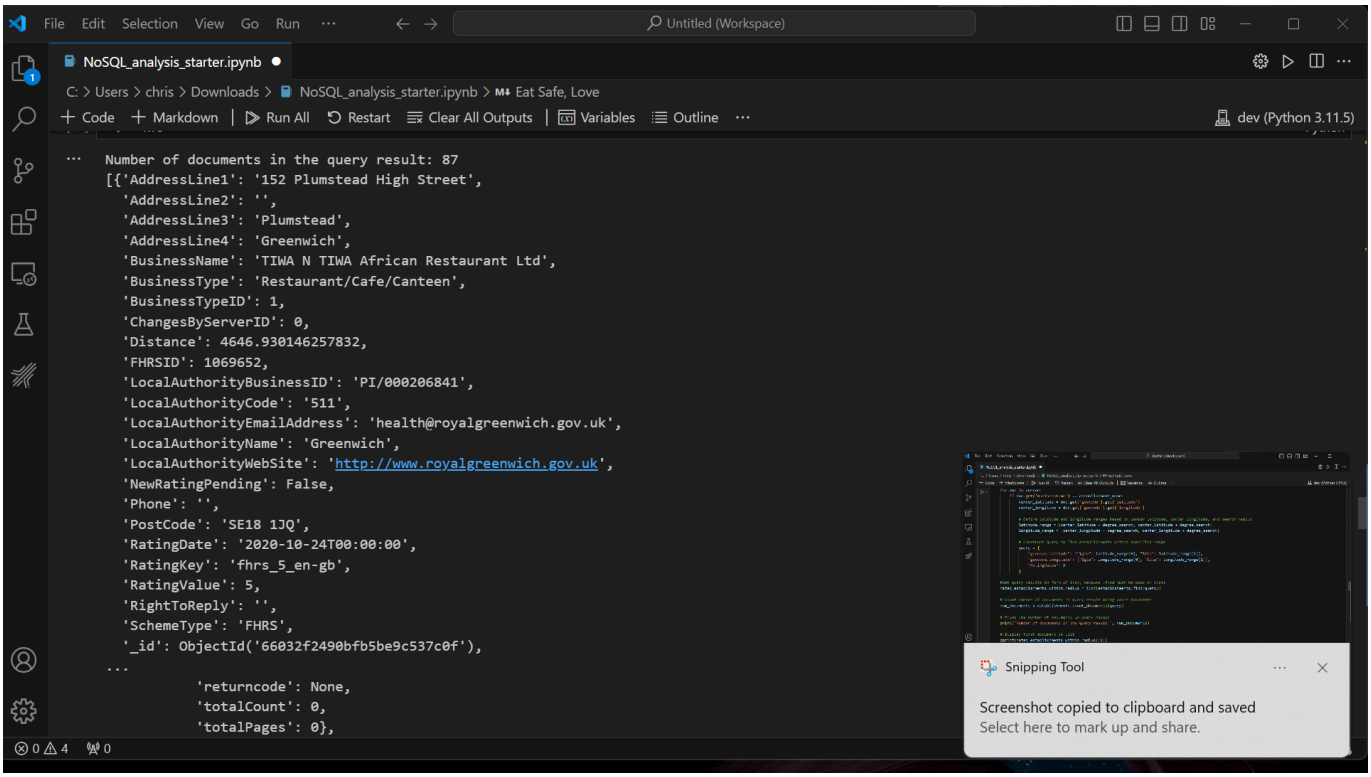

File Edit Selection View Go Run
Untitled (Workspace)
NoSQL_analysis_starter.ipynb
C: > Users > chris > Downloads > NoSQL_analysis_starter.ipynb > Eat Safe, Love
Code Markdown Run All Restart Clear All Outputs Variables Outline
dev (Python 3.11.5)
# Display the first 10 rows of the DataFrame
print(result_df.head())
[7] 0.1s
Python
[{'AddressLine1': '5-6 Southfields Road',
'AddressLine2': 'Eastbourne',
'AddressLine3': 'East Sussex',
'AddressLine4': '',
'BusinessName': 'The Chase Rest Home',
'BusinessType': 'Caring Premises',
'BusinessTypeID': 5,
'ChangesByServerID': 0,
'Distance': 4613.888288172291,
'FHRSID': 110681,
'LocalAuthorityBusinessID': '4029',
'LocalAuthorityCode': '102',
'LocalAuthorityEmailAddress': 'Customerfirst@eastbourne.gov.uk',
'LocalAuthorityName': 'Eastbourne',
'LocalAuthorityWebSite': 'http://www.eastbourne.gov.uk/foodratings',
'NewRatingPending': False,
'Phone': '',
'PostCode': 'BN21 1BU',
'RatingDate': '2021-09-23T00:00:00',
'RatingKey': 'fhrs_0_en-gb',
'RatingValue': 0,
'RightToReply': '',
'SchemeType': 'FHRS',
'_id': ObjectId('66032f2090bfb5be9c532404'),
'geocode': {'latitude': 50.769705, 'longitude': 0.27694},
Cell 20 of 21

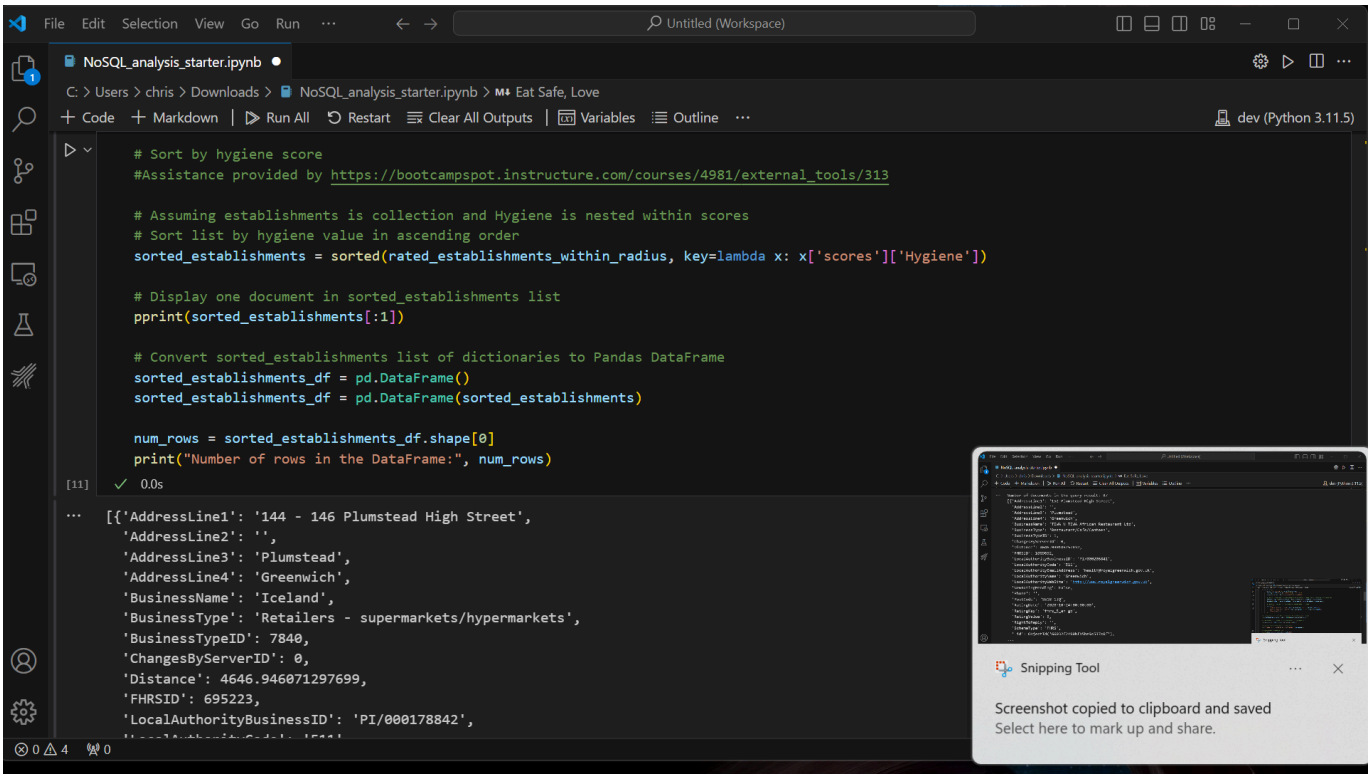

File Edit Selection View Go Run
Untitled (Workspace)
NoSQL_analysis_starter.ipynb
C: > Users > chris > Downloads > NoSQL_analysis_starter.ipynb > Eat Safe, Love
Code Markdown Run All Restart Clear All Outputs Variables Outline
dev (Python 3.11.5)
2. Which establishments in London have a RatingValue greater than or equal to 4?
Code Markdown
# Find the establishments with London as the Local Authority and has a RatingValue greater than or equal to 4.
# 'City of London Corporation'
query = {'$and': [
{'LocalAuthorityName': {'$regex': "London"}},
{'RatingValue': {'$gte': 4}}
]}
results1 = establishments.find(query)
# Use count_documents to display the number of documents in the result
results2 = establishments.count_documents(query)
print(f"{results2} establishments in London have a rating value greater than or equal to 4.")
# Display the first document in the results using pprint
results3 = establishments.find(query).limit(1)
pprint(list(results3))
[8] 0.2s
Python
33 establishments in London have a rating value greater than or equal to 4.
[{'AddressLine1': 'Oak Apple Farm Building 103 Sheernes Docks',
'AddressLine2': 'Sheppy Kent',
'AddressLine3': '',
'AddressLine4': '',
'BusinessName': "Charlie's",
0 4 0
Cell 20 of 21


File Edit Selection View Go Run
Untitled (Workspace)
NoSQL_analysis_starter.ipynb
C: > Users > chris > Downloads > NoSQL_analysis_starter.ipynb > Eat Safe, Love
Code Markdown Run All Restart Clear All Outputs Variables Outline
dev (Python 3.11.5)
# Convert the result to a Pandas DataFrame
results3 = establishments.find(query).limit(1)
# a. Convert results to list
document_list = []
for document in results3:
document_list.append(document)
# document_list now contains all documents from results3
pprint(document_list)
#b. Create Pandas DataFrame from list
result_df = pd.DataFrame()
result_df = pd.DataFrame(document_list)
# Display the number of rows in the DataFrame
print("Number of rows in result_df:", result_df.shape[0])
# Display the first 10 rows of the DataFrame
print(result_df.head())
[9] 0.0s
Python
[{'AddressLine1': 'Oak Apple Farm Building 103 Sheernes Docks',
'AddressLine2': 'Sheppy Kent',
'AddressLine3': '',
'AddressLine4': '',
'BusinessName': "Charlie's",
'BusinessType': 'Other catering premises',
'BusinessTypeID': 7841,
'ChangesByServerID': 0,
'Distance': 4627.439467780196,
0 4 0
Cell 20 of 21


File Edit Selection View Go Run
Untitled (Workspace)
NoSQL_analysis_starter.ipynb
C: > Users > chris > Downloads > NoSQL_analysis_starter.ipynb > Eat Safe, Love
Code Markdown Run All Restart Clear All Outputs Variables Outline
dev (Python 3.11.5)
3. What are the top 5 establishments with a RatingValue rating value of 5, sorted by lowest hygiene score, nearest to the new restaurant added, "Penang Flavours"?
Code Markdown
Search within 0.01 degree on either side of the latitude and longitude.
Rating value must equal 5
Sort by hygiene score
#degree_search = 0.01 #latitude = #longitude = #query = #sort = #limit = 5
https://bootcampspot.instructure.com/courses/4981/external_tools/ provided assistance
#To iterate over documents in collection, must first use find() method to retrieve cursor object
#establishments.find({}) retrieves all documents in 'establishments' collection and returns cursor object that can be
#iterated over to access each document
cursor = establishments.find({}) # Retrieve all documents in collection
# Define establishment name to search for
establishment_name = 'Penang Flavours'
# Define search radius (0.01 degree)
degree_search = 0.01
# Define empty list to store RativgValue5 establishments within specified range
rated_establishments_within_radius = []
# Iterate over documents to find establishment "at center" based on BusinessName and retrieve its lat, lon values
Cell 14 of 21


File Edit Selection View Go Run
Untitled (Workspace)
NoSQL_analysis_starter.ipynb
C: > Users > chris > Downloads > NoSQL_analysis_starter.ipynb > Eat Safe, Love
Code Markdown Run All Restart Clear All Outputs Variables Outline
dev (Python 3.11.5)
for doc in cursor:
if doc.get('BusinessName') == establishment_name:
center_latitude = doc.get('geocode').get('latitude')
center_longitude = doc.get('geocode').get('longitude')
# Define latitude and longitude ranges based on center latitude, center longitude, and search radius
latitude_range = [center_latitude - degree_search, center_latitude + degree_search]
longitude_range = [center_longitude - degree_search, center_longitude + degree_search]
# Construct query to find establishments within specified range
query = {
"geocode.latitude": {"$gte": latitude_range[0], "$lte": latitude_range[1]},
"geocode.longitude": {"$gte": longitude_range[0], "$lte": longitude_range[1]},
"RatingValue": 5
}
#Get query results in form of list, because .find must be used on lists
rated_establishments_within_radius = list(establishments.find(query))
# Count number of documents in query result using count_documents
num_documents = establishments.count_documents(query)
# Print the number of documents in query result
print("Number of documents in the query result:", num_documents)
# Display first document in list
pprint(rated_establishments_within_radius[:1])
[10] 1.7s
Python
Number of documents in the query result: 87
Spaces: 4 Cell 14 of 21


File Edit Selection View Go Run
Untitled (Workspace)
NoSQL_analysis_starter.ipynb
C: > Users > chris > Downloads > NoSQL_analysis_starter.ipynb > Eat Safe, Love
Code Markdown Run All Restart Clear All Outputs Variables Outline
dev (Python 3.11.5)
Number of documents in the query result: 87
[{'AddressLine1': '152 Plumstead High Street',
'AddressLine2': '',
'AddressLine3': 'Plumstead',
'AddressLine4': 'Greenwich',
'BusinessName': 'TIWA N TIWA African Restaurant Ltd',
'BusinessType': 'Restaurant/Cafe/Canteen',
'BusinessTypeID': 1,
'ChangesByServerID': 0,
'Distance': 4646.930146257832,
'FHRSID': 1069652,
'LocalAuthorityBusinessID': 'PI/000206841',
'LocalAuthorityCode': '511',
'LocalAuthorityEmailAddress': 'health@royalgreenwich.gov.uk',
'LocalAuthorityName': 'Greenwich',
'LocalAuthorityWebSite': 'http://www.royalgreenwich.gov.uk',
'NewRatingPending': False,
'Phone': '',
'PostCode': 'SE18 1JQ',
'RatingDate': '2020-10-24T00:00:00',
'RatingKey': 'fhrs_5_en-gb',
'RatingValue': 5,
'RightToReply': '',
'SchemeType': 'FHRS',
'_id': ObjectId('66032f2490bfb5be9c537c0f'),
'returncode': None,
'totalCount': 0,
'totalPages': 0},
Snipping Tool
Screenshot copied to clipboard and saved
Select here to mark up and share.


File Edit Selection View Go Run
Untitled (Workspace)
NoSQL_analysis_starter.ipynb
C: > Users > chris > Downloads > NoSQL_analysis_starter.ipynb > Eat Safe, Love
Code Markdown Run All Restart Clear All Outputs Variables Outline
dev (Python 3.11.5)
Sort by hygiene score
#Assistance provided by https://bootcampspot.instructure.com/courses/4981/external_tools/313
# Assuming establishments is collection and Hygiene is nested within scores
# Sort list by hygiene value in ascending order
sorted_establishments = sorted(rated_establishments_within_radius, key=lambda x: x['scores']['Hygiene'])
# Display one document in sorted_establishments list
pprint(sorted_establishments[:1])
# Convert sorted_establishments list of dictionaries to Pandas DataFrame
sorted_establishments_df = pd.DataFrame()
sorted_establishments_df = pd.DataFrame(sorted_establishments)
num_rows = sorted_establishments_df.shape[0]
print("Number of rows in the DataFrame:", num_rows)
[11] 0.0s
[{'AddressLine1': '144 - 146 Plumstead High Street',
'AddressLine2': '',
'AddressLine3': 'Plumstead',
'AddressLine4': 'Greenwich',
'BusinessName': 'Iceland',
'BusinessType': 'Retailers - supermarkets/hypermarkets',
'BusinessTypeID': 7840,
'ChangesByServerID': 0,
'Distance': 4646.946071297699,
'FHRSID': 695223,
'LocalAuthorityBusinessID': 'PI/000178842',
Snipping Tool
Screenshot copied to clipboard and saved
Select here to mark up and share.

File Edit Selection View Go Run ··· Untitled (Workspace)

NoSQL_analysis_starter.ipynb

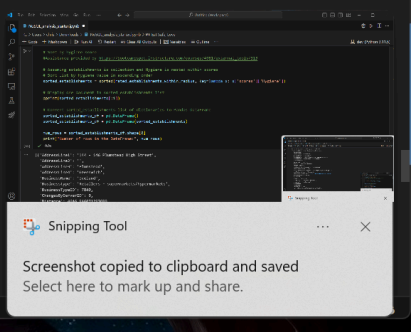
C: > Users > chris > Downloads > NoSQL_analysis_starter.ipynb > Eat Safe, Love

+ Code + Markdown | Run All Restart Clear All Outputs | Variables Outline ··· dev (Python 3.11.5)

```
# Print first 10 rows of data frame
pprint(sorted_establishments_df.head(10))
```

[12] ✓ 0.0s Python

```
                        _id   FHRSID  ChangesByServerID  \
0  66032f2490bfb5be9c537c36   695223                  0
1  66032f2490bfb5be9c537c46  1380578                  0
2  66032f2490bfb5be9c537c63   694609                  0
3  66032f2490bfb5be9c537c7b   695241                  0
4  66032f2490bfb5be9c537c7e   694478                  0
5  66032f2490bfb5be9c537c80   909912                  0
6  66032f2490bfb5be9c537c9e   694482                  0
7  66032f2490bfb5be9c537c9f   695204                  0
8  66032f2490bfb5be9c537cc6  1211907                  0
9  66032f2490bfb5be9c537cc8   940026                  0

  LocalAuthorityBusinessID                      BusinessName  \
0            PI/000178842                           Iceland
1                  14425  Howe and Co Fish and Chips - Van 17
2            PI/000116619                         Volunteer
3            PI/000179088             Plumstead Manor Nursery
4            PI/000086506                  Atlantic Fish Bar
5            PI/000201100     Lumbini Grocery Ltd T/A Al-Iman
6            PI/000086551                            Greggs
7            PI/000177985                             Tesco
8                   13217               Dosa & Sambal Express
9            PI/000177403                        Abbi Wines

                        BusinessType  BusinessTypeID  \
```



⊗ 0 ⚠ 4 0

File Edit Selection View Go Run ··· Untitled (Workspace)

NoSQL_analysis_starter.ipynb

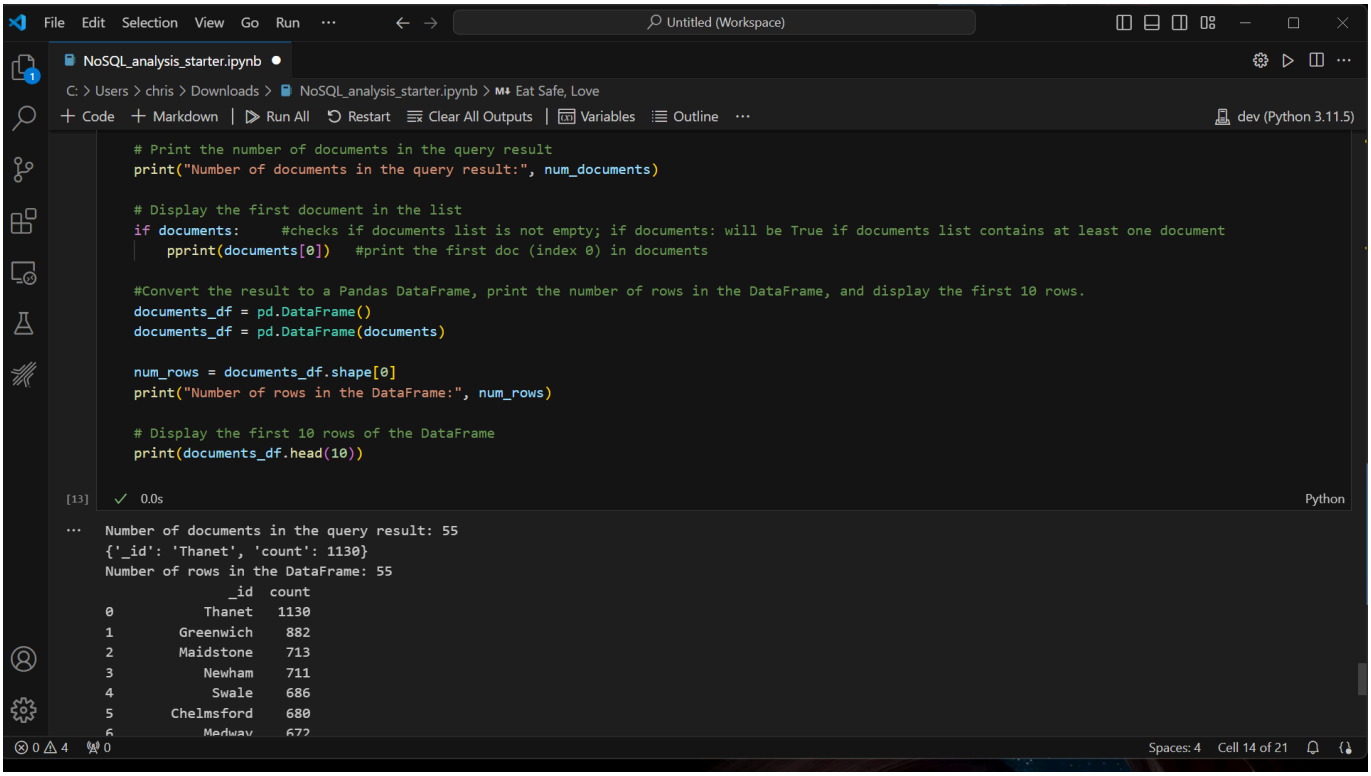
C: > Users > chris > Downloads > NoSQL_analysis_starter.ipynb > Eat Safe, Love

+ Code + Markdown | Run All Restart Clear All Outputs | Variables Outline ··· dev (Python 3.11.5)

4. How many establishments in each Local Authority area have a hygiene score of 0?

```
# Create a pipeline that:
# 1. Matches establishments with a hygiene score of 0
# 2. Groups the matches by Local Authority
# 3. Sorts the matches from highest to lowest

# https://bootcampspot.instructure.com/courses/4981/external_tools/ provided assistance
# Write match query to find only documents with hygiene score of 0
match_query = {'$match': {'scores.Hygiene': 0}}

#Output of previous query serves as input for this next query in aggregation pipeline
group_query = {'$group': {'_id': "$LocalAuthorityName", 'count': { '$sum': 1 }}}

# Create dictionary that will allow pipeline to sort by LocalAuthorityName count in descending order
sort_values = {'$sort': { 'count': -1}}

# Put pipeline together
pipeline = [match_query, group_query, sort_values]
result = establishments.aggregate(pipeline)

# Iterate over the cursor to get the actual documents
documents = list(result)

# Count the number of documents in the query result
num_documents = len(documents)
```

⊗ 0 ⚠ 4 0 Spaces: 4 Cell 14 of 21


File Edit Selection View Go Run
Untitled (Workspace)
NoSQL_analysis_starter.ipynb
C: > Users > chris > Downloads > NoSQL_analysis_starter.ipynb > Eat Safe, Love
Code Markdown Run All Restart Clear All Outputs Variables Outline
dev (Python 3.11.5)
# Print the number of documents in the query result
print("Number of documents in the query result:", num_documents)
# Display the first document in the list
if documents: #checks if documents list is not empty; if documents: will be True if documents list contains at least one document
pprint(documents[0]) #print the first doc (index 0) in documents
#Convert the result to a Pandas DataFrame, print the number of rows in the DataFrame, and display the first 10 rows.
documents_df = pd.DataFrame()
documents_df = pd.DataFrame(documents)
num_rows = documents_df.shape[0]
print("Number of rows in the DataFrame:", num_rows)
# Display the first 10 rows of the DataFrame
print(documents_df.head(10))
[13] 0.0s
Python
Number of documents in the query result: 55
{'_id': 'Thanet', 'count': 1130}
Number of rows in the DataFrame: 55
_id count
0 Thanet 1130
1 Greenwich 882
2 Maidstone 713
3 Newham 711
4 Swale 686
5 Chelmsford 680
Spaces: 4 Cell 14 of 21