Homework Problem Set J Submission Form

# Overview

| Your Name | Nick Videtti |
| --- | --- |
| Your SU Email | nvidetti@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 screenshot guidelines from the first homework.

Remember as you complete the homework that 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 that 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.

**This assignment helped me to better understand the REST API (curl commands/query strings), Kibana index patterns, and Kibana data viz.**

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

**Spent many hours to get started due to trying to figure out question 1 since querying datetime fields in a curl query string was not covered in the asynchronous material to my knowledge. I tried my best for question 1 but was unable to get accurate results. I also had a tough time figuring out how to create the geo\_point field in Kibana, but was able to find how to do it while reviewing the asynchronous material.**

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

**I think I was about as prepared as you could be. The asynchronous does not always prepare me for these assignments, but they do give a foundation that makes the challenge of these assignments possible. I would say going over the asynchronous material again would help me be better prepared, but I feel that happens naturally simply by completing this assignment, as well as completing quizzes.**

1. 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.

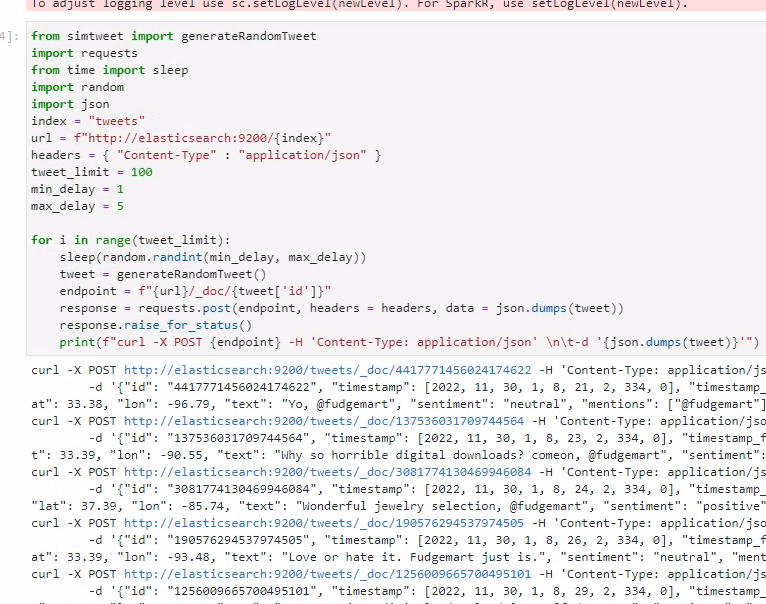
# Part II: Questions

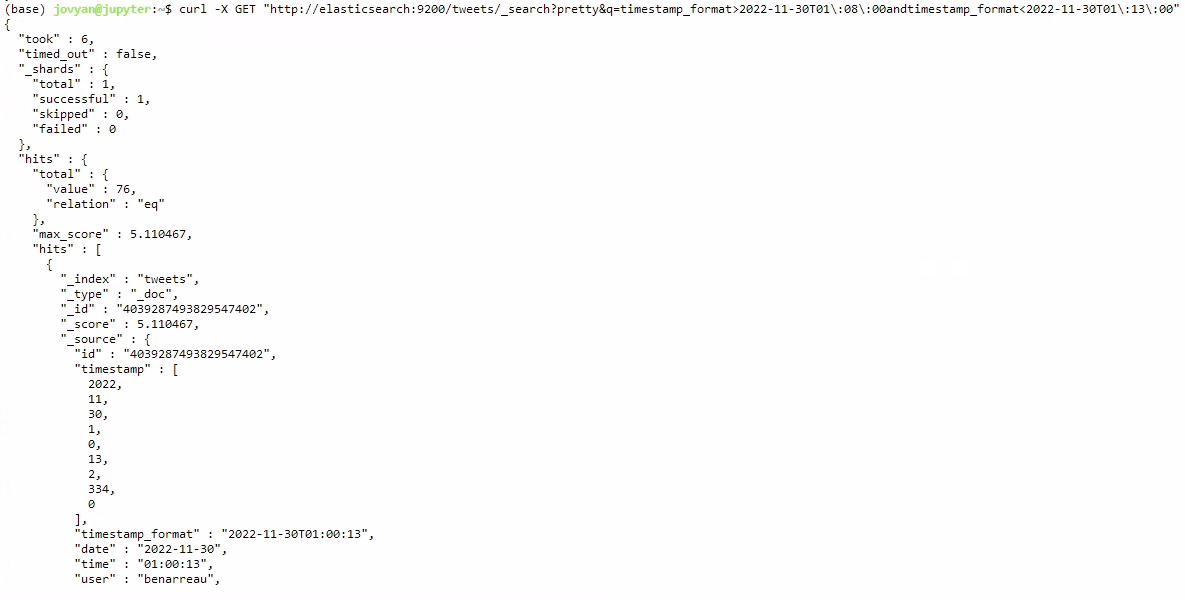
**For each question, include a copy of the code required to complete the question along with a screenshot of the code and a screenshot of the output.**

1. Turn on tweets from the example **J-Elasticsearch** notebook. Search for tweets in the last 5 minutes from the (Windows or Linux) shell with **curl** (do the math yourself—don’t make this calculation automatic).

**(base) jovyan@jupyter:~$ curl -X GET "http://elasticsearch:9200/tweets/\_search?pretty&q=timestamp\_format>2022-11-30T01\:08\:00andtimestamp\_format<2022-11-30T01\:13\:00"**

**I don’t believe this query string acted as expected. I tried for many hours to figure out how to query last 5 minutes and this is the best I could do.**

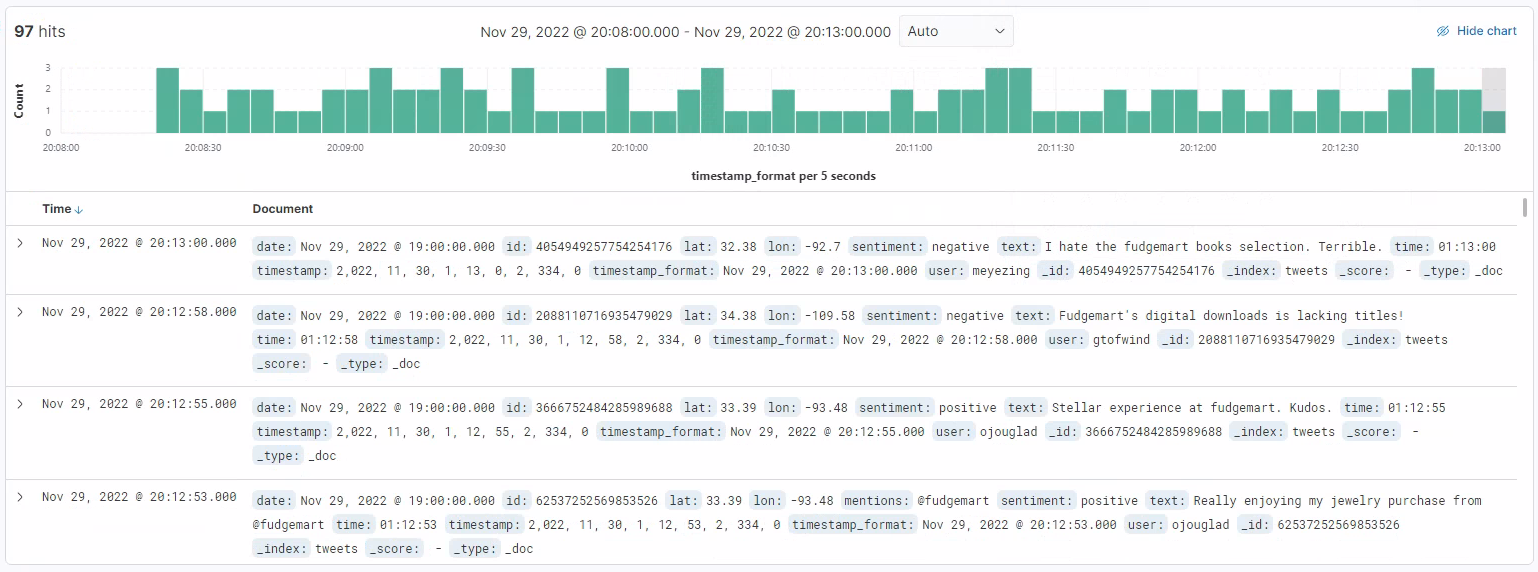
****

****

1. Do the same thing from the Kibana UI for Elasticsearch. What must you do before you can do that and why? Turn off the tweets.

**Prior to any Kibana query, an index pattern must be specified. Due to this, it was required to create an index pattern of tweets\*. I also needed to specify a datetime field and chose timestamp\_format as this allowed a very easy way to query based on time.**

****

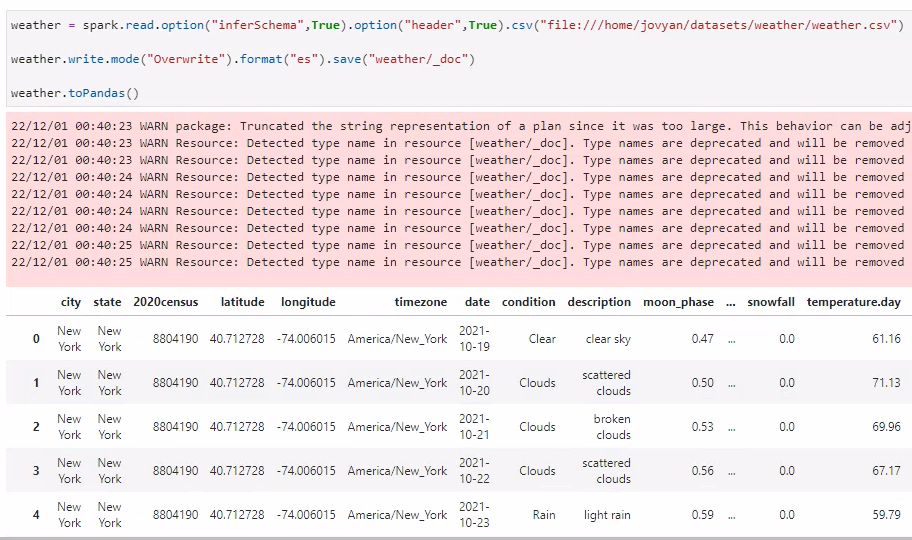
****

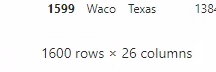
1. Write PySpark to load the 1,600-line weather data set into Elasticsearch under the index `weather` with default index type.

**weather = spark.read.option("inferSchema",True).option("header",True).csv("file:///home/jovyan/datasets/weather/weather.csv")**

**weather.write.mode("Overwrite").format("es").save("weather/\_doc")**

**weather.toPandas()**

****

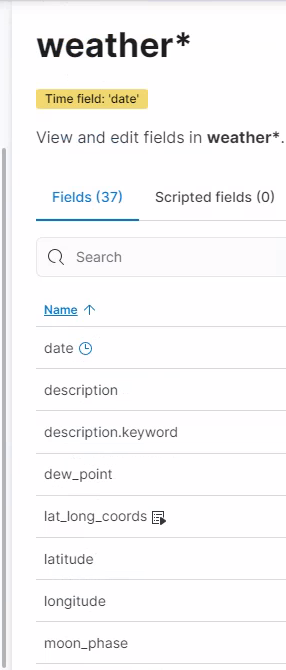


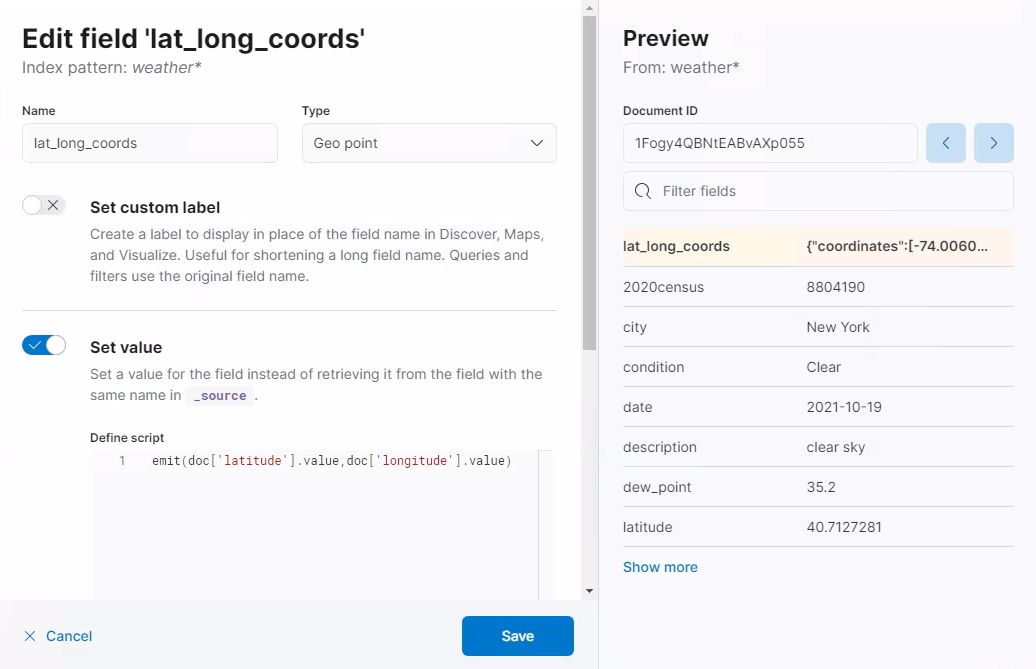
1. Use a **curl** command from the command line to hit the Elasticsearch API and demonstrate that there are 1,600 documents in the **weather** index.

**(base) jovyan@jupyter:~$ curl -X GET "http://elasticsearch:9200/weather/\_search?pretty"**



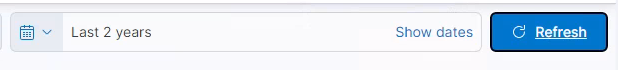
1. Set up a **weather** index pattern in Kibana based on the **weather** index from Elasticsearch. Make sure you have a **geo\_point** based on lat/lon type and have selected a **@timestamp** field using the date field. Provide a screenshot including the fields in question.

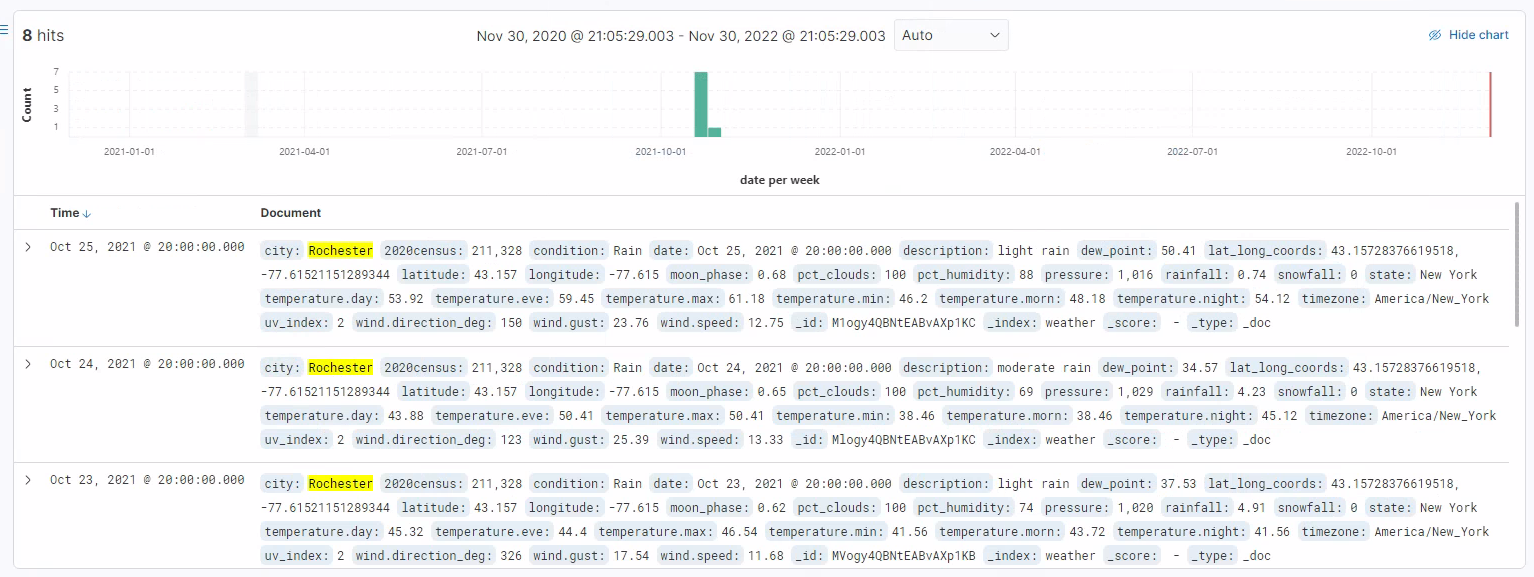


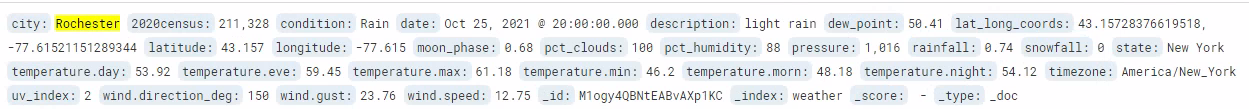


1. Demonstrate your Kibana index pattern is functional. For the most recent entry, get the weather for a city of your choice.

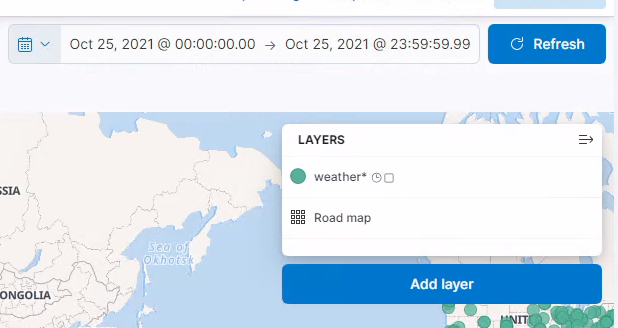


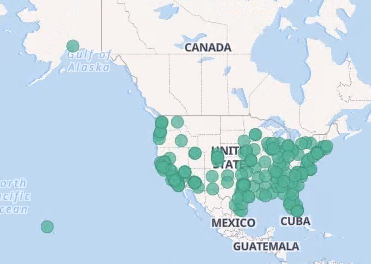


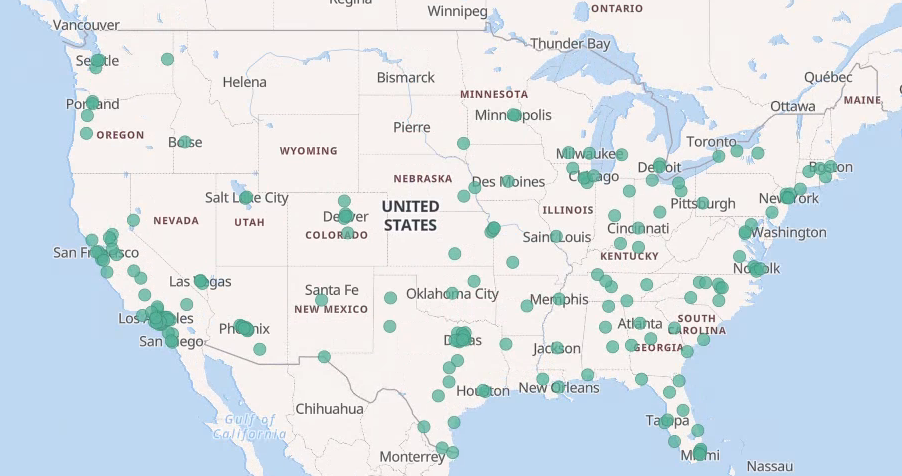




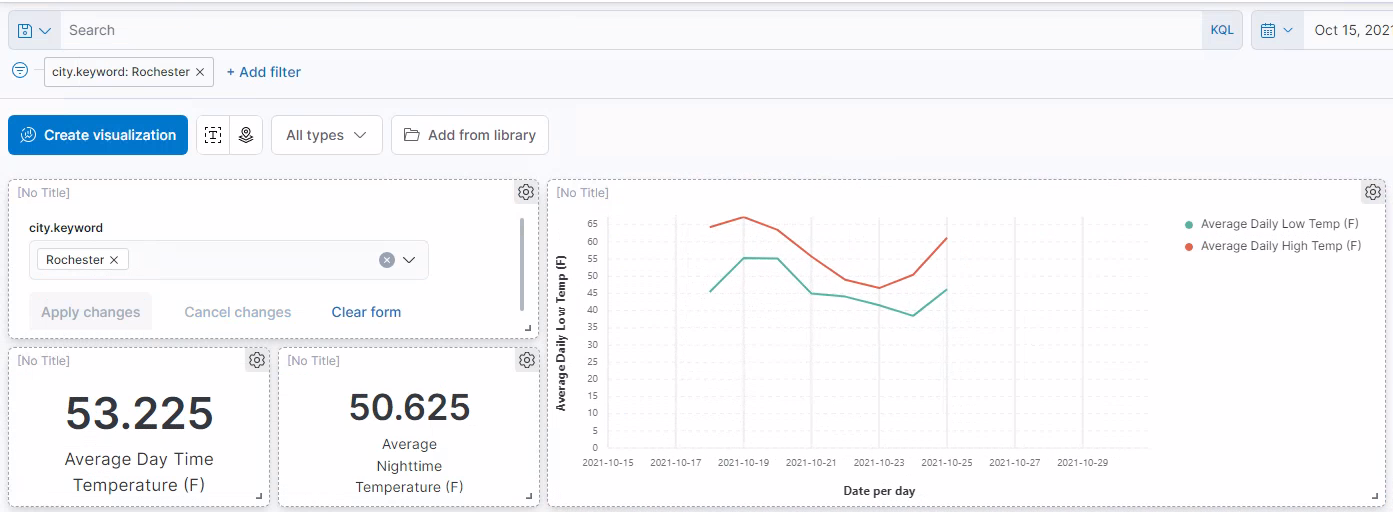
1. Create a Kibana map displaying the weather locations for the most recent weather data. Use any layer(s) of data you wish. Provide a screenshot of the map with data points on it.

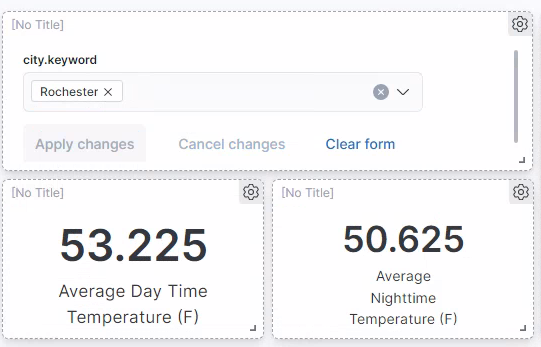


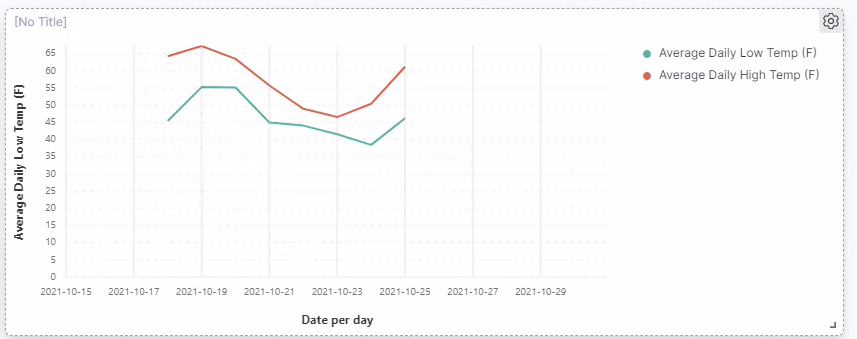




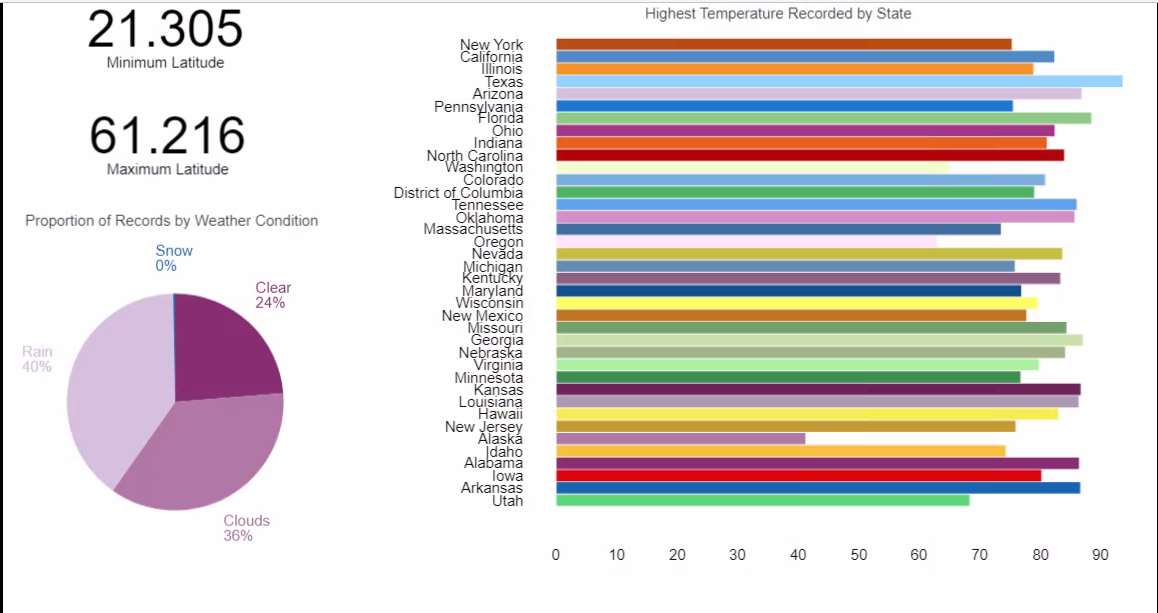
1. Create a Kibana dashboard that, when you select a city, will display the average day time and nighttime temperature for that city, in addition to a line chart of the average daily high and lows for all data on that city. Provide a screenshot of the dashboard in action.

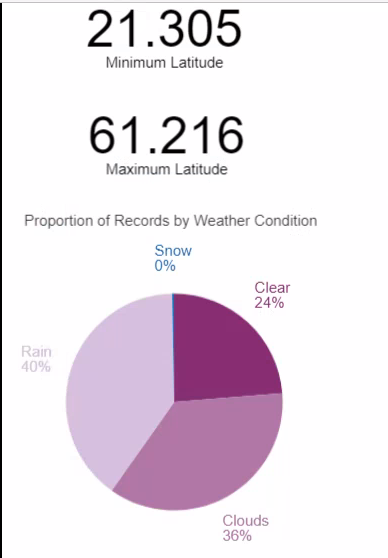


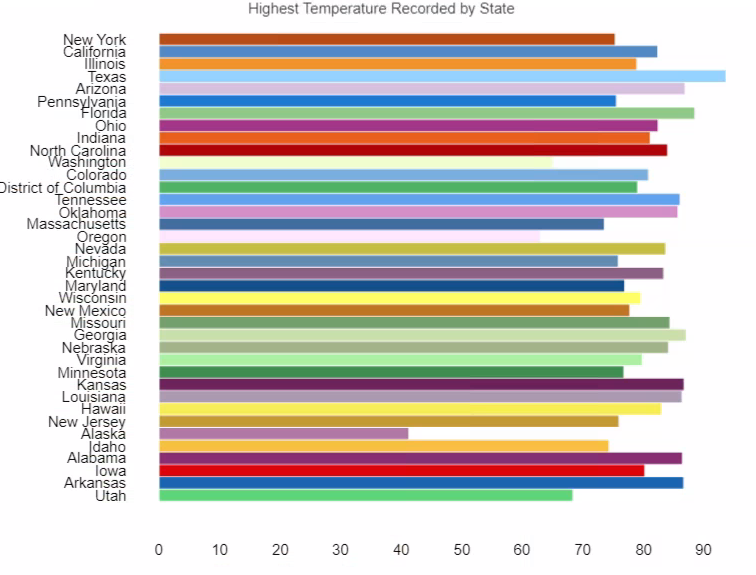




1. Create a Kibana Canvas! Display at least two metrics and two charts. Decide which data you want to display and how you would like to present it. Provide a screenshot of the Canvas. You can do this for a set of cities or a single city.





****