



Covid-19 Weather Patterns

By Andrew Wolfe and Christopher Gamboa

Agenda

- I. Goals
- II. Problems
- III. Explanation of Code
- IV. Data and Visualizations
- V. Conclusion

Goals

- Original Goals
 - Compare the rise and fall of Covid-19 cases with Fitbit Data in the same time period
 - Discover any correlations for how Covid-19 has affected the physical and mental health of people
 - Create at least 2 visualizations
- Final Goals
 - Compare the rise and fall of Covid-19 cases with temperature and humidity changes in Colorado
 - Discover any correlations between Covid-19 and increasing temperatures
 - Create at least 4 visualizations
 - Come up with calculations relating to the data to make a well-thought-out conclusion

Problems

- API
 - Had trouble finding reliable API with current 2020 information relating to Covid and weather
- Tables sharing a key
 - Problems with establishing the correct counting involving our shared keys
- Counting up to 25 data points
 - Solved by using a counting variable and using break/continue
- Small coding issues, involving JOIN, matplotlib, and CSV files
 - Solved by google searching and attending office hours
- API Changed at the last second (30 minutes before presenting!); no more Colorado
 - Solved by using Montana to run the code instead

Problems

Side Note: Our Covid API (CDC) changed from having Colorado to Montana data 30 minutes before presenting. All of our visualizations and data points shown in this presentation are from thirty minutes before the presentation when Colorado was present in the API. For our python Covid cases code file, we changed the state name to Montana so that you can see how our code is ran.

Instructions

def create_table_Colorado():

#Grab data from the following api:

[https://data.cdc.gov/resource/9mfq-cb36.json?\\$\\$app_token=vUO9dTV5pir6UBXemfD6uL8lZ](https://data.cdc.gov/resource/9mfq-cb36.json?$$app_token=vUO9dTV5pir6UBXemfD6uL8lZ). After this, create a database and inside that database, create a table named Covid with the following columns: date_id (integer), state (text), cases (integer), deaths (integer). Create a table named Dates with the following columns: date (text unique), date_id (Integer, primary key). Grab the data from the API and insert it into the table. Associate two tables with a key using the date_id. Lastly, make sure to only insert 25 data points per run. Please keep in mind that our Covid API changed at the last second and does not contain Colorado information anymore. Our presentation contained the information from Colorado Covid Cases. If you run our code now on Visual Studio, the create_table_Colorado() function will contain information from Montana, not Colorado, and thus have different data points.

def get_data():

#Grab data from the following api: <https://www.metaweather.com/api/location/>. After this, create a table in the same database as the previous function named weather with the following parameters: "id" INTEGER UNIQUE PRIMARY KEY , "temperature" FLOAT, "date" TEXT, "humidity" FLOAT)'). Grab the temperature and humidity data from Colorado in the API during the 100 day span of March1-June8. Convert to Fahrenheit! Make sure to only submit 25 data points per run! Please keep in mind that our Covid API changed at the last second and does not contain Colorado information anymore. Our presentation contained the information from Colorado Covid Cases. If you run our code now on Visual Studio, the create_table_Colorado() function will contain information from Montana, not Colorado, and thus have different data points.

Instructions Continued

def joining():

#Once all of the databases are correct and have 100 data points in each table, use the join operator on SQL to create a new table named Covid_Weather that contains date_id, state, cases, deaths, temperature, and humidity from the tables Covid and weather. Please keep in mind that our Covid API changed at the last second and does not contain Colorado information anymore. Our presentation contained the information from Colorado Covid Cases. If you run our code now on Visual Studio, the create_table_Colorado() function will contain information from Montana, not Colorado, and thus have different data points.

Visualizations Code (no functions used!)

#Select data from the weather and Covid table. From these tables calculate the average temperature (for each month), the average humidity (for each month), the average number of Covid-19 cases per day (for each month), and the average number of Covid-19 deaths per day (for each month). Using this information, create three subplots and one bar graph. One subplot will contain the averages from temperature and humidity, the second subplot will contain the averages from cases and deaths, and the third subplot will contain the total number of Covid cases and deaths per month (for each month). The fourth visualization will be a bar graph comparing the average number of deaths per day due to Covid for each month. Lastly, insert all of the data calculated into a CSV file. Please keep in mind that our Covid API changed at the last second and does not contain Colorado information anymore. Our presentation contained the information from Colorado Covid Cases. If you run our code now on Visual Studio, the create_table_Colorado() function will contain information from Montana, not Colorado, and thus have different data points.

Database Tables

	date	date_id
	Filter	Filter
45	2020-04-14T00:00:00.000	45
46	2020-04-15T00:00:00.000	46
47	2020-04-16T00:00:00.000	47
48	2020-04-17T00:00:00.000	48
49	2020-04-18T00:00:00.000	49
50	2020-04-19T00:00:00.000	50
51	2020-04-20T00:00:00.000	51
52	2020-04-21T00:00:00.000	52
53	2020-04-22T00:00:00.000	53
54	2020-04-23T00:00:00.000	54
55	2020-04-24T00:00:00.000	55
56	2020-04-25T00:00:00.000	56
57	2020-04-26T00:00:00.000	57
58	2020-04-27T00:00:00.000	58
59	2020-04-28T00:00:00.000	59
60	2020-04-29T00:00:00.000	60
61	2020-04-30T00:00:00.000	61
62	2020-05-01T00:00:00.000	62
63	2020-05-02T00:00:00.000	63
64	2020-05-03T00:00:00.000	64
65	2020-05-04T00:00:00.000	65
66	2020-05-05T00:00:00.000	66
67	2020-05-06T00:00:00.000	67
68	2020-05-07T00:00:00.000	68
69	2020-05-08T00:00:00.000	69
70	2020-05-09T00:00:00.000	70

	date_id	state	cases	deaths
	Filter	Filter	Filter	Filter
45	45	CO	250	21
46	46	CO	339	28
47	47	CO	302	16
48	48	CO	465	18
49	49	CO	386	20
50	50	CO	201	10
51	51	CO	385	27
52	52	CO	349	37
53	53	CO	457	22
54	54	CO	357	44
55	55	CO	1002	121
56	56	CO	684	-2
57	57	CO	482	9
58	58	CO	448	27
59	59	CO	440	30
60	60	CO	437	30
61	61	CO	507	11
62	62	CO	486	43
63	63	CO	452	12
64	64	CO	414	10
65	65	CO	344	9
66	66	CO	439	52
67	67	CO	421	18
68	68	CO	580	23
69	69	CO	475	16
70	70	CO	523	7

	id	temperature	date	humidity
	Filter	Filter	Filter	Filter
45	45	49.703	2020-04-14	51.0
46	46	56.426	2020-04-15	50.0
47	47	62.933	2020-04-16	56.0
48	48	53.285	2020-04-17	64.0
49	49	54.689	2020-04-18	73.0
50	50	57.29	2020-04-19	55.0
51	51	58.568	2020-04-20	47.0
52	52	59.387	2020-04-21	46.0
53	53	62.528	2020-04-22	44.0
54	54	64.688	2020-04-23	49.0
55	55	63.599	2020-04-24	53.0
56	56	56.354	2020-04-25	65.0
57	57	62.555	2020-04-26	51.0
58	58	61.169	2020-04-27	60.0
59	59	49.397	2020-04-28	88.0
60	60	52.826	2020-04-29	72.0
61	61	50.432	2020-04-30	76.0
62	62	54.797	2020-05-01	67.0
63	63	58.784	2020-05-02	49.0
64	64	57.605	2020-05-03	59.0
65	65	56.795	2020-05-04	58.0
66	66	53.627	2020-05-05	56.0
67	67	58.334	2020-05-06	59.0
68	68	66.488	2020-05-07	46.0
69	69	68.855	2020-05-08	49.0
70	70	69.17	2020-05-09	43.0

	date_id	state	cases	deaths	temperature	humidity
	Filter	Filter	Filter	Filter	Filter	Filter
45	45	CO	250	21	49.703	51.0
46	46	CO	339	28	56.426	50.0
47	47	CO	302	16	62.933	56.0
48	48	CO	465	18	53.285	64.0
49	49	CO	386	20	54.689	73.0
50	50	CO	201	10	57.29	55.0
51	51	CO	385	27	58.568	47.0
52	52	CO	349	37	59.387	46.0
53	53	CO	457	22	62.528	44.0
54	54	CO	357	44	64.688	49.0
55	55	CO	1002	121	63.599	53.0
56	56	CO	684	-2	56.354	65.0
57	57	CO	482	9	62.555	51.0
58	58	CO	448	27	61.169	60.0
59	59	CO	440	30	49.397	88.0
60	60	CO	437	30	52.826	72.0
61	61	CO	507	11	50.432	76.0
62	62	CO	486	43	54.797	67.0
63	63	CO	452	12	58.784	49.0
64	64	CO	414	10	57.605	59.0
65	65	CO	344	9	56.795	58.0
66	66	CO	439	52	53.627	56.0
67	67	CO	421	18	58.334	59.0
68	68	CO	580	23	66.488	46.0
69	69	CO	475	16	68.855	49.0
70	70	CO	523	7	69.17	43.0

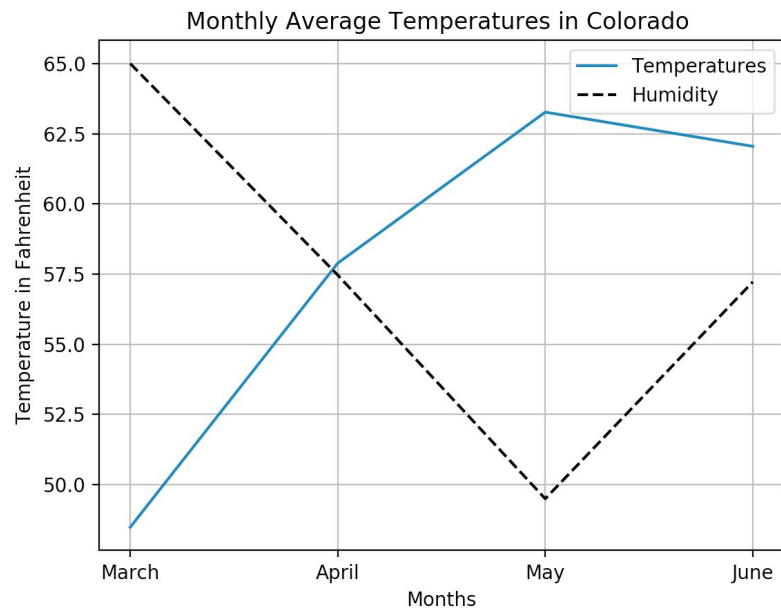
CSV File/Text File

weather_avgs

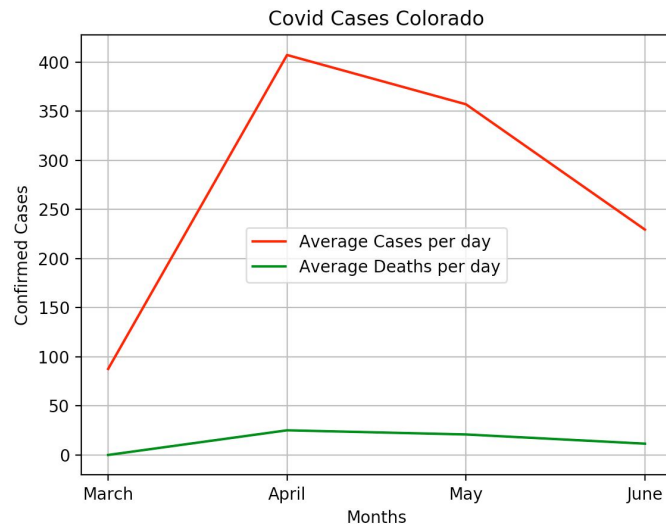
Type	March	April	May	June
Temperature Averages	48	58	63	62
Humidity Averages	65	57	50	57
Average Covid Cases Per Day	88	407	357	229
Average Covid Deaths Per Day	0	25	21	12

Visualizations

This graph shows the relationship between the monthly average temperatures to the monthly average humidity of March, April, May and June in Colorado.

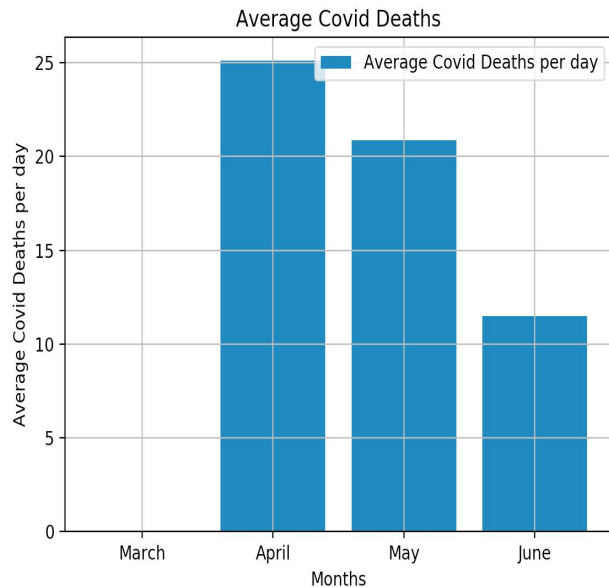


This graph shows the relationship between the monthly averages of new Covid-19 cases to the monthly averages of Covid-19 deaths.

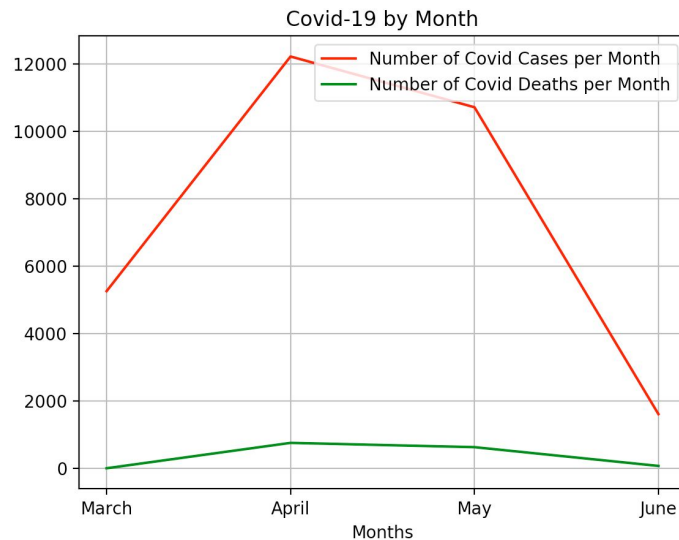


Visualizations

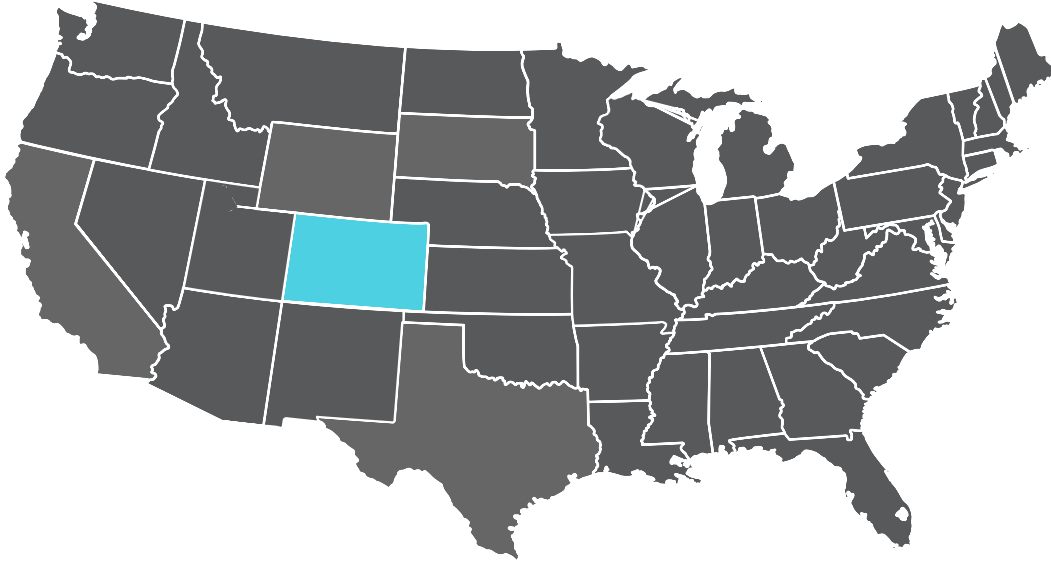
This bar graph shows that the average deaths per day from Covid-19 in Colorado decreased from April to June.



This graph shows the total number of Covid cases and deaths per month in Colorado. Keep in mind that June only took into account 8 days. However, June would prove to have less Covid cases than May.



Conclusion



14F ↑

Warmer Temperature

The temperature average rose by 14 Fahrenheit from March-June in Colorado.



8 g/kg⁻¹ ↓

Lower Humidity

The humidity fell by 8 g.kg⁻¹ from March-June in Colorado.



43% ↓

Covid Cases and Deaths

Covid-19 cases fell by 43% and deaths fell by about 50% from April-June.

Documentation

- Weather API (from MetaWeather): <https://www.metaweather.com/api/>
- Covid-19 API (from the CDC): [https://data.cdc.gov/resource/9mfq-cb36.json?\\$\\$app_token=vU09dTV5pir6UBXemfD6uL8lZ](https://data.cdc.gov/resource/9mfq-cb36.json?$$app_token=vU09dTV5pir6UBXemfD6uL8lZ)
- The function `def get_data()` gets data from MetaWeather API and creates a new table called `weather` in the `Covid_data` database. The function also limits the number of rows added to the table by increments of 25 and checks for and removes any duplicates in the table. This function also converts the date format so it may be used in python. The specific data that this function collects is the temperature, humidity, and date from March 1-June 8 in Colorado.
- The function `create_table_colorado()` gets data from the CDC API and creates a new table called `Covid` in the `Covid_data` database. The function limits the number of rows added to the table by increments of 25 and checks for and removes any duplicates in the table. This function also converts the date format so it may be used in python. The specific data that this function collects is date (`date_id`), new Covid-19 cases, new Covid-19 deaths, and state name (Colorado). Please keep in mind that our Covid API changed at the last second and does not contain Colorado information anymore. Our presentation contained the information from Colorado Covid Cases. If you run our code now on Visual Studio, the Covid function (`create_table_colorado`) will contain information from Montana, not Colorado, and thus will have different data points.
- The file `covid_viz.py` finds the monthly averages/sums for humidity, temperature, Covid-19 cases, and Covid-19 deaths. It stores this information into a csv file and creates line and bar charts of the data.
- The function `joining()` uses the join operator on SQL to create a new table named `Covid_Weather` in the `Covid_data` database that contains `date_id`, state, cases, deaths, temperature, and humidity from the tables `Covid` and `weather`. This code should be run after running the file `Final Project_COVID.py` four times.

Resources Used

- Difficulty formatting dates from weather and Covid API into database/sharing a key (12/6/20)
 - Used <https://docs.python.org/3/library/datetime.html> to learn about the import statement datetime and importance of UNIQUE in relation to SQL
- Difficulty in creating a graph with matplotlib (12/7/20)
 - Used <https://www.youtube.com/watch?v=UO98lIQ3QGI&feature=youtu.be> to learn more about the concepts of creating graphs with matplotlib and were able to successfully make two graphs
- Difficulty in storing our data into a file (12/08/2020)
 - Used <https://www.geeksforgeeks.org/writing-csv-files-in-python/> to learn about how to store more than one piece of data into a CSV file
- Difficulty in plotting two sets of data on same plot
 - Used Lecture matplotlib slide 15 to solve this problem

Thank you!

Questions?

