SI 206 Final Project Report

Snow Conditions in Vail VS. MTN Stock

The Olivia, Courtney, Jack Team
Courtney Newman, Olivia Sterling, Jack Tucker
SI 206 Fall 2020

LINK TO REPOSITORY: https://ocsterl.github.io/SI206FinalProject/

Table of Contents

Our Project Goals

As a Team of avid skiers, we were interested in exploring a connection between Skiing Conditions and Mountain Resort Stock prices.

We set 4 main goals for ourselves and this project:

1. Gather data from reputable APIs to calculate the typical skiing conditions for Vail throughout the 2019 Ski Season.
2. Supplement this data collection with information about MTN (Vail's IPO) stock.
3. Display this data in a comprehensible way through tables and visualizations.
4. Share the work necessary to complete this project in a fair and equitable way.

Our Achievements

Ultimately, we were successful in achieving many of our goals:

- We were able to find and utilize two unique Weather APIs (WorldWeather and WeatherAPI) to provide historical data.
- Using a third API, we honed in on MTN Stock and analyzed its performance throughout the 2018-2019 Ski Season.
- We created visualizations that display our findings.
- We completed the project feeling like we all contributed an equal amount.

Problems We Faced

Our group struggled initially in sourcing APIs that provided the information we needed:

- Firstly, because the 2020 Ski Season is just underway and the 2019-2020 Ski Season was interrupted due to the Pandemic, we needed to access Historical data.
  - Most historical data is not available under free API subscription plans.
    - We spent hours searching and eventually came out successful, using free trials for APIs
- Our initial goal was to find a relationship between Stocks and good skiing weather.
  - After collecting data, it become obvious that trends in stocks change over years and business decisions, rather than months and weather conditions.

Our Calculations

Below we have pictured screenshots of our Calculations text files and their corresponding links. The code for our calculations can be found in SnowfallAPI.py, WeatherAPI.py, stockAPI.py, respectively.

Conditions_Per_Month.TXT

https://github.com/ocsterl/SI206FinalProject/blob/main/Conditions_Per_Month.txt

```
Conditions in Vail Per Month
There were 18 clear days, 9 cloudy days, 0 rainy days, 3 snowy days in Vail during the month of December.
There were 0 clear days, 11 cloudy days, 0 rainy days, 21 snowy days in Vail during the month of January.
There were 0 clear days, 8 cloudy days, 0 rainy days, 21 snowy days in Vail during the month of February.
There were 0 clear days, 6 cloudy days, 1 rainy days, 24 snowy days in Vail during the month of March.
```

Weather_In_Vail.TXT

https://github.com/ocsterl/SI206FinalProject/blob/main/Average_Temp_Per_Month.txt

```
1  Average Temperature Per Month
2  Average Temperature in December was 12.548387096774194.
3  Average Temperature in January was 23.419354838709676.
4  Average Temperature in Feburary was 22.071428571428573.
5  Average Temperature in March was 29.35483870967742.
```

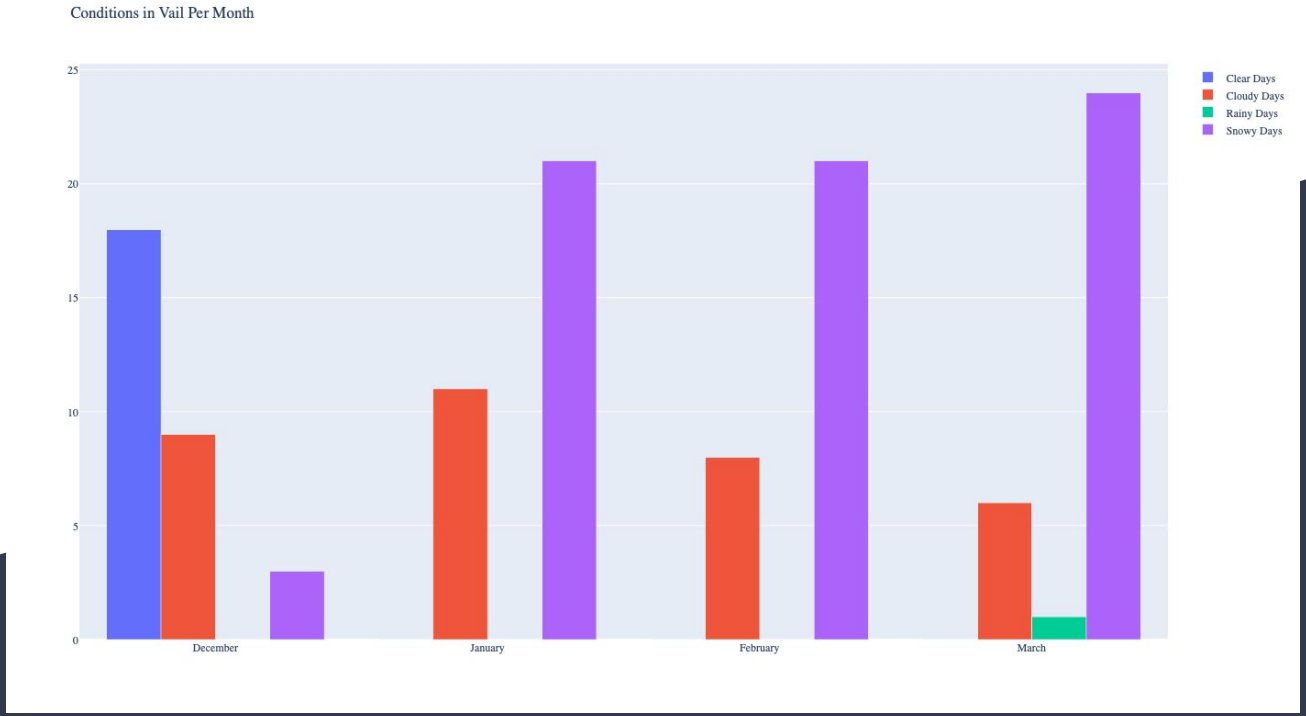
Average_Monthly_Stock_Price.TXT

https://github.com/ocsterl/SI206FinalProject/blob/main/Average_Monthly_Stock_Price.txt

```
Monthly Average Price of MTN For December, January, February, and March
The average price of MTN in December was $214.314.
The average price of MTN in January was $195.49809523809523.
The average price of MTN in February was $204.77421052631576.
The average price of MTN in March was $212.64857142857142.
```

Our Visualizations

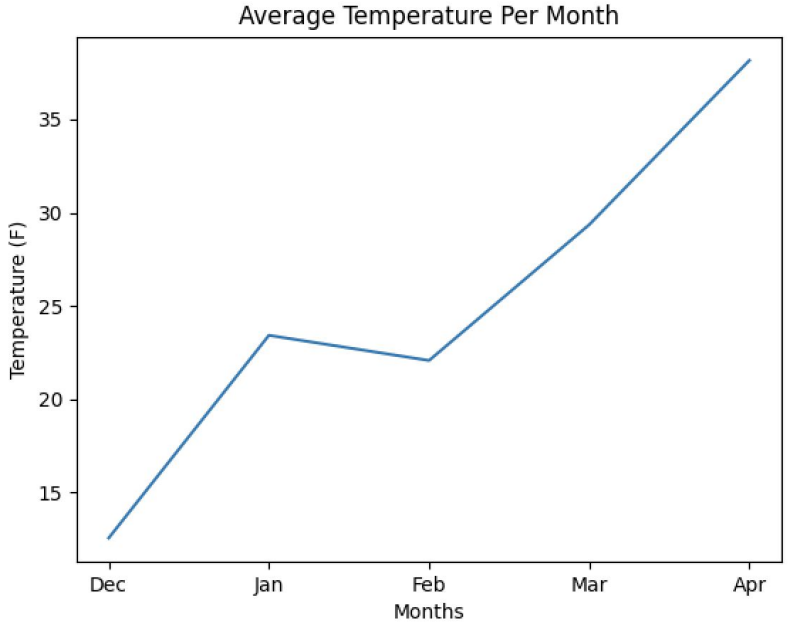
Figure 1. This graph demonstrates the typical monthly conditions in Vail throughout the four primary months (December, January, February, March) of Ski Season.



The data used in this graph was collected from the WeatherAPI and organized in the Get_Most_Common_Condition function in the SnowfallAPI.py file.

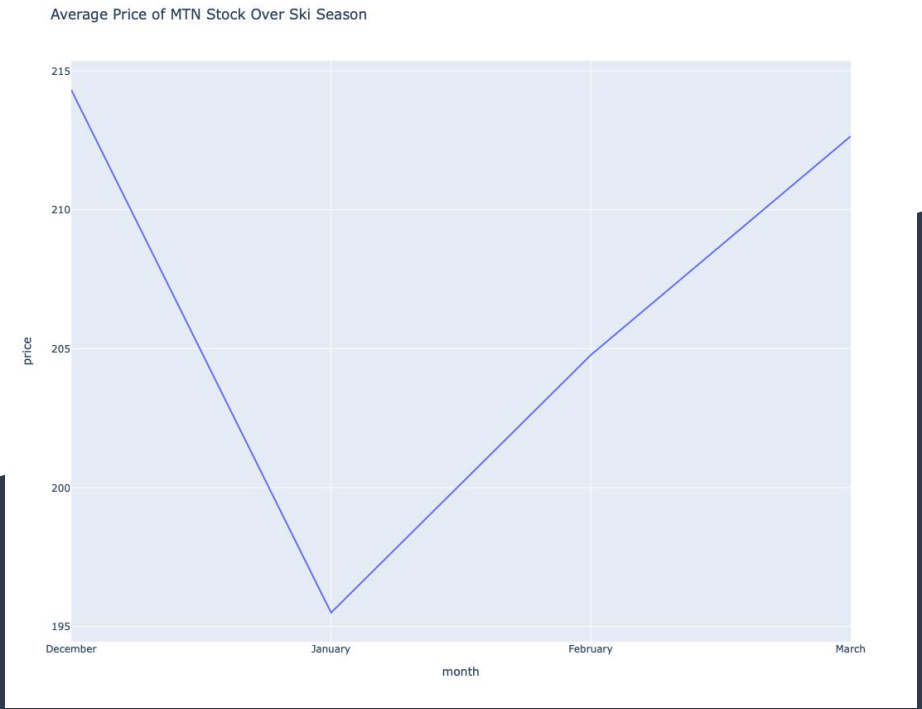
Our Visualizations

Figure 2. This graph demonstrates the average temperature in Vail throughout the four primary months (December, January, February, March) of Ski Season.



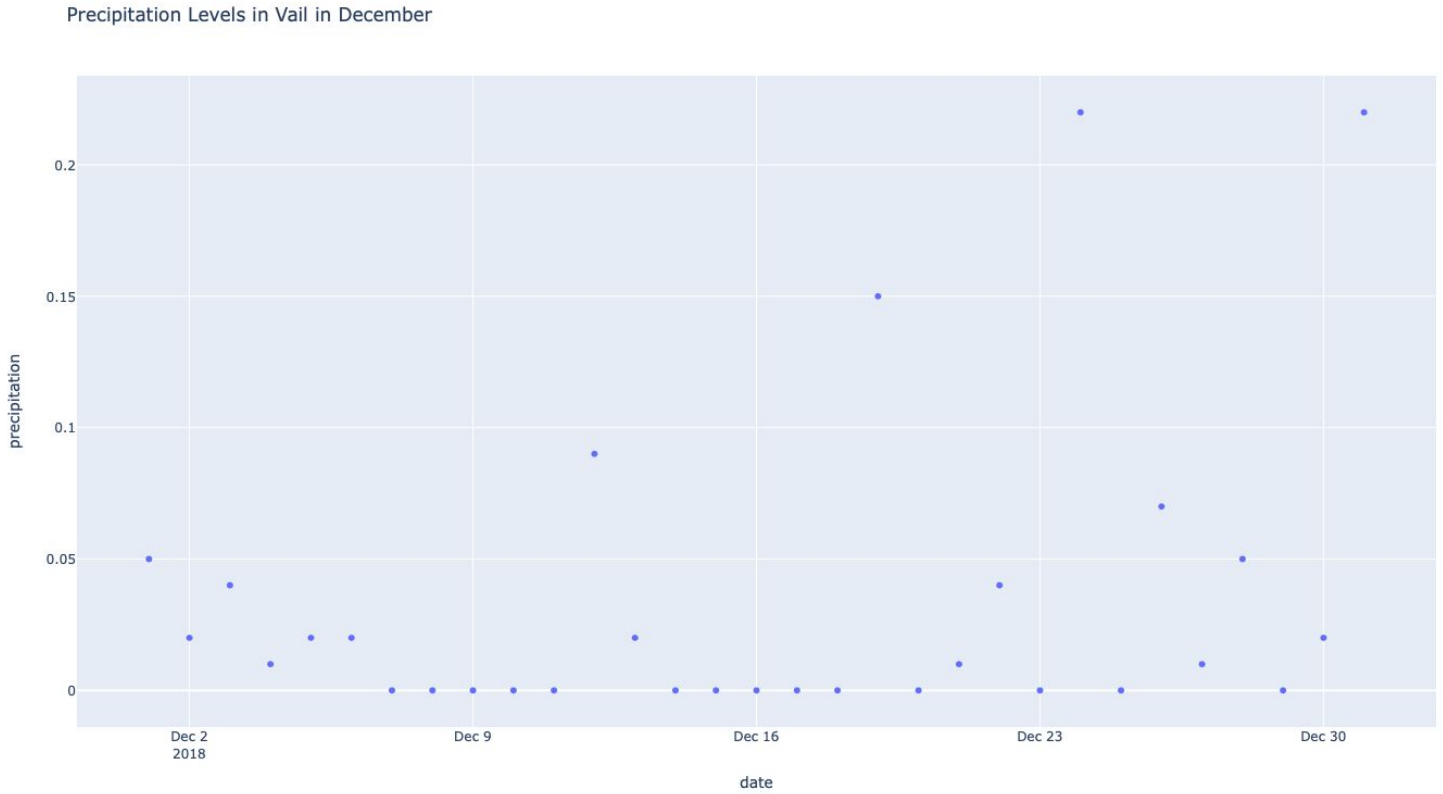
Our Visualizations

Figure 3. This graph demonstrates the average stock price of MTN throughout the four primary months (December, January, February, March) of Ski Season.



Our Visualizations

Figure 4. This graph represents the precipitation levels in Vail during the Month of December (One of the historically busiest months of Ski Season)



Instructions for Running Our Code

To run our code, follow the proceeding steps:

1. Open SnowfallAPI.py, WeatherAPI.py, and StockAPI.py
2. Run these three files to insert 6 tables into the finaldatabase.db file in the repository.
    - These files will also create the visualizations pictured in the previous 4 slides in a popup window, giving the option to save as a .png file.
3. In order for this code to execute properly, you will also need to install the following Python functions onto your local computer to access the imported modules:
    - SQLite3, Requests, DateTime, Plotly, Plotly.Graph_Objects, Matplotlib

Function Documentation

The table below explains the function documentation created in the SnowfallAPI.py

FUNCTION	RESULT
SetUp()	**OUTPUT:** Cur, Conn In this function I am creating the file path for the database that I will write to later in the code, creating cur and conn, and returning those variables.
Get_Data()	**OUTPUT:** c_results, p_results I am returning two lists of tuples that include dates and their respective conditions and dates and their respective precipitation levels.
ConditionTable()	**INPUT:** Cur, Conn **OUTPUT:** Condition Table in the Database In this function I am passing in cur and conn to create a table that consists of dates, their IDs, and the conditions for each day. The data is added 25 days at a time with each run to the database

FUNCTION	RESULT
PrecipitationTable()	**INPUT:** Cur, Conn **OUTPUT:** Table in the Database It creates a table in the database adding data to that table 25 days at a time with each run.
Get_Most_Common_Condition()	**OUTPUT:** Text file I am collecting a set of lists of weather conditions and writing the results into a text file.
ConditionsGraph()	**OUTPUT:** A popup webpage of my visualization
JoinTables()	**INPUT:** cur, conn **OUTPUT:** cur.fetchall() We are using a join statement to access the Dates from the Weather table and Precipitation levels from the PrecipitationInVail table.

Function Documentation

The table below explains the function documentation created in the weatherapi.py

FUNCTION	RESULT
SetUpDatabase()	**Output: Cur, Conn** I am creating a path to go into the api document for each day using datetime. Here, I am able to use this to run each day for 150 days in the following functions.
get_temp_and_day()	**Output: List of tuples for date, temperature, and hours of sun** I am allowing the database to run as long as it does not exceed 150 entries
create_table()	**Output: Weather and Averages Tables** I am creating a table for Weather and inserting date, temperature, and hours of sun. I am then creating another table for averages and calculating the average temperature for each month
createvisual()	**Output:** In this function I am calculating the averages of temperature for each month of the season and creating my table using matplot.lib

Function Documentation

The table below explains the function documentation created in the StocksAPI.py

FUNCTION	RESULT
getHistoricalPrices()	**Input:** stock ticker, start date, end date **Output:** returns the price and trading volume for the stock ticker over the date range that was inputted
SetUpDatabase()	**OUTPUT:** Cur, Conn In this function I am creating the file path for the database that I will write to later in the code, creating cur and conn, and returning those variables.
create_price_table(), create_volume_table()	**Input:** list of dates and corresponding stock price, list of dates and corresponding trading volume **Output:** creates a table for the day/price and day/ trading volume for the stock ticker. The data is added 25 days at a time with each run to the database
creategraph()	**Output:** a line graph of average stock price over 4 month period using matplot.lib

Resources

DATE	ISSUE DESCRIPTION	LOCATION OF RESOURCE	RESULT
12/5/2020	Snowfall API Layout Understanding	Within WeatherAPI: https://www.weatherapi.com/api-explorer.aspx	Helped in understanding how the data within the API was organized, especially when indexing for information
12/8/2020	API Understanding	Homework #7	Assisted in API data collection.
12/10/2020	Plotly Functionality	https://plotly.com/python/figure-labels/ https://plotly.com/python/bar-charts/#customize-bar-chart-with-plotly-express	Used when creating visualizations with Plotly. Provided a template for creating grouped bar graphs and changing fonts for Titles.
12/12/2020	Uploading to a Database	Homework #8	Used the information from this HW to better understand how to work within a database and to create a join statement