Balloon Flight Data Analysis

Intern: Kijjiketchme Southern-Fox, Mentors: Chris Yoder, Sarah Roth

NASA-WFF Balloon Program Office (BPO) / Balloon Research & Development Lab (BRDL)

ABSTRACT

The Balloon Program Office has been supporting the scientific community with the launch of scientific balloons since the 1980s, and over the course of that time has kept data about each balloon and flight in several databases. Last semester (Spring 2021), my project involved consolidating these databases into a master database, as well as developing Python tools to expand it further. This Summer, I have been developing a database-like tool from the master database which will enable comparisons of predicted vs. actual quantities for parameters such as air temperature, radiation, latitude/longitude/altitude versus time, and many others, for any balloon flight.

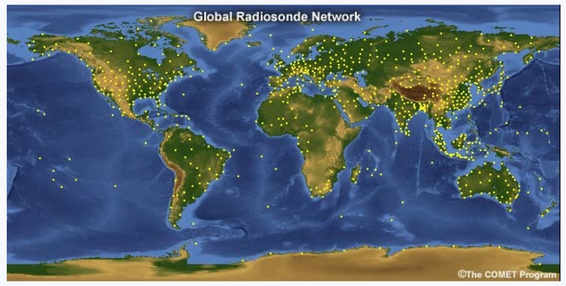
A TYPICAL USE CASE

On September 28, 2019, a 40 million cubic foot balloon – Flight 701N - was launched from Fort Sumner, New Mexico. Its science goal was to characterize Earth-like planets orbiting Sun-like stars elsewhere in the galaxy. On board were many instruments designed to take continuous measurements of the temperature, pressure, and radiation of the atmosphere. **How did that data compare with what we may have expected to find?** The best way to find out would be to compare it with a similar dataset from not too far away in space (including altitude) and time.

The WASP-Picture-C flight gondola. Flight hardware suspended below a scientific balloon can weigh 4000-6000 lbs and produce large amounts of data.

UTILIZING OPEN SOURCE DATA

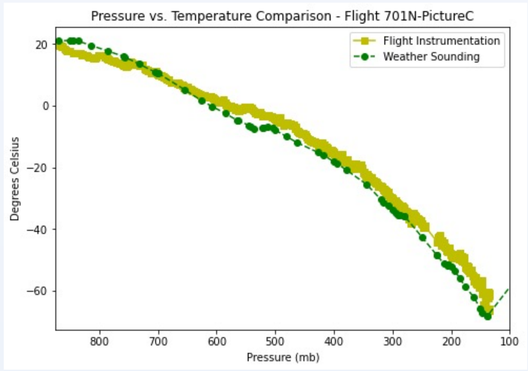
Fortunately, there are more than 2,700 locations throughout the world where regular atmospheric soundings record the temperature and pressure from the ground to the stratosphere. The National Centers for Environmental Information (NCEI) makes the data from those soundings available to the public through the Integrated Global Radiosonde Archive (IGRA).



Locations where regular atmospheric sounding data is taken.

METHOD

- The closest weather station to the launch location is identified, via a Python script which computes the “great circle distance” between the latitude and longitude coordinates.
- Data from IGRA for the closest weather station and the closest timestamp is scraped from the Web via a Python script.
- Air temperature versus pressure measurements from the scientific balloon is plotted. Any anomalous or outlying data is deleted from the dataset.
- Air temperature versus pressure data from the radiosonde is also plotted so that the curves can be visually compared. Interpolations are made within the dataset to facilitate point-by-point comparisons with the measurements from the balloon.
- Average and maximum error values are reported.



A plot output by the Python program showing pressure versus air temperature from both the flight hardware and the nearby weather sounding. Pressure decreases left to right to reflect the decrease with altitude.

STORING THE DATA

BPO personnel are interested in making comparisons like these for many flights, possibly going as far back into the database as the early 2000s. They are interested not only in air temperature and pressure but in flight trajectories, wind speeds, magnetic fields, and many other parameters. Having an efficient way to track which comparisons have been made from which data sources was of even more importance than creating the comparison tools themselves. Relying on a multitude of Excel spreadsheets is not ideal, because when a comparison is made, someone would then need to update every single spreadsheet! So, a bit of data science was applied…

Logos for Python and SQL

SQL, or Standard Query Language, is a powerful language for managing relational databases. It is frequently employed as the back end for popular apps. It is also more versatile and takes up less memory than storing all your data in spreadsheets. I wrote a Python program which looks up the relevant information for each balloon flight from the Master Balloons Database, and also checks file folders associated with those flights for any data files that were output by the flight hardware. This information is then dynamically combined into a SQLite database (SQLite is a version of SQL). Users of the program will not need to know any SQL syntax to issue queries, read the information, or check which files are available for making comparisons like those described in the ‘Method’ section. But they will get the benefits of the SQLite database, such as automatic updates when files are added, and customizable returns to their queries.

Logos for numpy and pandas Python libraries

WHAT I HAVE LEARNED

Through this project I have become increasingly more adept at using Python, especially with popular data science libraries like matplotlib, openpyxl, numpy, and pandas. I have discovered that there are vast amounts of Earth science data available, just waiting for analysis and good questions. Importantly, I have also learned that I have a real appetite for working with data to ask (and possibly answer) some of those types of questions.

REFERENCES

Durre, I., Vose, R. S., & Wuertz, D. B. (2006). Overview of the integrated global radiosonde archive. *Journal of Climate*, *19*(1), 53-68.

National Centers for Environmental Information (NCEI) (2021) *Integrated Global Radiosonde Archive (IGRA).* https://www.ncdc.noaa.gov/data-access/weather-balloon/integrated-global-radiosonde-archive

University of Wyoming College of Engineering, Department of Atmospheric Studies (n.d.). *Atmospheric Soundings.* Retrieved from http://weather.uwyo.edu/upperair/sounding.html

ACKNOWLEDGEMENTS

Thanks to **Sarah Roth** from the BPO for introducing me to the Balloons database.

Thanks to **Chris Yoder** from the BPO for helping me tackle many coding problems and providing general education about balloon flight.

Thanks to the Maryland Space Grant for funding my internship!

Thanks to Sarah Alspaw and Pat Benner, my internship coordinators for Goddard Space Flight Center and Wallops Flight Facility!