Dyson School of Design Engineering | MEng Design Engineering

**Module Exam**

|  |  |
| --- | --- |
| Module code and Name | DE4-SIOT Sensing & IoT |
| Student CID | 01234467 |
|  |  |
| Assessment date | **10am, 13th Jan 2020** |
|  |  |

**Presentation URL (publicly accessible link):**

https://youtu.be/24N-qHffTkw

**Code & Data (publicly accessible link):**

**Website:**

https://sites.google.com/view/jameskrasuckisiot/home

**Code:** <https://script.google.com/d/1mpDU1llcKbCSgT_mhn9pP1h6P_Ym9Eisw2kDJPYDqNAmA7tWubk3KO7R/edit?usp=sharing>

or

https://github.com/jk1916/SIOT.git

**Live Recording Data:**

https://docs.google.com/spreadsheets/d/1pNXfNzlWG-bxmejvyEvmosPsRDcQx1h6AlrGTUwPfaM/edit?usp=sharing

**Historic Data:**

https://docs.google.com/spreadsheets/d/1zSV6Ng7qW-ErtawKj\_eDr5mW7J4ZlQXj95HPjX2MLd0/edit?usp=sharing

Coursework 1: Sensing

**Introduction**

As a runner, I have been collecting data using a sports watch for over 2 years. This has provided me with a huge amount of data regarding my progress and activity levels over this time, and all this data has been uploaded to Strava [1]. Strava is excellent for analysing workouts activity by activity, in terms of sports specific data. However, I wanted to examine the effect of external factors on my performance by creating a system in which Strava activity data and weather data could be combined.

There are a lot of apps that perform useful tasks using the Strava API [2], and one of the notable ones is the Klimat app [3]. The Klimat app takes weather data for a given activity and updates the weather summary as the description of that activity. This is useful for browsing activities and seeing what the conditions were for a given performance. However, this weather data could not be compared within the Strava app, so any potential patterns in performance or macro-statistics were unattainable. By aiming to collect my own database of weather and fitness data, I hoped to gain a greater insight into the data.

The collected data could potentially be used for an athlete to recognise which weather conditions [4] yield a better performance and analyse how the seasons effect this. For example; does the athlete run faster in warmer conditions.

**Objectives**

1. Establish a robust system for collection of data autonomously without human interaction required.
2. Collect running performance data and store this information in real time.
3. Use running activity time and location to collect weather data for a given activity.
4. Complete data analysis of the two data sources to uncover any potential patterns between running performance and weather conditions.
5. Create an online visualisation of the collected data.

**Data Sources and Sensing set-up**

Before, data could be collected for the project, a storage location was needed. Google Sheets [5] was used to set up a database to save collected data. A Google Apps Scripts account was needed to retrieve third-party API data in the document. Google Apps Script [6] is a JavaScript-based scripting language hosted and run on Google servers that extends the functionality of Google services.

To acquire the running activity data captured on my sports watch there were two options. The first was to use the Garmin Connect app [7] to which the watch syncs activity files to via Bluetooth, and the second was to use the social media platform Strava. The Garmin app is set up to upload immediately to linked Strava accounts upon the synchronisation of a new activity file. Given Strava has a well-documented, free API, with overly generous call limits (30,000 per day), this was chosen over the Garmin API. The Strava API also produced easy to parse data in JavaScript Object Notation format, while the raw ‘fit.files’ from the Garmin Sports watch would likely have been more difficult to process.

Before setting up a Strava sensing platform, a developer account was made, and an app created. This allowed client IDs, secrets and access tokens to be generated for the app. These were then authenticated at the beginning of the programme script.

The second data source was weather for which there were multiple APIs available. The Dark Sky2 API [8] was chosen due to their renown hyper-local data provision. Dark Sky sources its data from many other weather providers and compiles this information to provide local weather data. Another identified weather source was the OpenWeather API [9]. However, Dark Sky 2 was chosen over this option as it allowed the free access to historic weather data using a specific time and location. This was particularly necessary to get weather information at the time of the start of a running activity file, rather than at the time of upload. It also meant that previous activities could have their weather data attained for a more comprehensive analysis of correlations. The API was free, with an upper limit of 1,000 calls per day.

To set up the Dark Sky API, a free account was made. This provided an API key that could be used within request URLs within the programme script.

**Data Collection and Storage Process**

**Sampling Rate**

After an initial authentication script was run to get specific access tokens for the Strava API, the main data collection could begin. Two separate Google Apps Scripts files were written to collect Strava and Dark Sky weather data and store it in a singular Google Sheet. These scripts were run using triggers within the application, and both were set up with a 5-minute timer. The scripts only add new data to the Google Sheet when a new Strava activity has been uploaded since the timestamp of the last activity. A 5-minute timer enabled the API to check for new activities often enough to account for an activity occurring any time within a given day. The Dark Sky API was also called using a 5-minute timer so that if a new activity had been uploaded, it would collect the weather data for that activity. Whilst a 1-minute trigger would have been more suited for instant activity checking, the 5-minute trigger ensured that neither API would exceed their respective call limit.

**Strava API**

The Strava API requires an initial authentication process. Strava uses the Oauth 2 [10] authentication procedure, and there is an Oauth2 Library by Google for Google Apps Scripts that must be used. The Google Apps Script file needs to be published as a web application that is publicly available so that the Strava Authentication can call the callback link during authentication. Once authentication is complete, data can be collected using the Strava service that provides various keys and tokens.

https://www.strava.com/api/v3/athlete/activities?per\_page=5&after=' + [time]

This URL requests the activities after a given timestamp. The ‘per\_page’ parameter changes the number activities that are returned. In the script this is set to 5, even though only 1 activity will ever be returned. This is to ensure running activities are discovered amongst a page that may contain several activities that aren’t running type. The time used is set to the time of the last activity discovered so that old activities are not added as new ones.

The data received is JSON data that contains many attributes, and this can be parsed to useful fields by changing the field names within the script. The data attributes collected were:

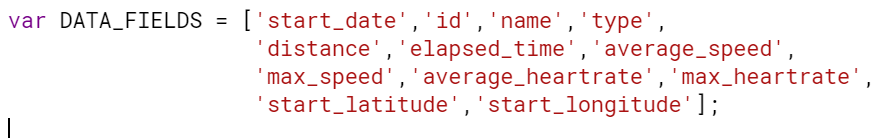


Figure 1 - Strava Data Fields

**Dark Sky API**

The Dark Sky API is well documented, and a lot of different data types can be returned within the JSON file.

https://api.darksky.net/forecast/[key]/[latitude],[longitude],[time] ?exclude=minutely,hourly,daily,alerts,flags&units=uk2

This URL is the ‘time machine’ request that Dark Sky offers. It enables access to weather at a given time and location when called. In this script, the time, latitude and longitude are determined from the most recent Strava activity data within the Google Sheet. The Dark Sky API returns a lot of data that is irrelevant to this project, so ‘exclude’ was used to get only the required data. Furthermore, the units were set the UK-based standards for weather data.

The attributes of the weather that are stored by the script are a selection of the available fields collected in the returned JSON file. Only certain attributes are pushed to be stored for the sake of relevance and simplification. The attributes chosen were:

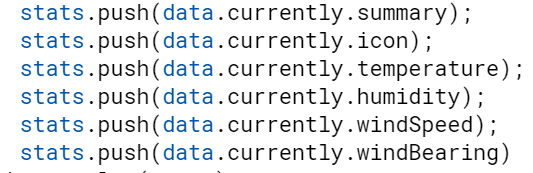


Figure 2 - Strava data attributes

**Data Storage**

Every 5 minutes, the data retrieval functions were run within their respective scripts for each API. If a new Strava activity is found then it is inserted to the bottom of the Google Sheet used to save the data, and subsequently the weather data joins this new row. As Google Sheets can store up to 5 million cells of data, there was no worry of the recording exceeding this limit. Furthermore, using automatically saving, cloud-based storage reduced the risk of data loss.

If an error occurs upon one of the functions running, the function triggers flare this error up. As it is likely only 1 new activity is saved to the Google Sheet each day, a daily error report was set up to allow intervention and to avoid continuous problems.

**Basic characteristics of the end-to-end systems set up and data**

A system diagram for the entire platform can be seen below.

**A screenshot of a cell phone

Description automatically generated**

Figure 3 - End-to-end System Diagram

The overall objectives of the project were achieved. Once authentication was set up for both APIs, Google Apps Script files [11] were successfully run autonomously using triggers for these files. The APIs were called and successfully collected data about performance metrics of new Strava activities, and subsequently weather data was provided for the given activity. This data was formatted into html code using a website called ‘awesometable.com’ [12]. This then allowed for the table URL to be embedded into a Google Sites website [13], alongside live graphs from the Google Sheets Spreadsheet.

**Basic Time Series Data Analysis**

The collected data was adequate but was only collected over 3 weeks, and so it was unlikely any performance correlations would be spotted. For this reason, the weather data for runs over the course of all of 2019 were captured to better demonstrate the effect of weather on running performance.

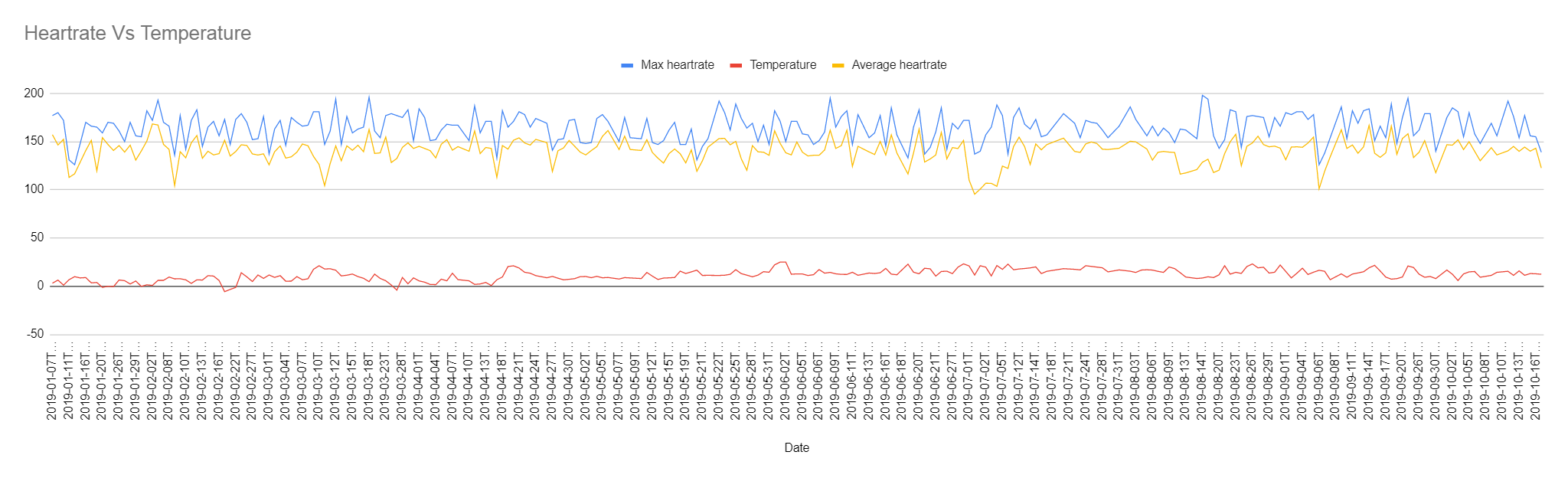


Figure 4 - Heartrate and Temperature Time Series Data

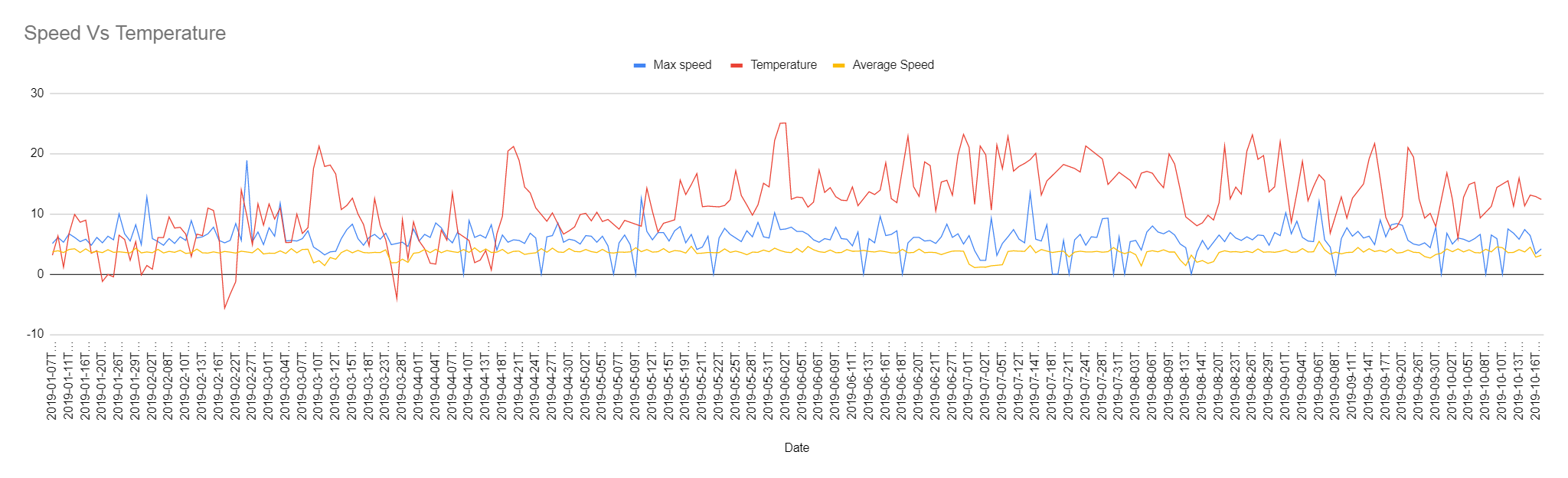


Figure 5 - Running Speed and Temperature Time Series Data

This time series analysis showed little correlation between general running speed and heartrate, and the temperature. If analysed carefully then peaks of high temperatures sometimes correlate with high speeds and heartrates. This may imply that the good weather inspires harder efforts when out running.

Coursework 2: Internet of Things

**Data Platform**

The data platform built was designed to dynamically load any necessary content in it without the requirement for a backend service. This dramatically reduces the effort required to deploy the site. A Google Sites webpage was created called ‘Strava Weather’ with the aim of collating and visualising some of the data collected:

*https://sites.google.com/view/jameskrasuckisiot/home*

**Interaction**

The data platform utilised an embedded table made using the Google integration ‘AwesomeTable’. By applying attribute filters to the Google Sheet in which data was being saved, it was possible to create a table that allowed for better data insights. E.g. it was possible to view runs by weather on the website and make comparisons between the performance metrics for a particular weather type. Similarly, it was possible to filter the performance data (like limiting the average speed to high values only) to analyse what weather occurred during particularly impressive performances.

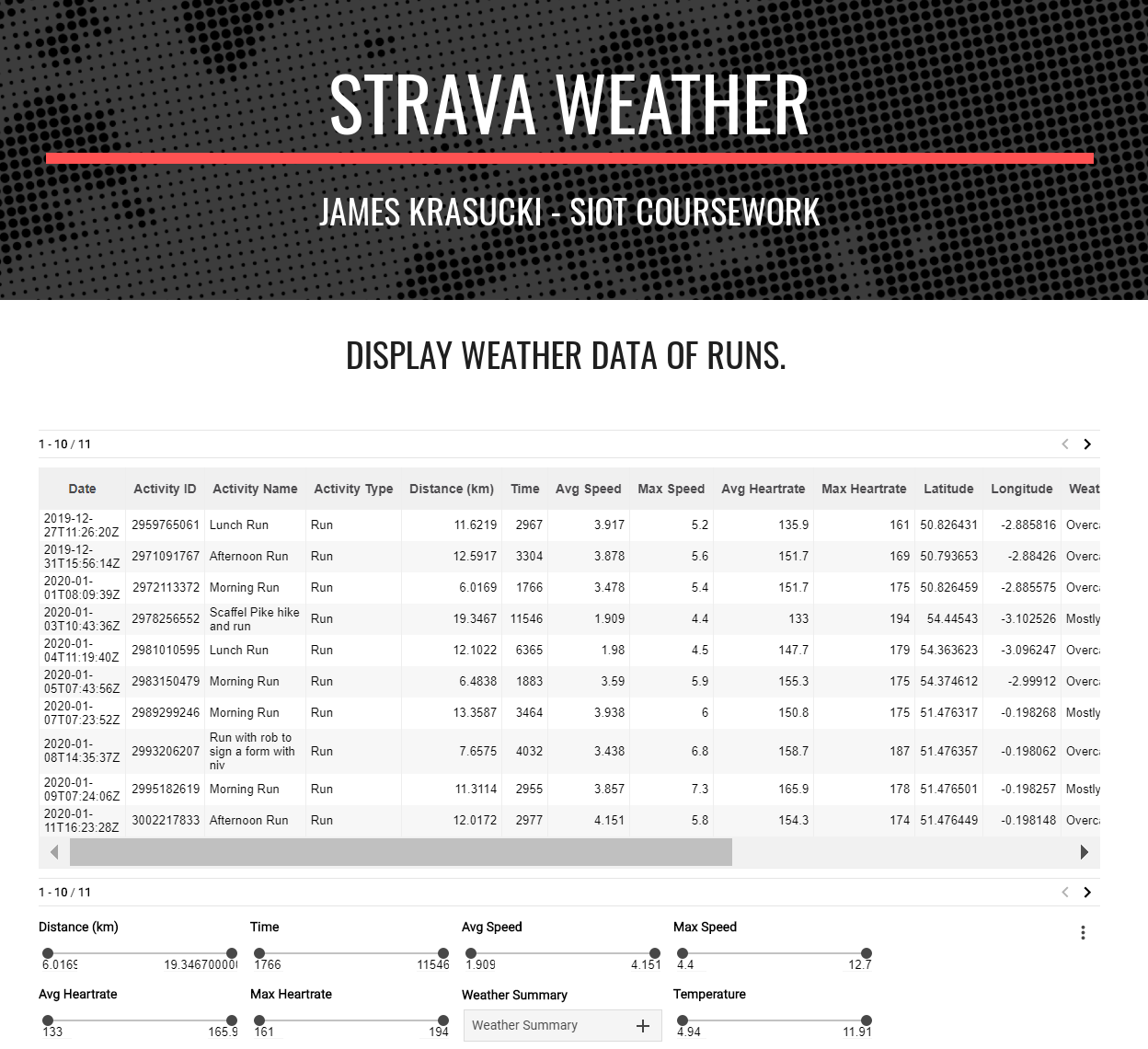


Figure 6 – Webpage Data Interaction

**Visualisation**

The webpage also contains charts that display data directly from the Google Sheet used to store the API data. Firstly, a pie chart is used to give the user an idea of the weather they experience most when running. This tells them what conditions they are used to performing in. The next charts show how speed (an externally comparable measure of performance) and heartrate (a personal measure of effort) vary with wind speed and temperature. This data is useful at examining the effect external factors may have had on a performance, e.g. if the user runs on a particularly hot day, they may see that their heart rate was higher than usual.

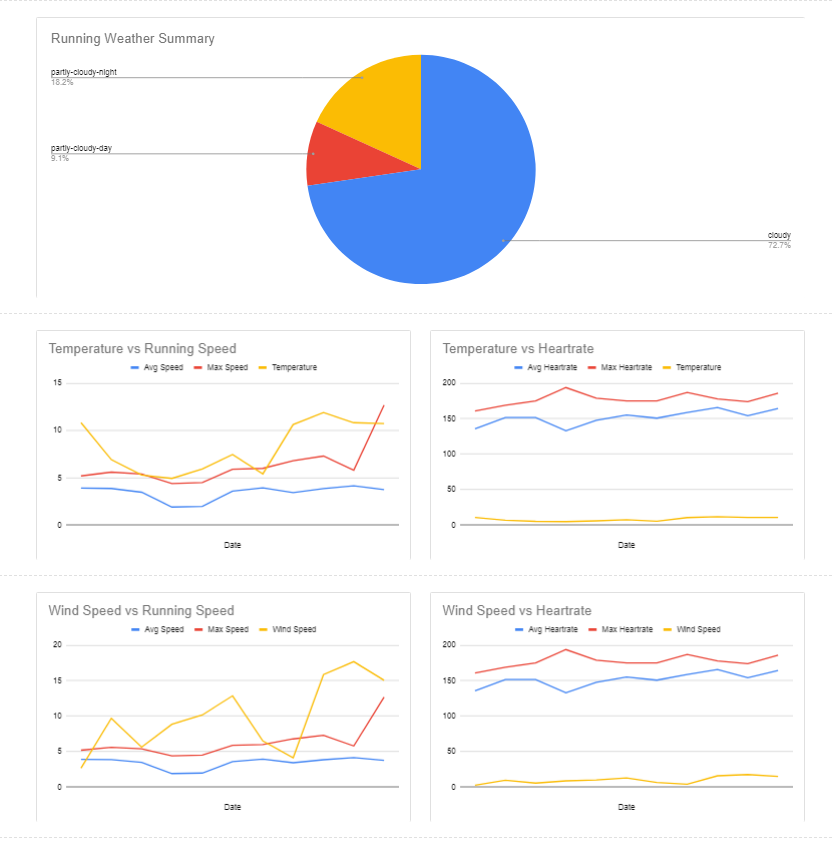


Figure 7 - Webpage Data Visualisation

**Data Analytics, Inferences and Insights**

To get a better idea of the patterns that occurring within the data, a separate data collection was performed. Here the Strava API was called to collect past running activities from the year 2019 and the weather data for these was also collected. This would allow a better identification of any correlations in the data and analysis of seasonal changes.

**Seasonality**

Seasonality analysis was performed on each data attribute collected to see how it changed over the course of the year. From the weather data collection, the temperature exhibited seasonal change as expected; rising over the summer.

A screenshot of a cell phone

Description automatically generated

Figure 8 - Temperature Change over the Year 2019

The Strava activity data lacked any real seasonality in any of the attributes (as seen in the heartrate and speed graphs). However, run distance did exhibit a slight average increase and peaks over the summer months. This implied that the warmer weather was promoting the desire to get outside more.

A screenshot of a cell phone

Description automatically generated

Figure 9 - Distance Run Time Series Analysis

**Correlation and Cross Correlation**

To get a clearer idea of whether temperature was having any affect on heart rate or running speed, further correlation graphs were produced.

Firstly, the original time-series graphs of heart rate, speed, and temperature were edited so that the data showed how an increase in temperature generally affected the variables. This again showed that there was almost no relation between temperature and performance; the trendlines for both heartrate and speed were almost completely flat.

However, to further confirm this hypothesis, cross correlation was performed in which the temperature and performance variables were plotted against each other. This again demonstrated the lack correlation between the 2 variables. Finally, the average value for the average heartrate and speed was calculated for each weather type to see if there was a preferential weather condition to perform in. However, the average was identical for every weather condition recorded.

A picture containing screenshot

Description automatically generated

Figure 10 - Average Heartrate as Temperature Increases

A screenshot of a social media post

Description automatically generated

Figure 11 - Average Speed as Temperature Increases

|  |  |
| --- | --- |
| Figure 12 - Cross Correlation of Temperature and Average Speed | Figure 13 - Cross Correlation of Temperature and Heartrate |

Table 1 - Average Performance Metric Values for Different Weather Types

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Weather** | **Clear** | **Partly Cloudy** | **Cloudy** | **Rain** | **Wind** |
| **Average Heartrate** | 139.2 | 139.2 | 139.2 | 139.2 | 139.2 |
| **Average Speed** | 2.91 | 2.91 | 2.91 | 2.91 | 2.91 |

**Insights**

The correlation analysis showed that running performance was not affected on a large scale by weather conditions. This was useful to show that weather did not influence overall performance and should not be factored in on a macro level to training analysis.

**Discussions on the important aspects of the project**

The project was successful in completing the objectives set out at the beginning. Data was collected efficiently and without errors. Whilst there was no outcome in terms of correlation of data for the project, the process of logging weather data for each run is still useful and novel. Often, I will look back at past runs to examine the data for them. By having logged the weather conditions it will be easier to remember these performances and even understand the ‘why’ behind that run a bit better.

Overall, I think the most important aspect of the project was the successful implementation and combination of two separate web APIs and their data.

**Avenues for Future Work and Potential Impact**

The data collected did not exhibit correlation on a macro level. However, I think that there would be scope for collecting weather data for specific parts of a run. Strava allows users to create ‘Segments’ that different users can compete along during their normal activities. If a user is one of the fastest along a particular segment, then their time is logged, and they are added to the ‘Course Records’ table publicly on Strava. Obviously, for a short segment, weather data (especially wind speed at a particular bearing) will have a large effect on user’s times. Therefore, a potential avenue for future work would be a Strava application that captures and uploads weather data when a Strava user achieves a course record on a particular segment. This weather data could then be displayed next to the record so that other users know what conditions they were facing.

A picture containing screenshot

Description automatically generated

Figure 14 - Strava Course Record for a Specific Segment

Furthermore, this project was performed using running data. Weather may be a more important factor in terms of performance for some sports. E.g. cycling is affected much more greatly by wind speed and aerodynamics. So this may be a more useful application for these other sports.

**References**

[1] <https://www.strava.com/>

[2] <http://developers.strava.com/docs/reference/>

[3] <https://klimat.app/>

[4] Knechtle B, Di Gangi S, Rüst CA, Villiger E, Rosemann T, et al. (2019) The role of weather conditions on running performance in the Boston Marathon from 1972 to 2018.

[5] https://docs.google.com/spreadsheets/d/1pNXfNzlWG-bxmejvyEvmosPsRDcQx1h6AlrGTUwPfaM/edit?usp=sharing

[6]<https://developers.google.com/apps-script>

[7] <https://connect.garmin.com/>

[8] <https://darksky.net/dev>

[9] <https://openweathermap.org/api>

[10] <https://developers.google.com/identity/protocols/OAuth2>

[11]https://script.google.com/d/1mpDU1llcKbCSgT\_mhn9pP1h6P\_Ym9Eisw2kDJPYDqNAmA7tWubk3KO7R/edit?usp=sharing

[12 <https://support.awesome-table.com/hc/en-us>

[13] ] <https://sites.google.com/view/jameskrasuckisiot/home>