College Use of Bluebikes in Boston

Kaitlyn O'Donnell

Northeastern University, Boston, MA, USA

Abstract

This project's goal is to see if Bluebike station usage correlates with college and university classes in the city of Boston. The project used Bluebike station trip data and coordinates to find the trips that were likely due to students in the area. Because the Bluebike trip data is large, the project focuses on July and September of 2019. The results indicate that Bluebikes are significantly used by college and university students, mainly Northeastern and Boston University students.

Introduction

The goal of this project was to analyze possible relationships between Bluebike station trips and proximity to various colleges in the city of Boston. It is unclear from just trip data if Bluebike trips numbers are influenced by nearby colleges. Depending on the results, Bluebike might want to install more bike locations close to colleges or advertise more to students. Because Bluebike is owned by the municipalities in the Boston area, this demographic analysis can help inform city policy as well.

The main feature of this database is the ability to retrieve all stations within a distance of meters to parcels owned by colleges in the city of Boston. This can be used by selecting a college, or by selecting a specific parcel.

Database Design

The key entities in the database are College, Parcel, Trip, and Station. A College can own zero or more Parcels. A Parcel must be owned by one College. Because the goal of this database is to analyze the relationship between Colleges and Bluebike trips, having Parcels not owned by a College would bloat the database unnecessarily.

The remaining key entities are Trip and Station. A Trip has exactly two stations associated with it. These are required because the original trip data did not have any entries with missing or extra stations. This is likely because a "trip" is characterized by these start and end locations. In contrast to the College-Parcel relationship, a Station can exist without any trips associated. This was mainly due to Bluebikes allowing access to their list of Stations. A secondary reason was

that, logically, a Station can exist within our scope without an associated Trip. If for some reason, a Station had no Trips for a month, that would be worthy of note.

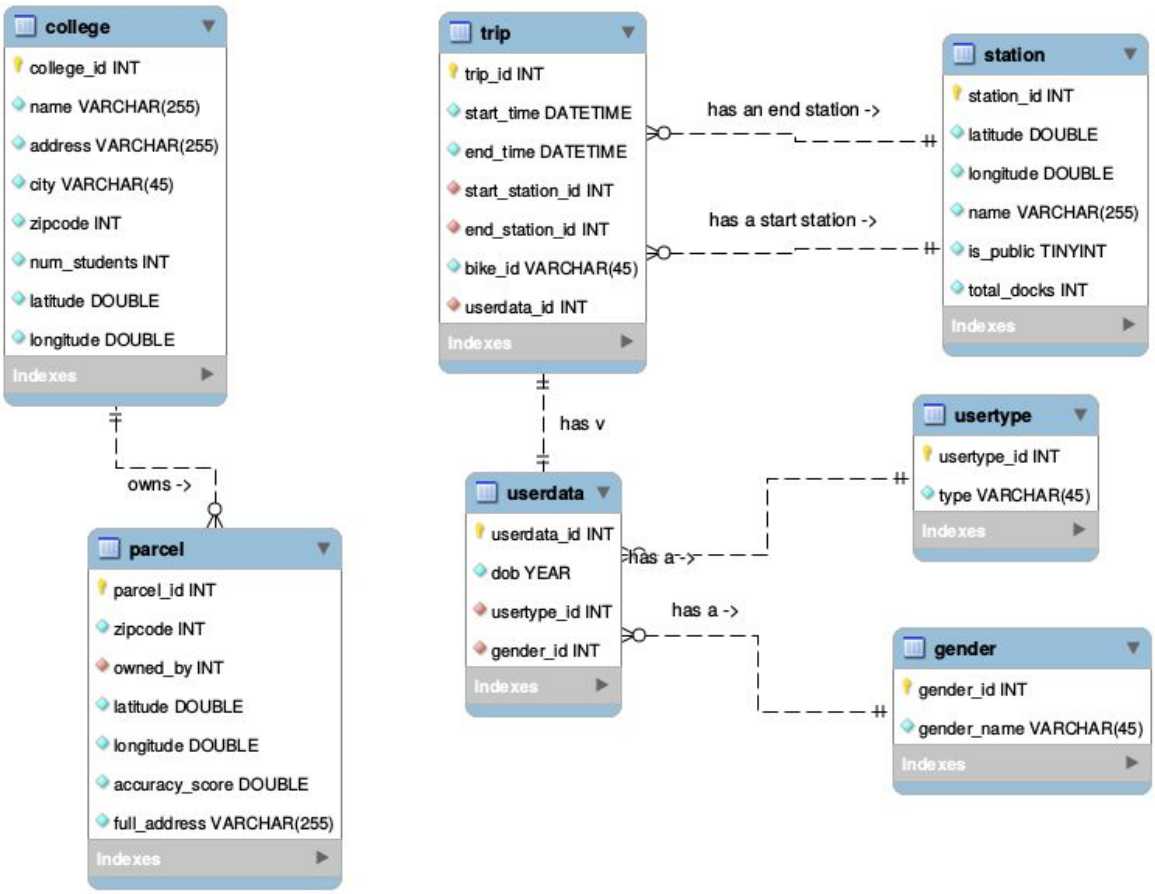

Figure 1: EER Diagram of Database

The non-key entities include the dictionary tables Gender and Usertype, and the table Userdata. Gender and Usertype are a product of normalization. It is interesting to note that the original values for "gender" from the trip data were one of the numbers 0, 1, and 2. There were no other indicators about what these might correspond to. I assigned them values based partially on their counts and the United States being a traditionally patriarchal society. Because 0 had the least values associated with it, it was assumed that meant undefined/not specified. This could also mean "other" to be inclusive of non-binary people, but it is rare that a company does this. Then it was assumed that 1 was "male" and 2 was "female". This is usually how people who grow up in a patriarchal society order gender. Because of all these assumptions about what the gender values meant, this report does not try to draw any conclusions based on gender.

Usertype is a straightforward table with the values "Subscriber" and "Customer". Subscriber probably means that the user has an account with Bluebikes rather than being an occasional user.

Userdata is the collection of fields about the user of a Trip. This is a separate entity in the ER Diagram, but is not a separate table in the actual database. This is due to the one-to-one

relationship between Userdata and Trip. Because the Userdata does not contain unique data for each user (birth year, usertype, and gender is not enough for uniqueness), this database cannot specify different users and associate them with multiple Trips.

Data Sources and Methods

The Bluebike trip data for July 2019, September 2019, and stations was pulled from the Bluebikes system data page on their website[1]. July and September of 2019 were used because colleges generally have more students taking classes in September. August was not used because some colleges start their fall semesters in August, but others start in early September. This database does not contain data for other months or years because of storage and time constraints. A possible expansion of this project would be analyzing if the trends persist over different years. It would also be interesting to compare with Cambridge colleges and universities, but the list of schools from Analyze Boston only contained schools in the city of Boston [2].

The data on colleges and parcel ownership in Boston was gathered from the Analyze Boston website [2, 3]. The Bluebike data used latitude and longitude values to indicate location of their stations. The parcel data used street addresses instead. To allow comparison, the author decided the parcel data should be converted to use latitude and longitude. There are many online services for this task, but the 166,044 parcels in the original dataset would have costed around one hundred dollars to convert. Instead of paying this fee, the parcel dataset was trimmed down to a new table using the following query:

```
create table colleges_parcels as
select
    parcels_2017.ZIPCODE,
    parcels_2017.owner,
    parcels_2017.full_address
from parcels_2017
join colleges_boston on parcels_2017.owner like concat("%", colleges_boston.Name, "%");
```

Figure 2: Query used to filter the parcels

This reduced the number of parcels down to a much more affordable 495 entries. This was within the maximum for free geocoding on the site Geocodio, so this table was exported as a csv and fed through the website [4]. Geocodio returns the table you gave it with the additional columns Latitude, Longitude, Accuracy Score, and Accuracy Type. Accuracy score is a double value from 0 to 1 indicating how accurate the latitude and longitude are expected to be based on the Accuracy Type. The value of 1 is the most accurate the site can provide.

Once all the information was available in an “exploration” database, a migration script was written to transform the source data to the desired database design described above. The database should be reconstructable from the database diagram and the notes above, but the author has placed the migration script on github for access by interested parties [5].

User Cases

The author has defined a few helper functions and procedures before performing queries. The first is a function for computing the distance in meters between two latitude and longitude values seen in Figure 3. This was written based on the JavaScript pseudocode on Movable-Type's website [6].

```
-- function for calculating the distance between two lat, long pairs
create function get_dist(lat1 double, long1 double, lat2 double, long2 double)
returns double -- dist in meters
deterministic
begin
    -- using the haversine formula to calculate distance as the crow flies over the earth's surface
    declare earth_radius double;
    declare lat1_rads double;
    declare lat2_rads double;
    declare delta_lat double;
    declare delta_long double;
    declare a double;
    declare c double;
    declare dist double;

    set earth_radius = 6371000; -- meters

    -- mult by pi and divide by 180 to convert to radians
    set lat1_rads = lat1 * pi() / 180;
    set lat2_rads = lat2 * pi() / 180;

    set delta_lat = lat2_rads - lat1_rads;
    set delta_long = (long2 - long1) * pi() / 180;

    set a = (sin(delta_lat / 2) * sin(delta_lat / 2))
        + (cos(lat1_rads) * cos(lat2_rads) * sin(delta_long / 2) * sin(delta_long / 2));

    set c = 2 * atan2(sqrt(a), sqrt(1-a));

    return earth_radius * c;

end //
```

Figure 3: Haversine distance function

The author also created two procedures that use this function for ease of use as well as query demonstrations. The first procedure, in Figure 4, creates a "near_parcel_result" table of all the Bluebike stations that are within a given maximum distance of a given parcel. The other procedure was similar, but generated a "near_university_result" table of Bluebike stations within a given maximum distance of a given university (Figure 5).

```
-- procedure to get nearby blue bikes stations to given parcel id
-- max_dist in meters
create procedure near_parcel(parcel_id int, max_dist double)
begin
    declare parcel_lat double;
    declare parcel_long double;


    set parcel_lat = (select latitude from parcel where parcel.parcel_id = parcel_id);
    set parcel_long = (select longitude from parcel where parcel.parcel_id = parcel_id);

    drop table if exists near_parcel_result;
    create table near_parcel_result (
        station_id int,
        name varchar(255),
        latitude double,
        longitude double,
        is_public tinyint,
        total_docks int
    ) as select *
        from station
        where get_dist(parcel_lat, parcel_long, station.latitude, station.longitude) <= max_dist;
end //
```

Figure 4: Procedure for generating a table of Bluebike stations near a given parcel

```
-- procedure to get nearby blue bikes stations for given university id
create procedure near_university(university_id int, max_dist double)
begin
    drop table if exists near_university_result;
    create table near_university_result (
        parcel_id int,
        station_id int,
        station_name varchar(255),
        station_latitude double,
        station_longitude double,

        constraint near_uni_fk_parcel
            foreign key (parcel_id)
            references parcel (parcel_id),
        constraint near_uni_fk_station
            foreign key (station_id)
            references station (station_id)
    ) as
        select
            parcel_id,
            station_id,
            station.name as station_name,
            station.latitude as station_latitude,
            station.longitude as station_longitude
        from parcel
        left join station on get_dist(parcel.latitude, parcel.longitude, station.latitude, station.longitude) <= max_dist
        where owned_by = university_id and accuracy_score > 0.5;
end //
```

Figure 5: Procedure for generating a table of Bluebike stations near a given university

The first query performed asked how many parcels the database had for each university (see Figure 6). This was important for the subsequent queries because the number of parcels affects how many Bluebike stations are nearby. The results are shown in Figure 7. Interestingly, there are not that many parcels owned by schools other than Northeastern. The author is not sure this is 100% accurate, but it is the data that the city of Boston provides. Because there are significantly more entries for Northeastern, the results of the rest of the queries are likely the most accurate for Northeastern.

```sql
-- How many parcels are there for each university? Ordered by most parcels to least.
select college.name, count(*) as 'parcel_count'
from parcel
join college on parcel.owned_by = college_id
group by owned_by
order by parcel_count desc;
```

Figure 6: SQL query for the number of parcels owned by each college

name	parcel_count
Northeastern University	57
Berklee College of Music	6
Suffolk University	6
Boston University	6
Fisher College	6
Wheelock College	4
Boston University Trustees	3
Simmons College	3
New England School of Law	3
Boston Baptist College	2
Boston Architectural College	2
The Boston Conservatory	1
MCPHS University	1

Figure 7: Table of parcel numbers for each college and university

The next query asked what Bluebike stations were within 100 meters (328 feet) of Northeastern's parcels (Figure 7). This used the previously defined procedure with the college_id for Northeastern and the maximum distance of 100.

```sql
-- What are the very nearby bluebike stations for Northeastern (within 100 meters of an owned parcel)?
call near_university((select college_id from college where college.name = "Northeastern University"), 100);
select distinct station_id, station_name, station_latitude, station_longitude from near_university_result;
```

Figure 7: Query for nearby Bluebike stations to Northeastern

station_id	station_name	station_latitude	station_longitude
139	Northeastern University - North Parking Lot	42.341814	-71.090179
157	Ruggles T Stop - Columbus Ave at Melnea Cass Blvd	42.33624445	-71.08798563

Figure 8: Table of nearby Bluebike stations to Northeastern

The next query asked how many trips do these stations from Figure 7 see over both July and August. This used the "near_university_result" table created by the procedure to count trips in the trip table (see Figure 9).

```
-- How many trips do these blue bike stations see during BOTH july and august (ie the whole trip table)?
select near_university_result.station_name, count(*) as 'num_trips'
from trip
join near_university_result on near_university_result.station_id in (trip.start_station_id, trip.end_station_id)
group by near_university_result.station_name
order by num_trips desc;
```

Figure 9: Query for how many trips each station near Northeastern had

station_name	num_trips
Northeastern University - North Parking Lot	60312
Ruggles T Stop - Columbus Ave at Melnea Cass Blvd	29022

Figure 10: Results table for how many trips each station near Northeastern had

Looking at the same stations again, the next query separated July and September trips (Figure 11).

```
-- How many trips do these blue bike stations see during July vs September (separate counts for each month)
select
    near_university_result.station_name,
    sum(case when month(trip.start_time) = 7 then 1 else 0 end) as count_july,
    sum(case when month(trip.start_time) = 9 then 1 else 0 end) as count_september
from trip
join near_university_result on near_university_result.station_id in (trip.start_station_id, trip.end_station_id)
group by near_university_result.station_name;
```

Figure 11: Query for comparing July and September rides at Northeastern stations

station_name	count_july	count_september
Northeastern University - North Parking Lot	25984	34328
Ruggles T Stop - Columbus Ave at Melnea Cass Blvd	10395	18627

Figure 12: Results table for how many trips each station near Northeastern had in July vs September

This shows a significant increase in rides between July and September. The next query shows these values for all the universities with nearby Bluebike stations. Birth year was filtered to be greater than 1995 to, hopefully, get more students than commuters. The usertype was also restricted to "Subscriber" to try to exclude tourists from the data. All of these restrictions are in an attempt to make the data more accurate, but these cannot be confirmed to be good choices since the user data about Bluebikes trips are limited to these parameters and gender. Figure 13 shows the query used to get this data. Figures 14 and 16 show the results for stations within 150

meters of a college and within 400 meters of a college respectively. Included in these tables are the number of students at each school for perspective of scale. Figures 15 and 17 show these results as graphs for easier visualization.

```
select * from (
    select
        stations_near_unis.college_name,
        sum(case when month(trip.start_time) = 7 then 1 else 0 end) as 'count_july',
        sum(case when month(trip.start_time) = 9 then 1 else 0 end) as 'count_september',
        stations_near_unis.num_students
    from trip
    join (
        select distinct
            college.name as 'college_name',
            college.num_students,
            station_id,
            station.latitude as station_latitude,
            station.longitude as station_longitude
        from parcel
        join station on get_dist(parcel.latitude, parcel.longitude, station.latitude, station.longitude) <= 400
        join college on owned_by = college.college_id
        where accuracy_score > 0.5
    ) stations_near_unis on stations_near_unis.station_id in (trip.start_station_id, trip.end_station_id)
    where trip.dob > 1995
        and stations_near_unis.num_students > 0
        and (select usertype.usertype from usertype where trip.usertype_id = usertype.usertype_id) = "Subscriber"
    group by stations_near_unis.college_name, stations_near_unis.num_students
)
trip_counts order by count_september - count_july desc;
```

Figure 13: Query for showing colleges' July and September Bluebike station usage

College	July	September	Number of Students
Northeastern University	2093	4490	27537
Berklee College of Music	1711	2263	4145
Boston Architectural College	702	881	1406
New England School of Law	128	181	1096
Fisher College	333	358	1593
Suffolk University	1190	1038	9148
Boston University	1308	1023	31960

Figure 14: Results table for Bluebike station usage for stations within 150 meters of a college in July and September

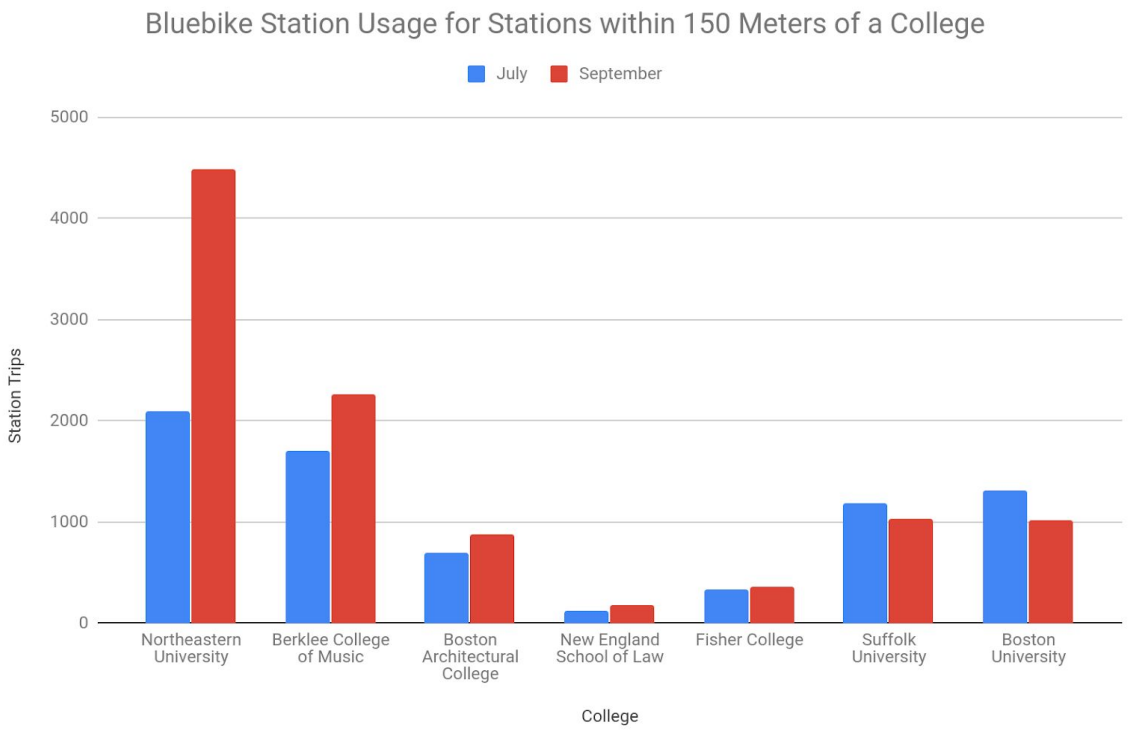

Figure 15: Results bar graph for Bluebike station usage for stations within 150 meters of a college in July and September

College	July	September	Number of Students
Northeastern University	5622	8287	27537
Boston University	7402	9805	31960
The Boston Conservatory	1886	3714	682
Boston Architectural College	2424	3150	1406
MCPHS University	1930	2243	4252
Berklee College of Music	4432	4743	4145
New England School of Law	1288	1556	1096
Wheelock College	1233	1427	1055
Fisher College	1388	1270	1593
Suffolk University	3149	2992	9148

Figure 16: Results table for Bluebike station usage for stations within 400 meters of a college in July and September

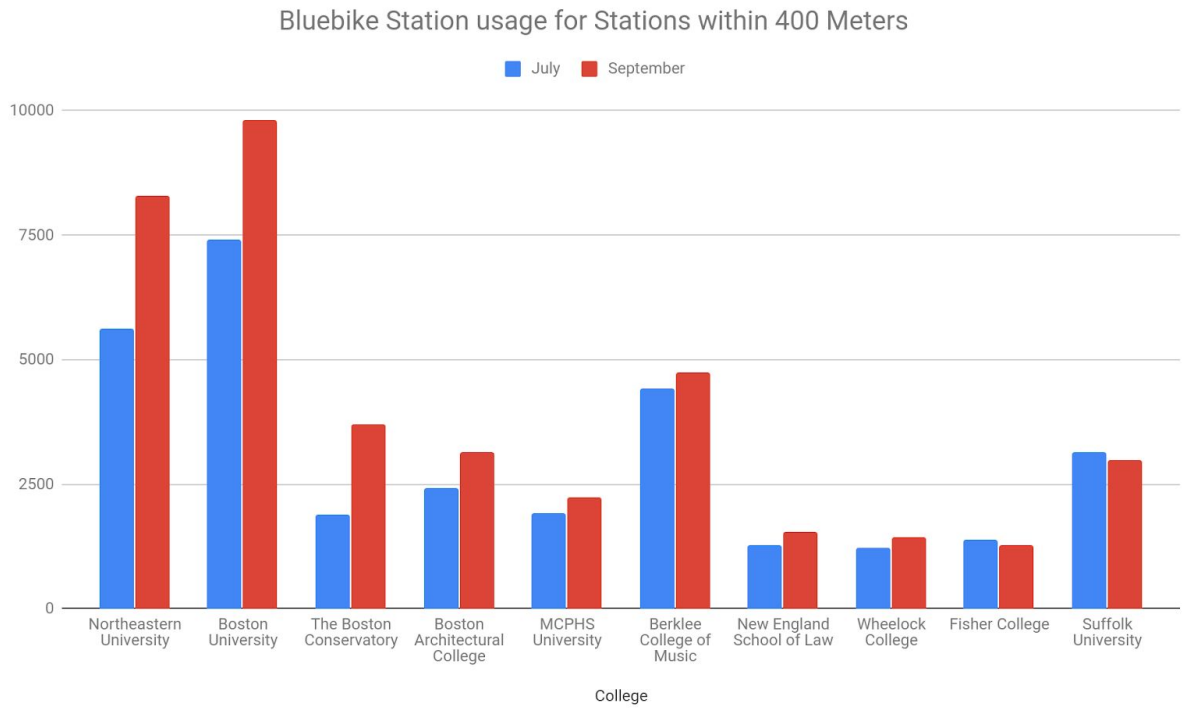

Figure 17: Results bar graph for Bluebike station usage for stations within 400 meters of a college in July and September

University students seem to have a significant impact on Bluebike usage. This depends on the university, but Northeastern students seem to have a large impact on the Bluebike system. Boston University has a large impact when the maximum distance to a station is 400 meters (~0.25 miles), which could be because their campus is more spread out than Northeastern's. Other Boston universities and colleges seem to have smaller impacts, likely due to having significantly less students than Northeastern and Boston University. The Boston Conservatory has a large increase in bike usage for stations within 400 meters, but this might overlap with Northeastern's stations due to the larger distance.

Other interesting queries might graph the usage rates as the stations get further from colleges. Or it could be examined if stations that are close to colleges generally get more use.

Conclusions

University students seem to have a large impact on nearby Bluebike station usage. This could be concluded more concretely if the Bluebike trip data had more information about their users. However the restrictions of born after 1995 and usertype "Subscriber" hopefully narrowed down the trip data to those from students.

Limitations of this project include only looking at July and September of 2019, seemingly incomplete parcel data, and guesses about user data relations to actual users.

References

1. Bluebikes. 2020. *Bluebikes System Data*. [online] Available at: <https://www.bluebikes.com/system-data> [Accessed 18 June 2020].

2. Analyze Boston. 2020. *Colleges And Universities*. [online] Available at: <https://data.boston.gov/dataset/colleges-and-universities> [Accessed 18 June 2020].

3. Analyze Boston. 2020. *Parcels 2017 Data Lite*. [online] Available at: <https://data.boston.gov/dataset/parcels-2017-data-lite> [Accessed 18 June 2020].

4. Geocodio. 2020. *Hassle-Free Geocoding*. [online] Available at: <https://www.geocod.io/> [Accessed 18 June 2020].

5. Github. 2020. *Database Final Project - Scripts - Migration Script*. [online] Available at: <https://github.com/kodonnell327/database_final_project/blob/master/scripts/migration_script.sql> [Accessed 18 June 2020].

6. Movable Type Scripts. 2020. *Calculate Distance And Bearing Between Two Latitude/Longitude Points Using Haversine Formula In Javascript*. [online] Available at: <https://www.movable-type.co.uk/scripts/latlong.html> [Accessed 18 June 2020].