Lab 1

Carmen Canedo

About

For this lab, I used the GPS tracking app on my iPhone, myTracker, to trace myself walking a straight path outside of my house in Nashville, TN. I tried to maintain a steady pace as I walked up and down the street.

Below you will find an analysis of my path that was created using R statistical software. My code is available below, but my step by step process can be found on my GitHub page.

Necessary Packages

```
library(mdsr)
library(XML)
library(OpenStreetMap)
library(lubridate)
library(ggmap)
library(raster)
```

```
## Warning: package 'sp' was built under R version 3.6.2
```

```
library(sp)
```

Getting the data ready

Loading in data

```
walking <- read.csv("lab-1.csv", header = TRUE)
```

Cleaning data

Getting rid of unnecessary columns

```
walking <- walking %>%
  dplyr::select(-type, -desc, -name)
```

Making column names simpler

```
walking <- walking %>%
  rename(altitude = altitude..ft.) %>%
  rename(speed = speed..mph.) %>%
  rename(distance_mi = distance..mi.) %>%
  rename(distance_int_ft = distance_interval..ft.)
```

Calculating Summary and Linear Regression model

Summary stats

We will be working with a full population

```
sum_latitude <- favstats( ~ latitude, data = walking)

sum_longitude <- favstats( ~ longitude, data = walking)

sum_altitude <- favstats( ~ altitude, data = walking)

sum_speed <- favstats( ~ speed, data = walking)

sum_distance_mi <- favstats( ~ distance_mi, data = walking)

sum_dist_int_ft <- favstats( ~ distance_int_ft, data = walking)

# Results
sum_latitude
```

```
##       min       Q1  median       Q3      max    mean           sd   n
##  36.18001 36.18029 36.1806 36.18091 36.18117 36.1806 0.0003514294 221
##  missing
##        0
```

```
sum_longitude
```

```
##        min        Q1   median        Q3       max     mean           sd
##  -86.74297 -86.74294 -86.7429 -86.74286 -86.74278 -86.7429 4.645446e-05
##    n missing
##  221       0
```

```
sum_altitude
```

```
##    min    Q1 median    Q3   max     mean       sd   n missing
##  500.8 505.8  511.2 512.4 516.1 509.3181 4.041696 221       0
```

```
sum_speed
```

```
##  min    Q1 median    Q3 max     mean       sd   n missing
##    0 2.275    2.7 3.325  10 2.839545 1.226507 220       1
```

```
sum_distance_mi
```

```
##  min    Q1 median    Q3   max       mean         sd   n missing
##    0 0.047   0.09 0.137 0.181 0.09093213 0.05260591 221       0
```

```r
sum_dist_int_ft
```

```
##   min   Q1 median   Q3   max    mean       sd   n missing
##     0 3.14   4.01 5.29 14.58 4.32362 1.937173 221       0
```

Analysis

#####Question 1: The standard deviation is larger for latitude.

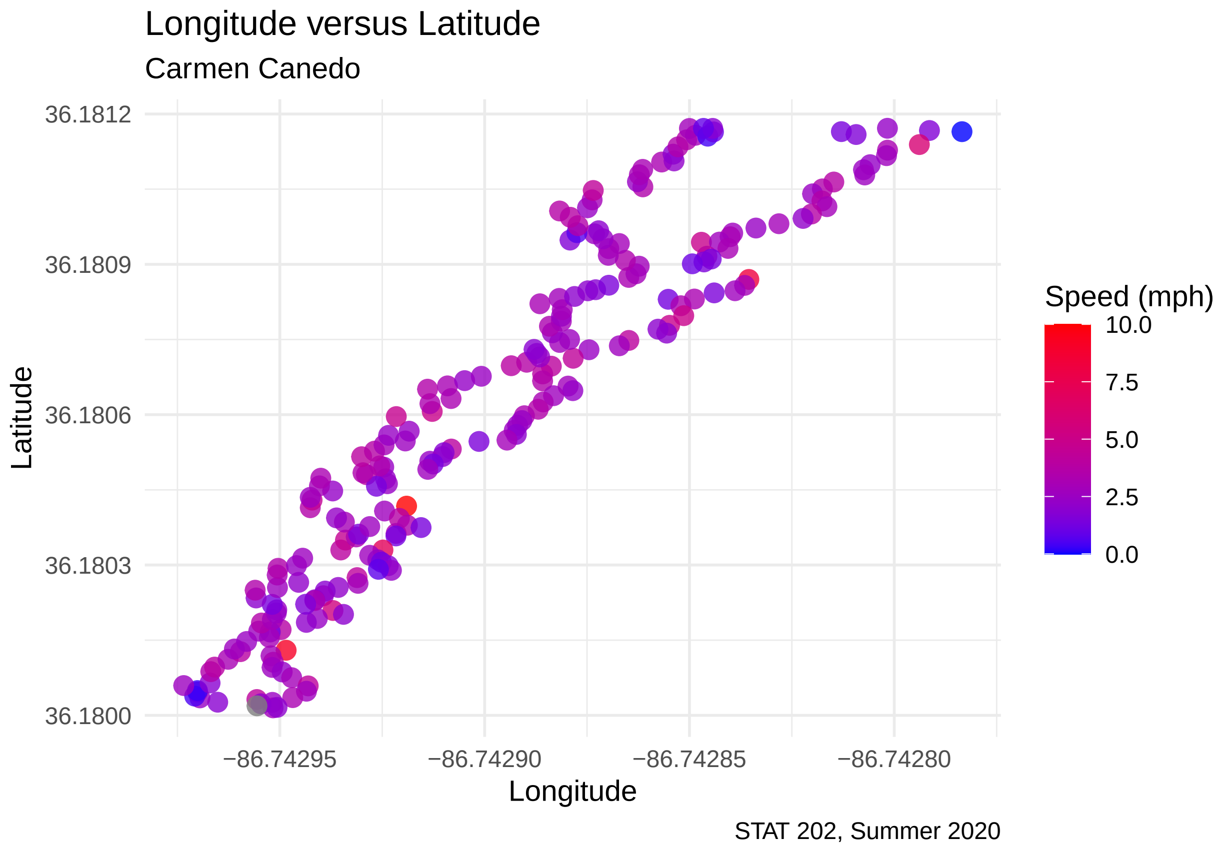
#####Question 2: This tells us that the latitude moves farther from the mean latitude.

Creating Latitude v. Longitude Scatter Plot

x = longitude y = latitude group by = speed color scheme = red to blue scale

```r
lat_v_long <- walking %>%
  ggplot(aes(x = longitude, y = latitude)) +
  geom_point(alpha = 0.8, aes(color = speed), size = 3) +
  scale_color_gradient(low = "blue", high = "red") +
  theme_minimal() +
  labs(title = "Longitude versus Latitude",
       subtitle = "Carmen Canedo",
       caption = "STAT 202, Summer 2020",
       x = "Longitude",
       y = "Latitude",
       color = "Speed (mph)")

lat_v_long
```



Adding Line of Best Fit

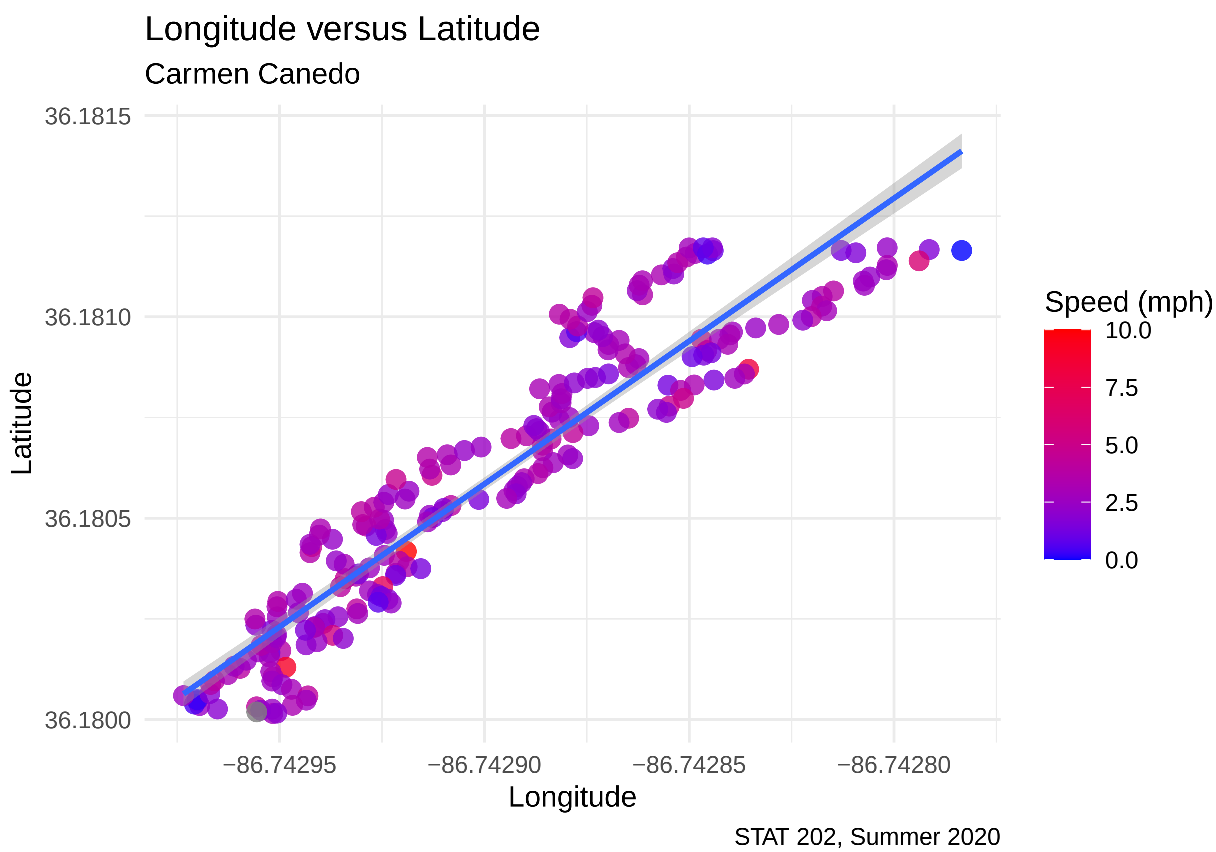
x = longitude y = latitude

Let's calculate the linear regression model

```
lat_v_long <- lat_v_long +
  geom_smooth(method = "lm")

lat_v_long
```

```
## 'geom_smooth()' using formula 'y ~ x'
```



Simple Linear Regression Results

```
# Calculating model
model <- lm(latitude ~ longitude, data = walking)

# Finding correlation coefficient
coef(model)
```

```
## (Intercept)   longitude
##  651.652871    7.095362
```

Formula for line of best fit $latitude = 651.653 + 7.0954(longitude)$

Analysis

- Equation for regression line and the correlation coefficient
- Is the line of best fit a good tool to estimate the path traveled? Why or why not?
- How does the correlation help you answer part b?

Mapping the route

I followed exercises from here

Getting the data

In order to ensure that all the values work when mapped, this equation places the vectors correctly.

```
# Function to shift vectors
shift_vec <- function(vector, shift) {
  if (length(vec) <= abs(shift)) {
    rep(NA, length(vec))
  } else {
    if (shift >= 0) {
      c(rep(NA, shift), vec[1:(length(vec) - shift)])
    } else {
      c(vec[(abs(shift) + 1):length(vec)])
    }
  }
}
```

Reading in GPX file

```
options(digits = 10)

# Parsing the GPX file
parsed_file <- htmlTreeParse(file = "lab-1-raw-data.gpx", error = function(...) {}, useInternalNodes =

# Get all times and coordinates via the respective xpath
times <- xpathSApply(parsed_file, path = "//trkpt/time", xmlValue)
coords <- xpathSApply(parsed_file, path = "//trkpt", xmlAttrs)

# Extract latitude and longitude from the coordinates
lats <- as.numeric(coords["lat",])
lons <- as.numeric(coords["lon",])
```

Putting values into dataframe

This allows us to have all of the GPX file in one place, ready to be placed onto a map.

```
geodf <- data.frame(lat = lats, lon = lons, time = times)
```

Querying map background

I used my Google API to access the static map used below.

```
street <- get_map(location = "409 N 15th St., Lockeland Springs, Nashville, Tennessee",
                  zoom = 19,
                  maptype = "roadmap")
```

Creating the map

```
# Plotting points
path <- ggmap(street) +
  geom_point(data = geodf,
             aes(x = lon, y = lat),
             size = 1,
             alpha = 0.7,
             color = "red")

# Adding details
path <- path +
  labs(x = "Longitude", y = "Latitude", title = "Walking Path Plotted using myTracks", subtitle = "Carme

path
```