Lab 1

Carmen Canedo

May 23, 2020

# 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](https://github.com/carmen-canedo/stat-202-lab1).

# Necessary Packages

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

# Summary stats calculations

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

# Results  
sum\_latitude

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

sum\_longitude

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

sum\_dist\_int\_ft

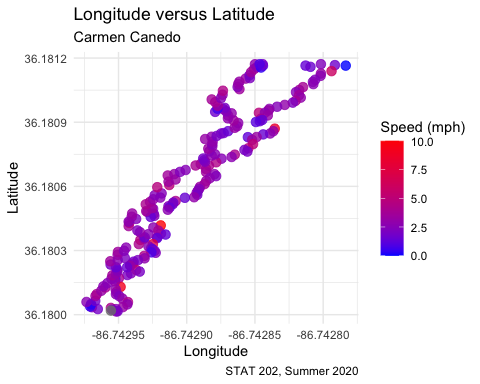
## min Q1 median Q3 max mean sd n missing  
## 0 3.14 4.01 5.29 14.58 4.32362 1.937173 221 0

### 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

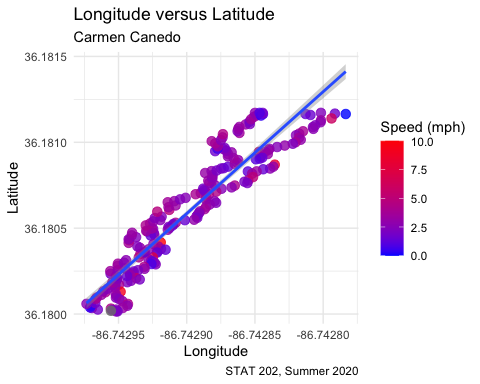
lat\_v\_long <- walking %>%  
 ggplot(aes(x = longitude, y = latitude)) +  
 geom\_point(alpha = 0.8, aes(color = speed), size = 3) +  
 scale\_color\_gradient(low = "blue", high = "red") +  
 theme\_minimal() +  
 labs(title = "Longitude versus Latitude",  
 subtitle = "Carmen Canedo",  
 caption = "STAT 202, Summer 2020",  
 x = "Longitude",  
 y = "Latitude",  
 color = "Speed (mph)")  
  
lat\_v\_long



## Adding Line of Best Fit

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:*

### Analysis

* 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 referenced exercises from [here](https://rpubs.com/ials2un/gpx1)

## 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

# Limits to 10 digits  
options(digits = 10)  
  
# Parsing the GPX file  
parsed\_file <- htmlTreeParse(file = "lab-1-raw-data.gpx",  
 error = function(...) {},  
 useInternalNodes = TRUE)  
  
# 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)

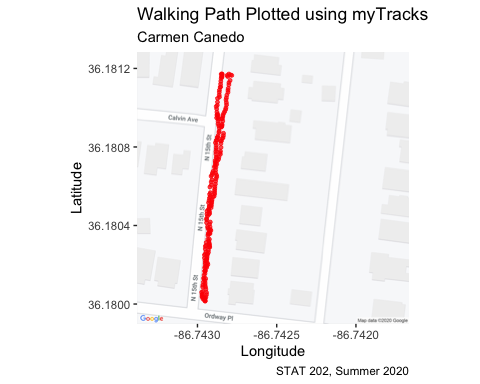
## Querying map background

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")

## Finished product

# 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 = "Carmen Canedo",  
 caption = "STAT 202, Summer 2020")  
  
path



# Conclusion

* What was learned
* Et ecetera