Project 2

Blinda Tian

11/1/2020

Open File

datainput 11 days

```
library(jsonlite)

dat_1 <- read_json('~/Downloads/gps/20200818114606.geojson')
dat_1 <- dat_1$features
dat_2 <- read_json('~/Downloads/gps/20200819132607.geojson')
dat_2 <- dat_2$features
dat_3 <- read_json('~/Downloads/gps/20200820151044.geojson')
dat_3 <- dat_3$features
dat_4 <- read_json('~/Downloads/gps/20200821111447.geojson')
dat_4 <- dat_4$features
dat_5 <- read_json('~/Downloads/gps/20200824130857.geojson')
dat_5 <- dat_5$features
dat_6 <- read_json('~/Downloads/gps/20200825121346.geojson')
dat_6 <- dat_6$features
dat_7 <- read_json('~/Downloads/gps/20200826131614.geojson')
dat_7 <- dat_7$features
dat_8 <- read_json('~/Downloads/gps/20200827113234.geojson')
dat_8 <- dat_8$features
dat_9 <- read_json('~/Downloads/gps/20200828122627.geojson')
dat_9 <- dat_9$features
dat_10 <- read_json('~/Downloads/gps/20200828130816.geojson')
dat_10 <- dat_10$features
dat_11 <- read_json('~/Downloads/gps/20200831115147.geojson')
dat_11 <- dat_11$features
```

Open and Conveetion data

1. variables in file include longitude & latitude , time(UTM), distance, time_long, accuracy, altitude, bearing, speed
2. function to grab time, longitude, latitude
3. function to remove dupliacates, then reorder time
4. function to covert UTM and timestamp
5. function to distance matrix between each stops

6. function to time difference (between two records) timetravel from start time, all in secs and speed 7 add days number (infact data 9 and data 10 is on the Same day)

Data Summarization

Timestamp_count Duration Start Pause Slocal Plocal Slocal Plocal startlong startlat

```
# data summarization 11days
dat_sum <-
list(dat_1,dat_2,dat_3,dat_4,dat_5,dat_6,dat_7,dat_8,dat_9,dat_10,dat_11)

dat_summarize <- c()
Timestamp_count <- c()
Duration <-c()
for(i in 1:length(dat_sum)){
  Timestamp_count[i] <- length(dat_sum[[i]][["time"]])
  Duration[i] <- sum(dat_sum[[i]]$Time_diff,na.rm=TRUE)
}
data_summarize <- data.frame(Timestamp_count,Duration)
data_summarize

##    Timestamp_count Duration
## 1               89 15393.15
## 2              225 16764.47
## 3              313 13103.17
## 4              604 25651.63
## 5              630 16054.08
## 6              652 14486.00
## 7              788 32539.00
## 8              654 20119.00
## 9              240  1925.00
## 10             365 17457.88
## 11             783 17719.84

# Time Start and Pause

library("countytimezones")
summary_grab <- function(dat_1){
  dat1_s <- format(as.POSIXct(dat_1$time[1], tz = "UTC","%Y-%m-
%dT%H:%M:%OS"),'%A, %B %d, %Y %H:%M:%S') #to UTM
  local_t <- calc_single_datetime(as.POSIXct(dat_1$time[1], tz = "UTC","%Y-
%m-%dT%H:%M:%OS"), tz = "MST") # to local time (MST)
  dat1_p <- format(as.POSIXct(tail(dat_1$time,1), tz = "UTC","%Y-%m-
%dT%H:%M:%OS"),'%A, %B %d, %Y %H:%M:%S') #to UTM
  local_p<- calc_single_datetime(as.POSIXct(tail(dat_1$time,1), tz =
"UTC","%Y-%m-%dT%H:%M:%OS"), tz = "MST") # to local time (MST)

# start location and stop location
  long_s <- (dat_1$longitude[1])
```

```
  long_p <- tail(dat_1$longitude,1)
  lat_s <- (dat_1$latitude[1])
  lat_p <- tail(dat_1$latitude,1)
  # return
    return(c(dat1_s,local_t,dat1_p,local_p,long_s,lat_s,long_p,lat_p))
}

dat1_s<- summary_grab(dat_1)
dat2_s<- summary_grab(dat_2)
dat3_s<- summary_grab(dat_3)
dat4_s<- summary_grab(dat_4)
dat5_s<- summary_grab(dat_5)
dat6_s<- summary_grab(dat_6)
dat7_s<- summary_grab(dat_7)
dat8_s<- summary_grab(dat_8)
dat9_s<- summary_grab(dat_9)
dat10_s<- summary_grab(dat_10)
dat11_s<- summary_grab(dat_11)



data_summarize$Start <-
c(dat1_s[1],dat2_s[1],dat3_s[1],dat4_s[1],dat5_s[1],dat6_s[1],dat7_s[1],dat8_
s[1],dat9_s[1],dat10_s[1],dat11_s[1])
data_summarize$Slocal <-
c(dat1_s[2],dat2_s[2],dat3_s[2],dat4_s[2],dat5_s[2],dat6_s[2],dat7_s[2],dat8_
s[2],dat9_s[2],dat10_s[2],dat11_s[2])
data_summarize$Pause<-
c(dat1_s[3],dat2_s[3],dat3_s[3],dat4_s[3],dat5_s[3],dat6_s[3],dat7_s[3],dat8_
s[3],dat9_s[3],dat10_s[3],dat11_s[3])
data_summarize$Plocal <-
c(dat1_s[4],dat2_s[4],dat3_s[4],dat4_s[4],dat5_s[4],dat6_s[4],dat7_s[4],dat8_
s[4],dat9_s[4],dat10_s[4],dat11_s[4])

data_summarize$startlong <-
c(dat1_s[5],dat2_s[5],dat3_s[5],dat4_s[5],dat5_s[5],dat6_s[5],dat7_s[5],dat8_
s[5],dat9_s[5],dat10_s[5],dat11_s[5])

data_summarize$startlat <-
c(dat1_s[6],dat2_s[6],dat3_s[6],dat4_s[6],dat5_s[6],dat6_s[6],dat7_s[6],dat8_
s[6],dat9_s[6],dat10_s[6],dat11_s[6])
data_summarize
```

```
##    Timestamp_count Duration                                  Start
Slocal
## 1               89 15393.15   Tuesday, August 18, 2020 17:50:40
20200818105040
## 2              225 16764.47 Wednesday, August 19, 2020 19:27:55
20200819122755
```

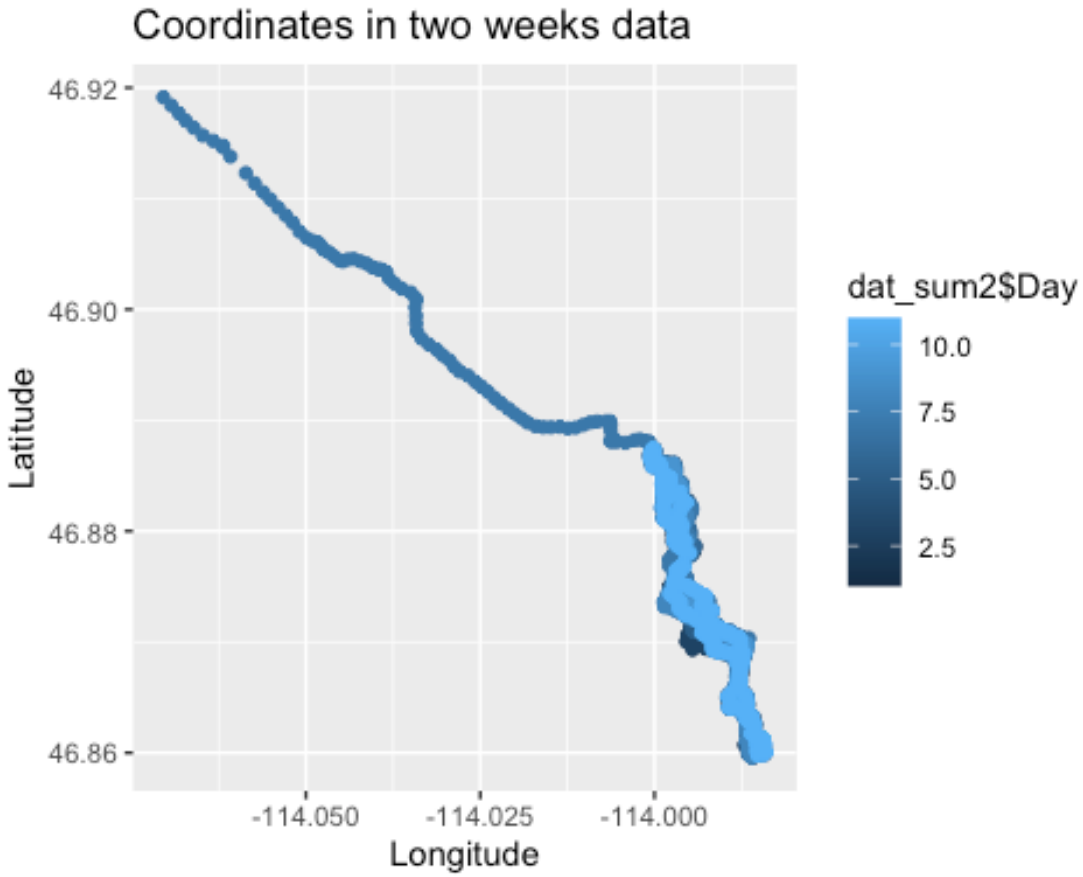
```
## 3              313 13103.17  Thursday, August 20, 2020 21:13:09
20200820141309
## 4              604 25651.63    Friday, August 21, 2020 17:17:21
20200821101721
## 5              630 16054.08    Monday, August 24, 2020 19:11:14
20200824121114
## 6              652 14486.00   Tuesday, August 25, 2020 18:15:31
20200825111531
## 7              788 32539.00 Wednesday, August 26, 2020 19:18:54
20200826121854
## 8              654 20119.00  Thursday, August 27, 2020 17:34:38
20200827103438
## 9              240  1925.00    Friday, August 28, 2020 18:28:33
20200828112833
## 10             365 17457.88    Friday, August 28, 2020 19:08:14
20200828120814
## 11             783 17719.84    Monday, August 31, 2020 17:53:35
20200831105335
##                                Pause         Plocal     startlong
startlat
## 1   Tuesday, August 18, 2020 22:07:14 20200818150714   -114.000521
46.8863239
## 2  Thursday, August 20, 2020 00:07:20 20200819170720  -114.0005151
46.8870326
## 3    Friday, August 21, 2020 00:51:33 20200820175133 -114.00013904
46.88713864
## 4  Saturday, August 22, 2020 00:24:53 20200821172453  -114.0007782
46.8868786
## 5    Monday, August 24, 2020 23:38:49 20200824163849    -114.00052
46.8872596
## 6   Tuesday, August 25, 2020 22:16:57 20200825151657  -114.0001423
46.88749925
## 7  Thursday, August 27, 2020 04:21:13 20200826212113 -114.00021406
46.88750954
## 8  Thursday, August 27, 2020 23:09:57 20200827160957 -114.00010499
46.88747926
## 9    Friday, August 28, 2020 19:00:38 20200828120038 -114.00011736
46.88747376
## 10   Friday, August 28, 2020 23:59:12 20200828165912  -113.9884352
46.8641179
## 11   Monday, August 31, 2020 22:48:55 20200831154855  -114.0002755
46.88748749
```

Description of data with visualization

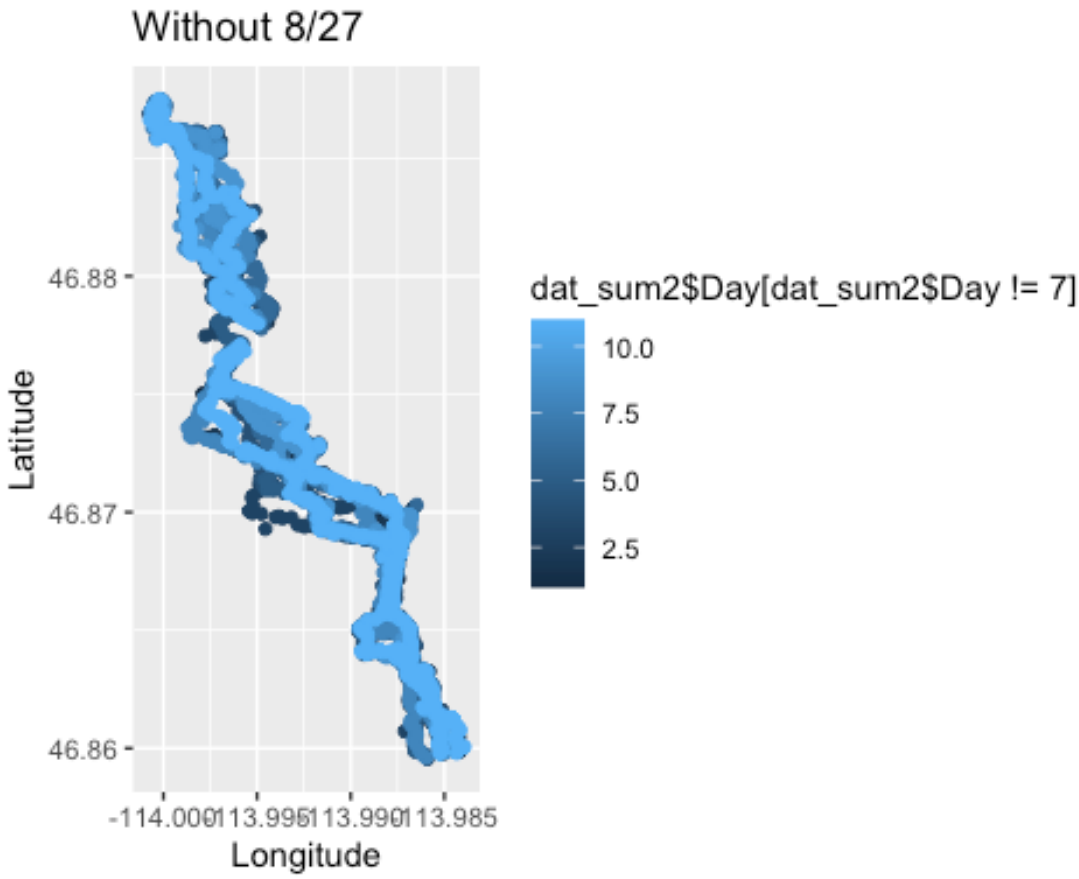
1. plot locations in the past two weeks (days diff cols)
2. spped over time

```
dat_sum2 <-
rbind(dat_1,dat_2,dat_3,dat_4,dat_5,dat_6,dat_7,dat_8,dat_9,dat_10,dat_11)
```

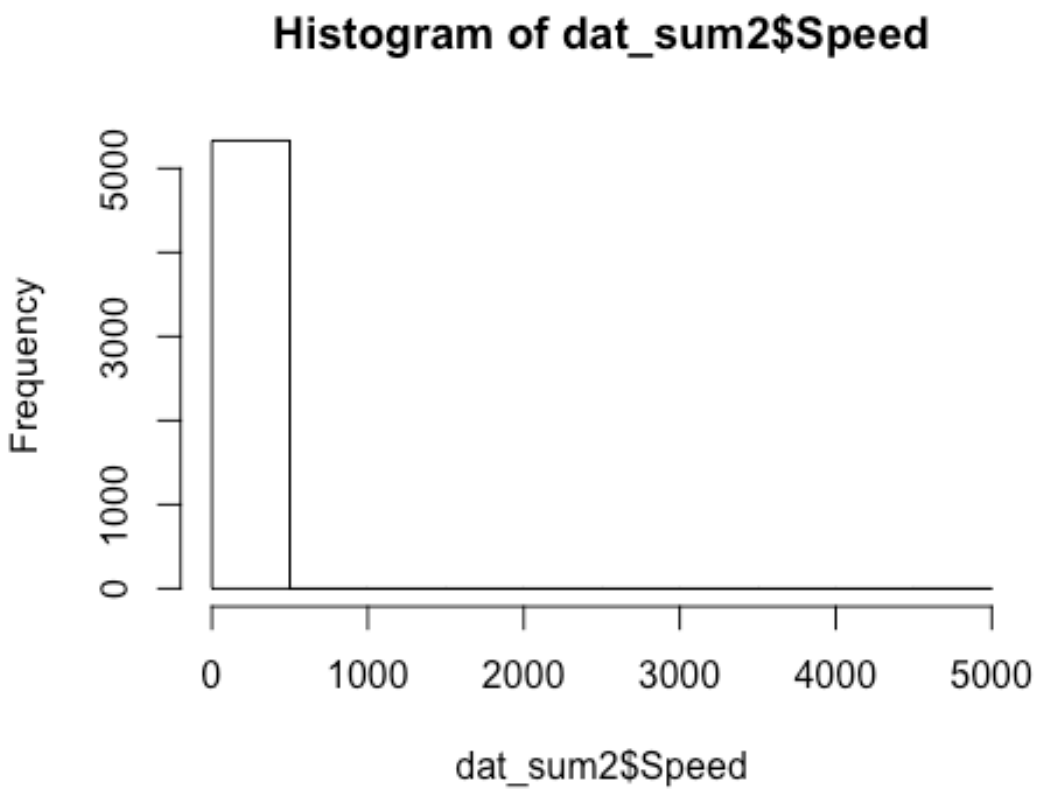
```r
# travel distance from the initial coordinates



# All records in 10days
library(ggplot2)
qplot(dat_sum2$longitude, dat_sum2$latitude, colour =  dat_sum2$Day,
      xlab="Longitude", ylab="Latitude", main="Coordinates in two weeks
data")
```



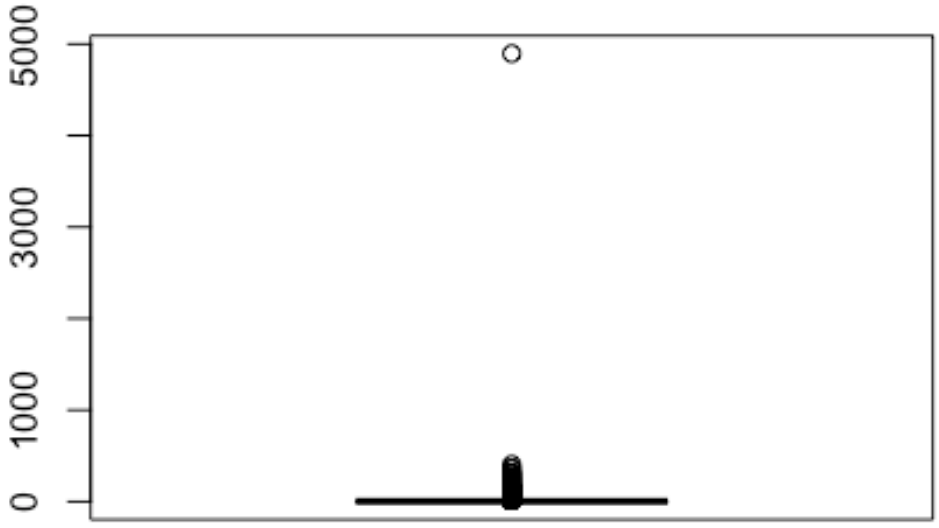

```r
# Zoom in to look at records, without 8/27 coordinates
library(ggplot2)
qplot(dat_sum2$longitude[dat_sum2$Day!=7],
      dat_sum2$latitude[dat_sum2$Day!=7],
      colour=dat_sum2$Day[dat_sum2$Day!=7],
      xlab="Longitude",ylab="Latitude", main="Without 8/27")
```



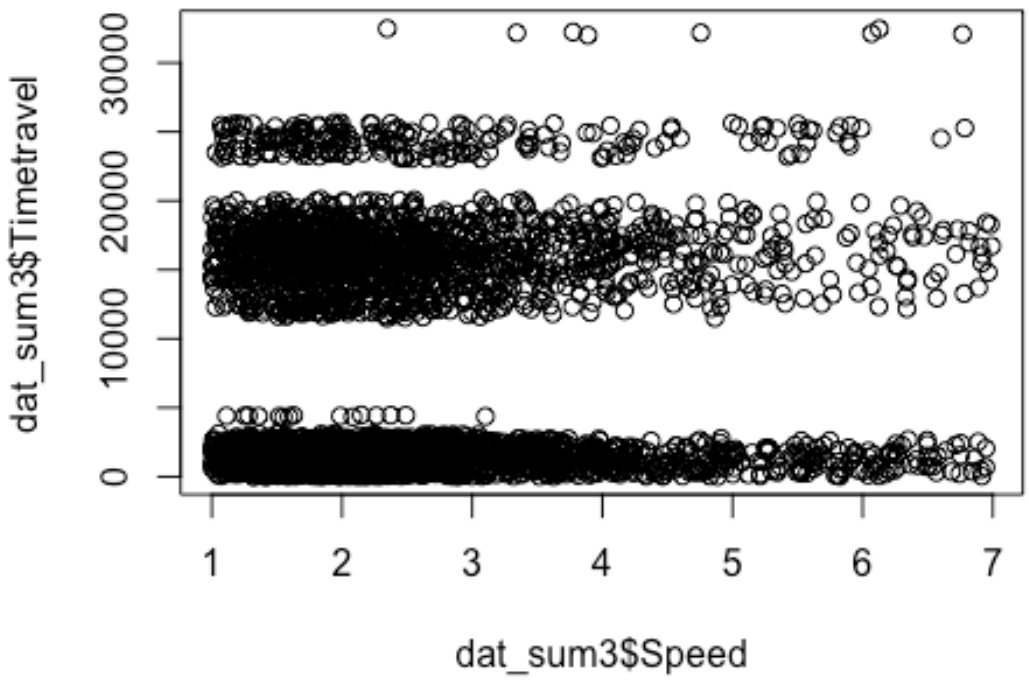

```
# Speed
par(2,4)

## [[1]]
## NULL
##
## [[2]]
## NULL

hist(dat_sum2$Speed)
```




```r
boxplot(dat_sum2$Speed)
```




```
  # take out stationary (<0),  extreme value (>30), and target speed will be
1-7
  #placed within 5 meters and 10 seconds to my position --> target walking &
cycling speeed


dat_sum3 <-dat_sum2[!dat_sum2$Speed>=7,]
dat_sum3 <- dat_sum3[!dat_sum3$Speed<=1,]

library(ggplot2)
p <- dat_sum3 %>%
  ggplot( aes(x=Timetravel, y=speed)) +
    geom_area(fill="#69b3a2", alpha=0.5) +
    geom_line(color="#69b3a2") +
    ylab("Speed Over Time")

plot(dat_sum3$Speed, dat_sum3$Timetravel)
```




```
#Bomb Place :
```

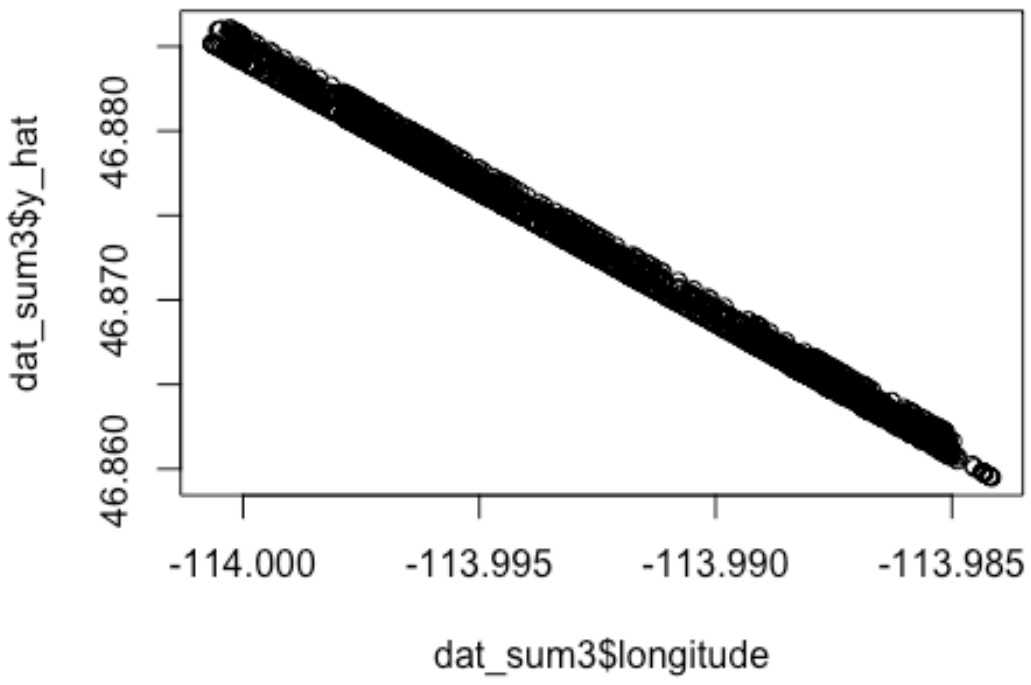
Model

Bomb Place : placed within 5 meters and 10 seconds to my position –> target walking & cycling speeed

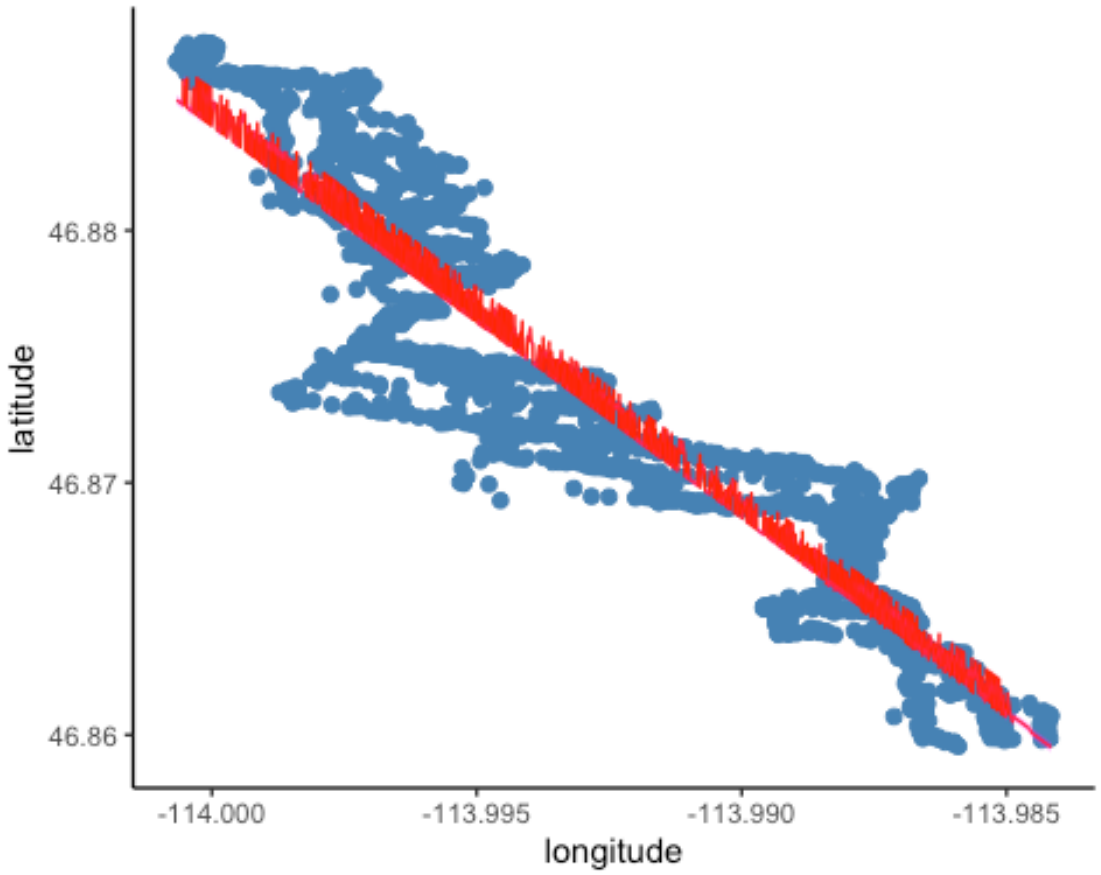
take out 8/27

```
# take out day 7
dat_sum3 <-dat_sum3[!dat_sum3$Day==7,]

dim(dat_sum3) # 2923  points

## [1] 2923    9

#
m1 <- lm( latitude ~ longitude + Timetravel , data = dat_sum3)
m1_predict <- predict(m1, newdata = dat_sum3, interval = "confidence")


dat_sum3$y_hat <- m1_predict[,1]
```

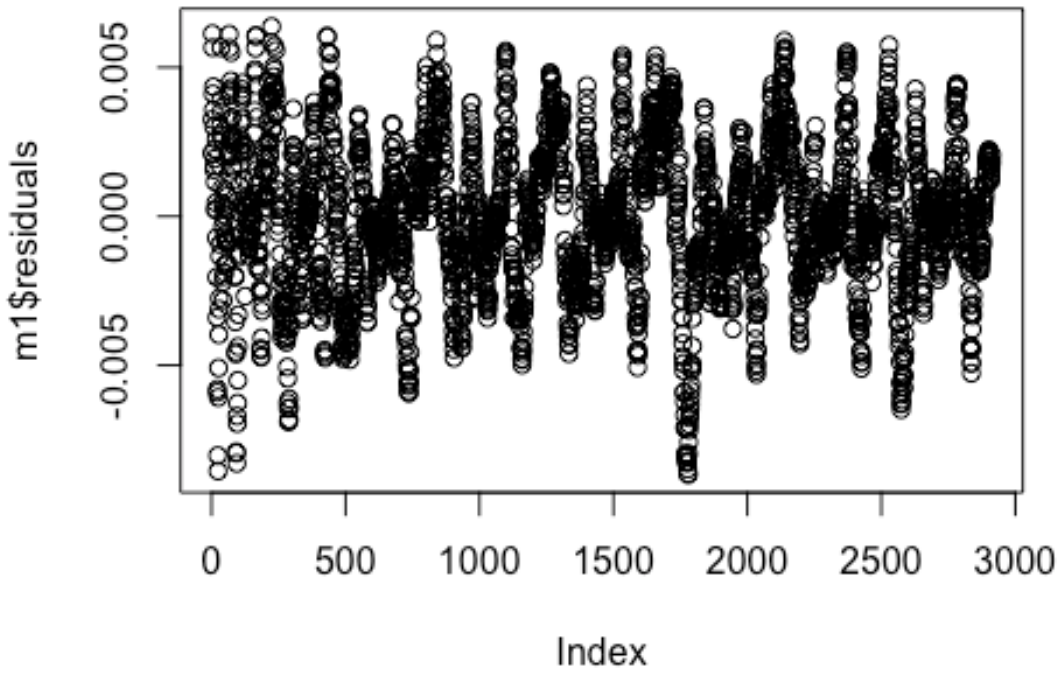
```
dat_sum3$y_lwr <- m1_predict[,2]
dat_sum3$y_upr <- m1_predict[,3]

plot(dat_sum3$longitude,dat_sum3$y_hat )
```



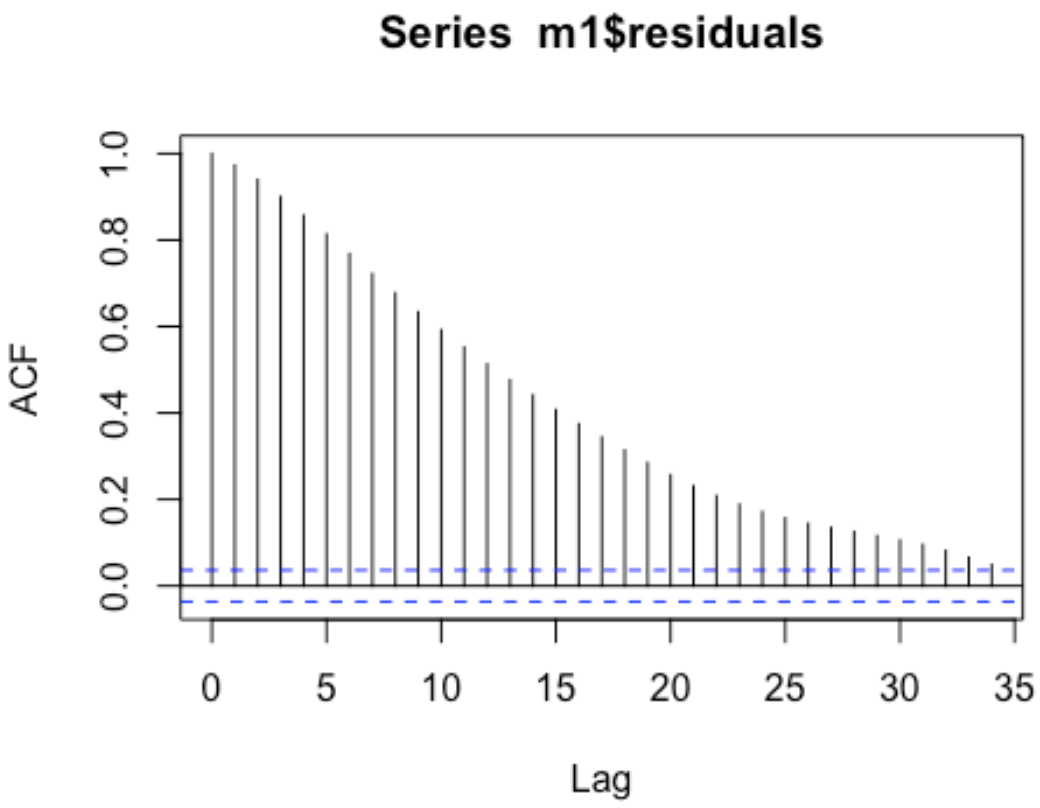

```
ggplot(data=dat_sum3, aes(longitude, latitude)) +
  geom_point(col="steelblue", size=2) +
  geom_line(aes(longitude, y_hat), col="red") +
  geom_ribbon(aes(ymin=y_lwr, ymax=y_upr), fill="magenta", alpha=.25) +
  theme(panel.grid.major = element_blank(), panel.grid.minor =
element_blank(),
        panel.background = element_blank(), axis.line = element_line(colour =
"black"))

## Warning: Removed 11 rows containing missing values (geom_point).

## Warning: Removed 11 row(s) containing missing values (geom_path).
```




```
plot(m1$residuals)
```



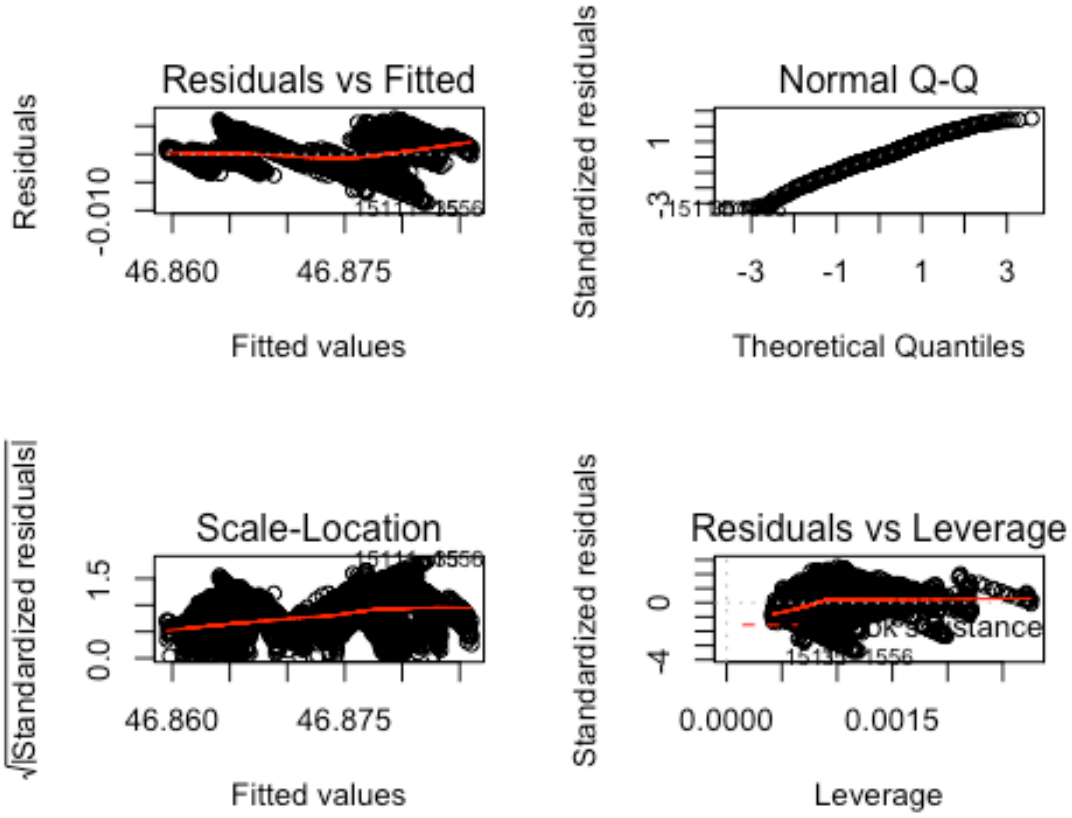

```r
acf(m1$residuals)
```




```
# The Deviance Residuals: shows us the variation in how far away our
observations are from the predicted values. Observations with a deviance
residual in excess of two may indicate lack of fit.

summary.lm(m1)

##
## Call:
## lm(formula = latitude ~ longitude + Timetravel, data = dat_sum3)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0086674 -0.0015452 -0.0000214  0.0017774  0.0063702
##
## Coefficients:
##               Estimate Std. Error  t value Pr(>|t|)
## (Intercept) -1.315e+02  1.142e+00 -115.169   <2e-16 ***
## longitude   -1.565e+00  1.002e-02 -156.224   <2e-16 ***
## Timetravel   5.736e-08  5.770e-09    9.942   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.002537 on 2909 degrees of freedom
```

```
##   (11 observations deleted due to missingness)
## Multiple R-squared:  0.8936, Adjusted R-squared:  0.8936
## F-statistic: 1.222e+04 on 2 and 2909 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2, 2))
plot(m1)
```




```
# check whether there is any residual spatial autocorrelation?
```

Model validation

Cross Validation validate the model with some independent data. Typically, we do this by removing a 20% of the data to act as a validation dataset, fitting the model on the remaining 80% and then predicting to the validation set.

```
# take 20% to act as validation set
dim(dat_sum3)
```

```
## [1] 2923   12
```

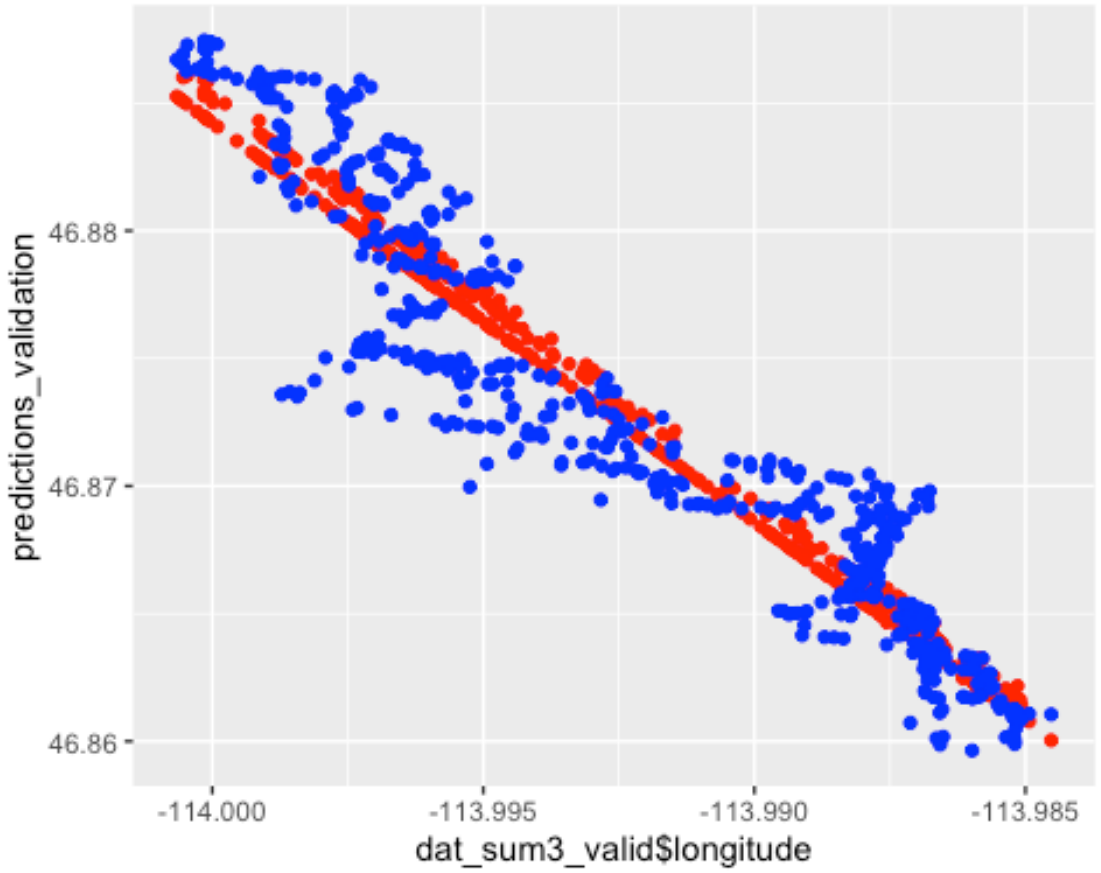
```
set.seed(1)
validation_rows <- sample(1:nrow(dat_sum3), 2923 *0.2)
dat_sum3_train <- dat_sum3[-validation_rows,] # 80%
```

```r
dat_sum3_valid <- dat_sum3[validation_rows,] #20%

# Fit model using 80%


m1_validation <- lm( latitude ~ longitude + Timetravel , data =
dat_sum3_train)

predictions_validation <- -1.324103e+02-
1.572759e+00*dat_sum3_valid$longitude+5.623e-08  *dat_sum3_valid$Timetravel


# Calculate mse
sqrt( mean( (dat_sum3_valid$latitude-predictions_validation)^2 , na.rm = TRUE
) )
```

```
## [1] 0.002548883
```

```r
#mse(dat_sum3_valid$latitude,predictions_validation,na.rm = TRUE)
ggplot() +
geom_point(aes(dat_sum3_valid$longitude,predictions_validation),col="red") +
geom_point(aes( dat_sum3_valid$longitude,dat_sum3_valid$latitude),col="Blue")
```

```
## Warning: Removed 1 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing missing values (geom_point).
```


predictions_validation
46.88
46.87
46.86
-114.000
-113.995
-113.990
-113.985
dat_sum3_valid$longitude