Wheresgeorge.com Mean Square Displacement

Nathan Williamson

April 7, 2018

Abstract

The intention of this project is to model the movement of paper currency bills throughout the United States using the data from the website www.wheresgeorge.com. The website allows users to input the serial number and location of dollar bills that they possess on a voluntary basis. Since the website was created in 1998 the site has collected over 200 million data points of time and location for dollar bills. We recieved a sample of this data from the website administrator. We intend to use this sample to build a model that predicts the movement of dollar bills.

To model the data we are assuming that the bills move according to particles in diffusion. We assume that the movement of the bills can be mathematically modelled by a random walk; a series of steps, one after the other, where each step is taken in a completely random direction from the one before. This kind of path can be modeled such that the mean squared distance traveled by a particle is proportional to the time elapsed. This relationship can be written as

$$< r^2 >= 6Dt + C$$

where $< r^2 >$ is the mean square distance, t is time and D and C are constants. If the data shows a clear linear relationship between $< r^2 >$ and t, then this model is a good fit for the data.

From previous research on this data set it is known that there is not sufficient linearity to justify this model. We believe that flight travel may be the key factor in skewing the data away from this linear relationship. We intend to separate the $< r^2 >$ variable into two categories: land and flight travel. For all data points where the distance traveled is likely due to a flight rather than land travel, we will use a value proportional to the frequency of passenger flights from the original location to the final location. By replacing these distance values for flight travel we believe we can restore linearity to the data.

Data

Our raw data looks like this:

```r
setwd("C:/Users/Nate/Desktop/wheresgeorge")
rawdata <- read.csv("RawData.csv", header=TRUE) #Read raw data into dataframe
head(rawdata)
```

```
##       bid       eid ent   zip            ts
## 1  827566    753472   1 60020  10/7/99 9:16
## 2  827566 253692813   2 60616  4/11/13 0:27
## 3 1902966   1881520   1 56301  5/2/12 18:12
## 4 1902966 252866888   2 60657 3/25/13 18:21
## 5 3211972   3305970   1 53208 8/14/12 18:16
## 6 3211972 259267815   2 60606  8/2/13 15:47
```

The columns of the data can be described as follows:

Bid: the serial number of the bill

Eid: a unique specific number given to all individual bill entries

Ent: integer marking the order of data points for each individual bill

Zip: zip code of the location of the bill

Ts: time stamp of when the entry was recorded

In order to do a mean square displacement we need the squared distance traveled and the time elasped between each data point for each bill. First the data is cleaned a bit. Then the zipcodes are changed to latitude and longitudes and the time stamps are changed into a time class.

```
#################### create time and distance data from raw data ######################



##### first we add the associated lat and long to the raw data with associated zip codes
setwd("C:/Users/Nate/Desktop/wheresgeorge")
rawdata <- read.csv("RawData.csv", header=TRUE)
library(zipcode) #use zipcode library
library(data.table)
data(zipcode)
rawdata$zip = clean.zipcodes(rawdata$zip) #clean non-US zip codes up with zipcode package
rawdata <- rawdata[!duplicated(rawdata$eid), ] #take out repeated rows
latlongdata = merge(rawdata, zipcode, by.x='zip', by.y='zip') #add lat and long to df
latlongdata <- latlongdata[order(latlongdata$bid), ] #reorder by bid data after merge
latlongdata = data.table(latlongdata) #Create data.table
latlongdata[, freq_bid := .N, by = bid] #Add column of frequencies of bid's
library(plyr)
latlongdata$ts <- as.Date(latlongdata$ts, format="%m/%d/%y") #Format time column to as.Date
head(latlongdata) #Preview cleaned data
```

```
##      zip     bid       eid ent         ts        city state latitude
## 1: 60020  827566    753472   1 1999-10-07    Fox Lake    IL 42.40944
## 2: 60616  827566 253692813   2 2013-04-11     Chicago    IL 41.84740
## 3: 56301 1902966   1881520   1 2012-05-02 Saint Cloud    MN 45.52607
## 4: 60657 1902966 252866888   2 2013-03-25     Chicago    IL 41.94083
## 5: 53208 3211972   3305970   1 2012-08-14   Milwaukee    WI 43.04786
## 6: 60606 3211972 259267815   2 2013-08-02     Chicago    IL 41.88258
##    longitude freq_bid
## 1: -88.17822        2
## 2: -87.63126        2
## 3: -94.20649        2
## 4: -87.65852        2
## 5: -87.96618        2
## 6: -87.63760        2
```

Next we perform our mean square displacement. Each bill is treated as an individual "particle" moving in space. For each bill all possible combinations of square distances and times are calculated between data points. For example, for a bill with 4 data points, there are $\binom{4}{2} = 6$ possible combinations between the data points. Square distances and times are calculated between each possible case and added to a new dataframe with the results.

```
#Calculate all combinations between groups of bids
library(data.table)
library(geosphere)
library(dplyr)
cumdata <- data.table(latlongdata, key = 'bid') #Use data.table library class
cumdata <- cumdata[cumdata, allow.cartesian = TRUE][ts < i.ts]#Prepare dataframe
cumdata[, t := i.ts - ts][, d := distHaversine(cbind(longitude, latitude),
                                    cbind(i.longitude, i.latitude))][, dsq := d*d]
#Add square distance and times to data
cumdata <- (subset(cumdata, cumdata$d>0))#Clean out zero values in distance
```

```
cumdatagraph <- select(as_data_frame(cumdata),t, d, dsq, freq_bid)#Select column for graph
cumdatagraph$t <-as.numeric(cumdatagraph$t)#Change time column class to numeric
head(cumdatagraph)#View sample of new dataframe
```

```
## # A tibble: 6 x 4
##       t          d          dsq freq_bid
##   <dbl>      <dbl>        <dbl>    <int>
## 1  4935  77159.802   5953635039        2
## 2   327 660457.013 436203465536        2
## 3   353 132494.756  17554860432        2
## 4  4722 727359.597 529051983960        2
## 5  2278   7063.703     49895903        5
## 6  1649  23415.356    548278901        5
```

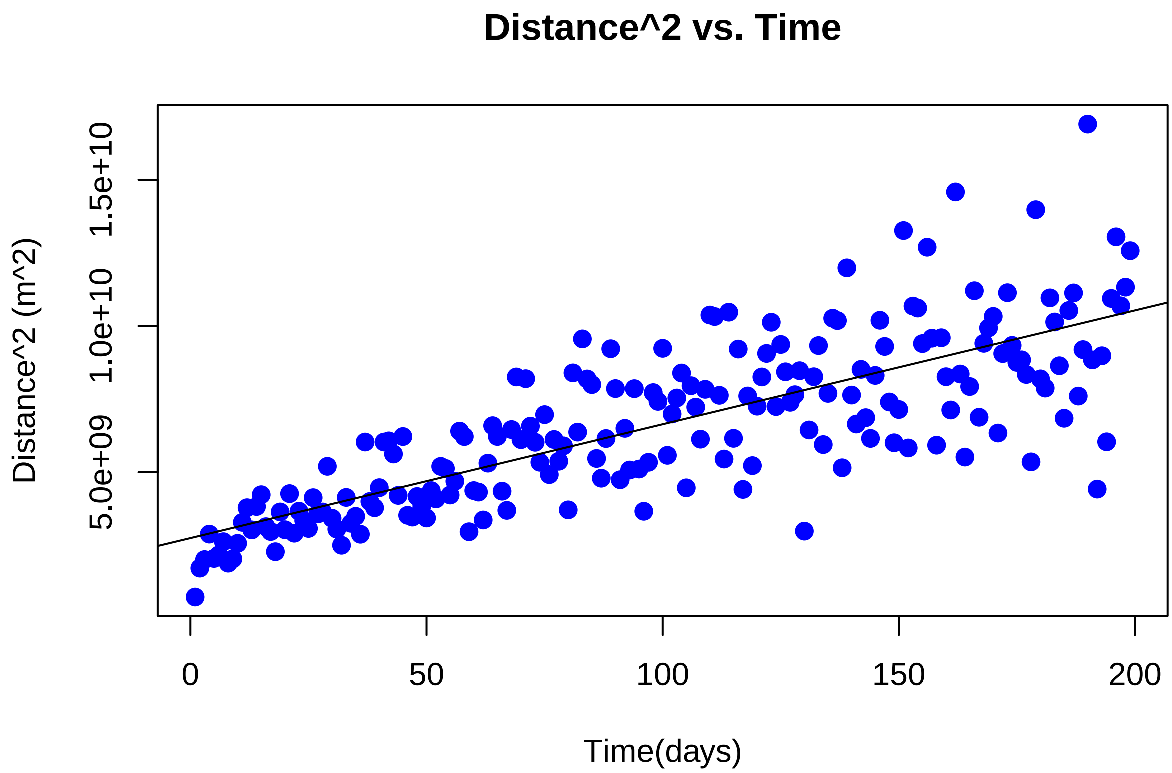
Now we graph the square distance vs. time and perform a linear least squares analysis on the data. We can limit the data we analyze based on distance and time limits. The column "freq_bid" is added to separate the data based on the number of data points for each bill.

```
#Graph Trajectories

time <- 200 #Time limit
distance <- 200000 #Distance limit
tdlim <- data.frame() #Set tdlim as dataframe
tdlim <- subset(cumdatagraph, d<distance & t<time & freq_bid > 0, select = c(t,dsq))
                    #select desired ranges of data
names(tdlim) <- c("t", "dsq") #Column names
attach(tdlim)
head(tdlim)
```

```
## # A tibble: 6 x 2
##       t         dsq
##   <dbl>       <dbl>
## 1    49  1033694939
## 2    89   890995562
## 3    22 28923840085
## 4    26    13015780
## 5     3 20933404758
## 6    15   224281144
```

```
#######plot the data
limdata <- aggregate(x=tdlim$dsq, by=list(tdlim$t), FUN=mean)
names(limdata) <- c("t", "xsq")
attach(limdata)
plot(limdata$t, limdata$xsq, pch=16, cex=1.3, col="blue", main="Distance^2 vs. Time",
     xlab="Time(days)", ylab="Distance^2 (m^2)")
abline(lm(xsq ~t))
```




```r
fit <- lm(xsq ~ t)
summary(fit)
```

```
##
## Call:
## lm(formula = xsq ~ t)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -5.799e+09 -1.103e+09 -1.849e+08  9.938e+08  6.760e+09
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.747e+09  2.563e+08   10.72   <2e-16 ***
## t           3.893e+07  2.222e+06   17.52   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.801e+09 on 197 degrees of freedom
## Multiple R-squared:  0.609,  Adjusted R-squared:  0.607
## F-statistic: 306.9 on 1 and 197 DF,  p-value: < 2.2e-16
```

Here we limited data points such that the number of days is less than 200 and distance is less than 200 km. As the time limit increases we see the data skewing away from linearity. If our hyothesis is correct, we can restore linearity to the data by identifying which data points include flight travel. We may then replace these distances with a distance value that is proportional to the frequency of flights between the two locations of the data points. We can achieve this using flight data of the number of passengers traveling between each location. The flights with a large volume of passenger travel would correspond to a "shorter distance" while the flights with a small volume of passenger travel would correspond with a "longer distance."

This relationship can be modeled as

$$\frac{a}{s^x}$$

where a is a constant, s is the number of seats per time interval for the flight, and x is an exponent to be determined. A variation of this equation will be used to replace the distance for data points that likely include flight travel. Data from the United States Bureau of Transportation Statistics will be used to achieve this.

```r
setwd("C:/Users/Nate/Desktop/wheresgeorge")
flights <- read.csv("US_CARRIER_ONLY_2017_ALL.CSV", header = TRUE)
attach(flights)
flights <- subset(flights, PASSENGERS > 0)
head(flights)
```

```
##       PASSENGERS DISTANCE UNIQUE_CARRIER
## 38042          1      102            HBQ
## 38043          1       27             J5
## 38044          1       18             J5
## 38045          1       50             J5
## 38046          1      102             J5
## 38047          1       18             J5
##                          UNIQUE_CARRIER_NAME ORIGIN_AIRPORT_ID ORIGIN
## 38042                    Harris Air Services             12610    KAE
## 38043 Kalinin Aviation LLC d/b/a Alaska Seaplanes             10204    AGN
## 38044 Kalinin Aviation LLC d/b/a Alaska Seaplanes             11545    ELV
## 38045 Kalinin Aviation LLC d/b/a Alaska Seaplanes             12523    JNU
## 38046 Kalinin Aviation LLC d/b/a Alaska Seaplanes             12728    KLW
## 38047 Kalinin Aviation LLC d/b/a Alaska Seaplanes             14062    PEC
##       DEST_AIRPORT_ID DEST MONTH DISTANCE_GROUP  X
## 38042           12728  KLW     5              1 NA
## 38043           15231  TKE     5              1 NA
## 38044           14062  PEC     5              1 NA
## 38045           15231  TKE     5              1 NA
## 38046           12610  KAE     5              1 NA
## 38047           11545  ELV     5              1 NA
```

The data includes all domestic flights in the U.S. with the number of monthly passengers specified by the year. The number of monthly passengers for the flights will be used to generate new distance values proportional to the number of monthly passengers rather than the actual distance traveled.

. . . More coming soon!