Where's George Data

Nathan Williamson

September 18, 2017

Get Data first...

```
setwd("C:/Users/Nate/Desktop/wheresgeorge")
fulldata <- read.csv("WG_ChicagoTrajectories_5.1.17.csv", header=TRUE)
attach(fulldata)
fulldata$Kilometers <- ((fulldata$Kilometers)/1000)
#head(fulldata)
```

Sort data method 1: We use the time value of the data point for delta t

```
deltatd1 <- data.frame()
 n <- numeric(1)
 n <- 1
 lengthdata <- nrow(fulldata)-1
   for (i in 1:lengthdata)
     {
         if (fulldata[i+1,1] == fulldata[i,1])
              {
                 deltatd1[n,1] <- fulldata[i+1,3]
                 deltatd1[n,2] <- (fulldata[i+1,4])^2
                 deltatd1[n,3] <- fulldata[i+1,4]
                 n <- n+1
             }

   }
attach(deltatd1)
names(deltatd1) <- c("t","dsq","d")
head(deltatd1)
```

```
##     t        dsq         d
## 1  53    741.717  27.23448
## 2 360 843799.321 918.58550
## 3 698 931523.016 965.15440
## 4  15   3177.311  56.36764
## 5  73   1318.775  36.31495
## 6 114   1559.773  39.49396
```

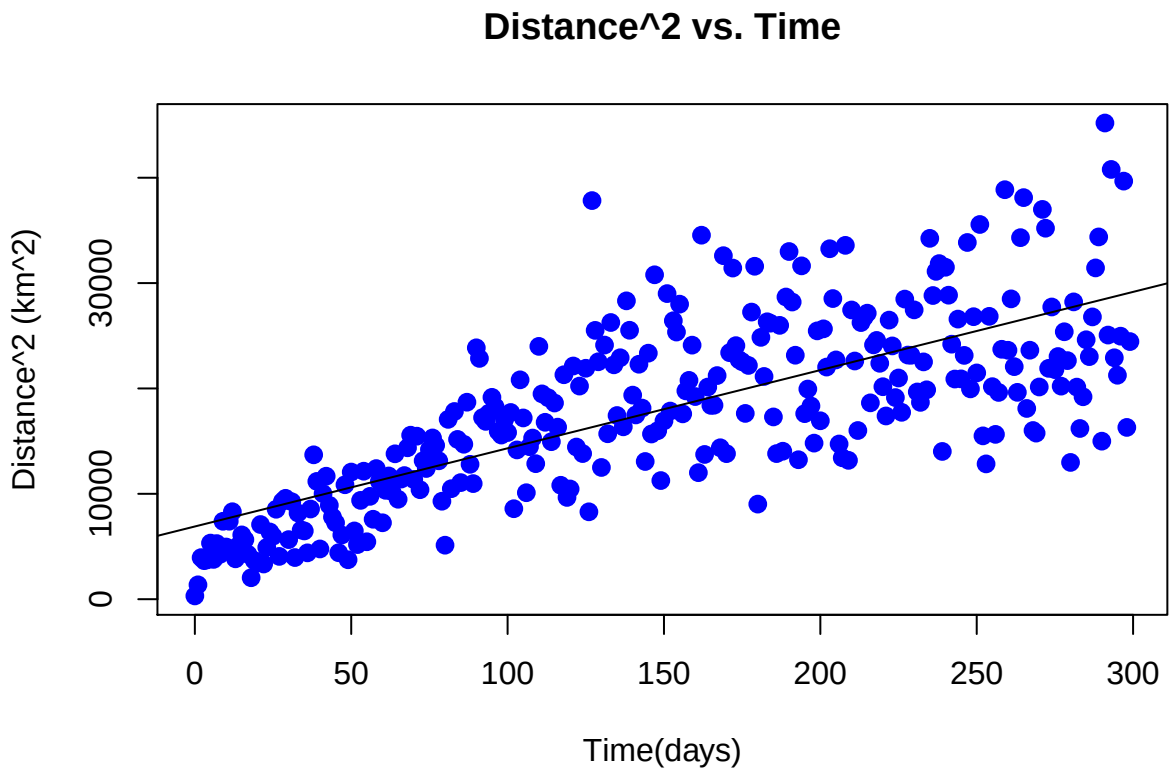
Now we can plot different subsets of the data...

```
#make a new dataframe by subsetting time and distance
time <- 300
distance <- 300
tdlim <- data.frame()
   tdlim <- subset(deltatd1, d<distance & t<time, select = c(t,dsq,d))
   names(tdlim) <- c("t", "dsq", "d")
   attach(tdlim)
   head(tdlim)
```

```
##     t        dsq         d
## 1  53   741.7170  27.23448
```

```
## 4  15  3177.3108  56.36764
## 5  73  1318.7752  36.31495
## 6 114  1559.7733  39.49396
## 7   9   243.3555  15.59986
## 8 100 15761.1076 125.54325
```

```r
#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 (km^2)")
abline(lm(xsq ~t))
```




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

```
##
## Call:
## lm(formula = xsq ~ t)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14699.2  -4119.6   -405.4   3599.8  21505.3
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 6901.010    666.246   10.36   <2e-16 ***
## t             74.213      3.856   19.25   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5784 on 298 degrees of freedom
## Multiple R-squared:  0.5541, Adjusted R-squared:  0.5526
## F-statistic: 370.4 on 1 and 298 DF,  p-value: < 2.2e-16
```

Sort data method 2: We use the difference between the time values for delta t

```
deltatd2 <- data.frame()
 n <- numeric(1)
 n <- 1
 lengthdata <- nrow(fulldata)-1
   for (i in 1:lengthdata)
     {
         if (fulldata[i+1,1] == fulldata[i,1])
              {
                 deltatd2[n,1] <- abs(fulldata[i+1,3] - fulldata[i,3])
                 deltatd2[n,2] <- (fulldata[i+1,4] - fulldata[i,4])^2
                 deltatd2[n,3] <- abs(fulldata[i+1,4] - fulldata[i,4])
                 n <- n+1
             }

   }
attach(deltatd2)
```

```
## The following objects are masked from deltatd1:
##
##     V1, V2, V3
```

```
names(deltatd2) <- c("t","dsq","d")
head(deltatd2)
```

```
##     t         dsq         d
## 1  53    741.7170  27.23448
## 2 360 843799.3208 918.58550
## 3 698 931523.0158 965.15440
## 4  15   3177.3108  56.36764
## 5  58    402.1106  20.05269
## 6 114   1559.7733  39.49396
```

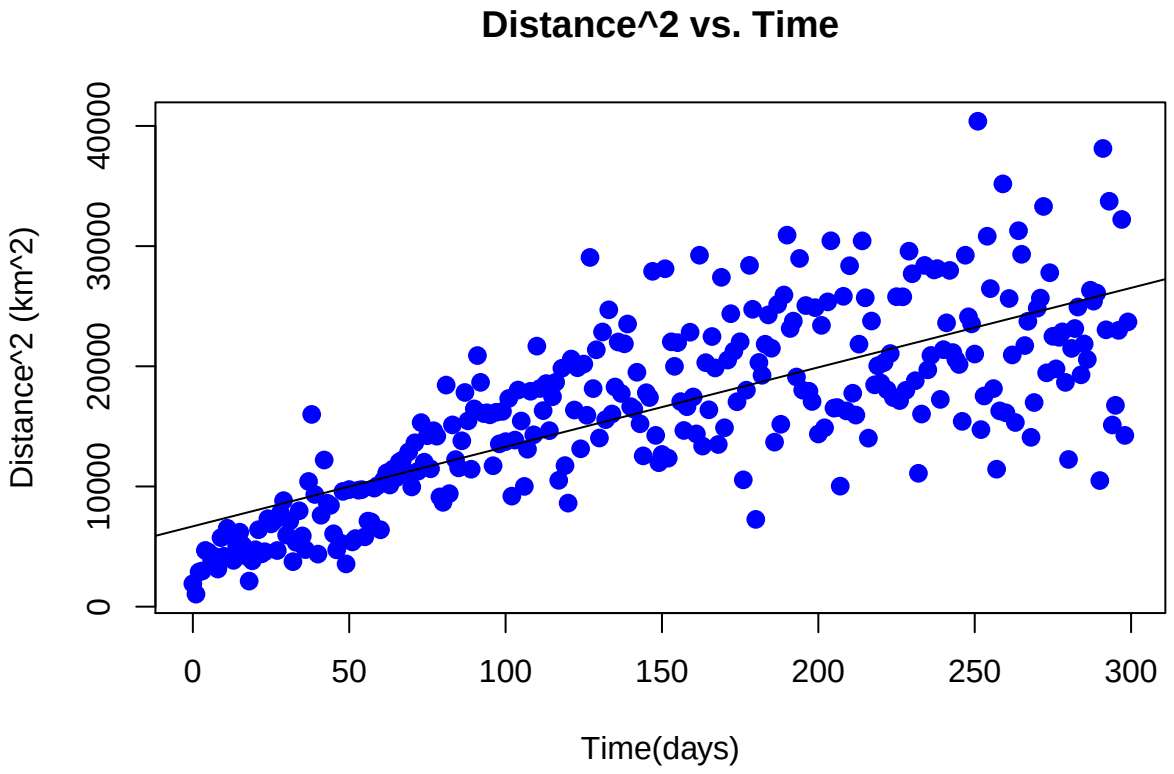
Now we can plot different subsets of the data...

```
#make a new dataframe by subsetting time and distance
time <- 300
distance <- 300
tdlim <- data.frame()
   tdlim <- subset(deltatd2, d<distance & t<time, select = c(t,dsq,d))
   names(tdlim) <- c("t", "dsq", "d")
   attach(tdlim)
   head(tdlim)
```

```
##     t        dsq         d
## 1  53   741.7170  27.23448
## 4  15  3177.3108  56.36764
## 5  58   402.1106  20.05269
```

```
## 6 114  1559.7733  39.49396
## 7   9   243.3555  15.59986
## 8 100 15761.1076 125.54325
```

```
   #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 (km^2)")
   abline(lm(xsq ~t))
```




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

```
##
## Call:
## lm(formula = xsq ~ t)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15370.0  -3342.0   -496.4   3146.8  17106.1
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6683.656    576.369   11.60   <2e-16 ***
## t             66.137      3.336   19.82   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5004 on 298 degrees of freedom
## Multiple R-squared:  0.5688, Adjusted R-squared:  0.5673
## F-statistic:   393 on 1 and 298 DF,  p-value: < 2.2e-16
```