Exploratory Analysis on Accelerometer and GPS Data in Kyrgyzstan

Ernest (Jiachang) Xu

9/1/2017

Section 1: Synopsis

The objective of this file is to perform exploratory analysis on accelerometer and GPS data in Kyrgyzstan. This step is crucial to our understanding to the Machine Learning Project. The ultimate goal of the Machine Learning Project is to train independent models to identify different types of road inpediments. This step, exploratory data analysis, give us a sense of how to approach the main problem and picks machine learning models.

Section 2: Data Processing

Before data cleaning, we need to load the raw dataset from **kyrgyzstan.csv**. We can count the number of rows in the raw dataset, and take a look at the first 10 rows.

```
if (!exists("KG.raw")) {
    KG.raw <- read.csv("./kyrgyzstan.csv")
}
print(nrow(KG.raw))
```

```
## [1] 4760
```

```
head(KG.raw, 10)
```

```
##    Batch.id                      time Longitude Latitude     x     y     z
## 1        12 2017-08-04 06:06:42+00:00  78.40248 42.48322 0.050 0.005 0.000
## 2        12 2017-08-04 06:06:42+00:00  78.40248 42.48322 0.058 0.002 0.001
## 3        12 2017-08-04 06:06:42+00:00  78.40248 42.48322 0.061 0.006 0.006
## 4        12 2017-08-04 06:06:42+00:00  78.40248 42.48322 0.057 0.012 0.006
## 5        12 2017-08-04 06:06:42+00:00  78.40248 42.48322 0.047 0.009 0.017
## 6        12 2017-08-04 06:06:43+00:00  78.40248 42.48322 0.052 0.012 0.007
## 7        12 2017-08-04 06:06:43+00:00  78.40238 42.48316 0.052 0.012 0.007
## 8        12 2017-08-04 06:06:43+00:00  78.40238 42.48316 0.065 0.010 0.004
## 9        12 2017-08-04 06:06:43+00:00  78.40238 42.48316 0.051 0.005 0.002
## 10       12 2017-08-04 06:06:43+00:00  78.40238 42.48316 0.060 0.006 0.009
```

Section 3: Data Cleaning

Let's remove any missing values in the raw dataset.

```
KG.valid <- KG.raw[!is.na(KG.raw$Batch.id) &
                       !is.na(KG.raw$time) &
                       !is.na(KG.raw$Longitude) &
                       !is.na(KG.raw$Latitude) &
                       !is.na(KG.raw$x) &
```

```
                      !is.na(KG.raw$y) &
                      !is.na(KG.raw$z), ]
KG.valid$Batch.id <- as.factor(KG.valid$Batch.id)
KG.valid$time <- substr(KG.valid$time, 1, 19)
KG.valid$time <- as.POSIXct(KG.valid$time, format="%Y-%m-%d %H:%M:%OS")
print(levels(KG.valid$Batch.id))
```

```
## [1] "12" "13"
```

```
head(KG.valid[KG.valid$Batch.id == "12", ], 10)
```

```
##    Batch.id                time Longitude Latitude     x     y     z
## 1        12 2017-08-04 06:06:42  78.40248 42.48322 0.050 0.005 0.000
## 2        12 2017-08-04 06:06:42  78.40248 42.48322 0.058 0.002 0.001
## 3        12 2017-08-04 06:06:42  78.40248 42.48322 0.061 0.006 0.006
## 4        12 2017-08-04 06:06:42  78.40248 42.48322 0.057 0.012 0.006
## 5        12 2017-08-04 06:06:42  78.40248 42.48322 0.047 0.009 0.017
## 6        12 2017-08-04 06:06:43  78.40248 42.48322 0.052 0.012 0.007
## 7        12 2017-08-04 06:06:43  78.40238 42.48316 0.052 0.012 0.007
## 8        12 2017-08-04 06:06:43  78.40238 42.48316 0.065 0.010 0.004
## 9        12 2017-08-04 06:06:43  78.40238 42.48316 0.051 0.005 0.002
## 10       12 2017-08-04 06:06:43  78.40238 42.48316 0.060 0.006 0.009
```

```
head(KG.valid[KG.valid$Batch.id == "13", ], 10)
```

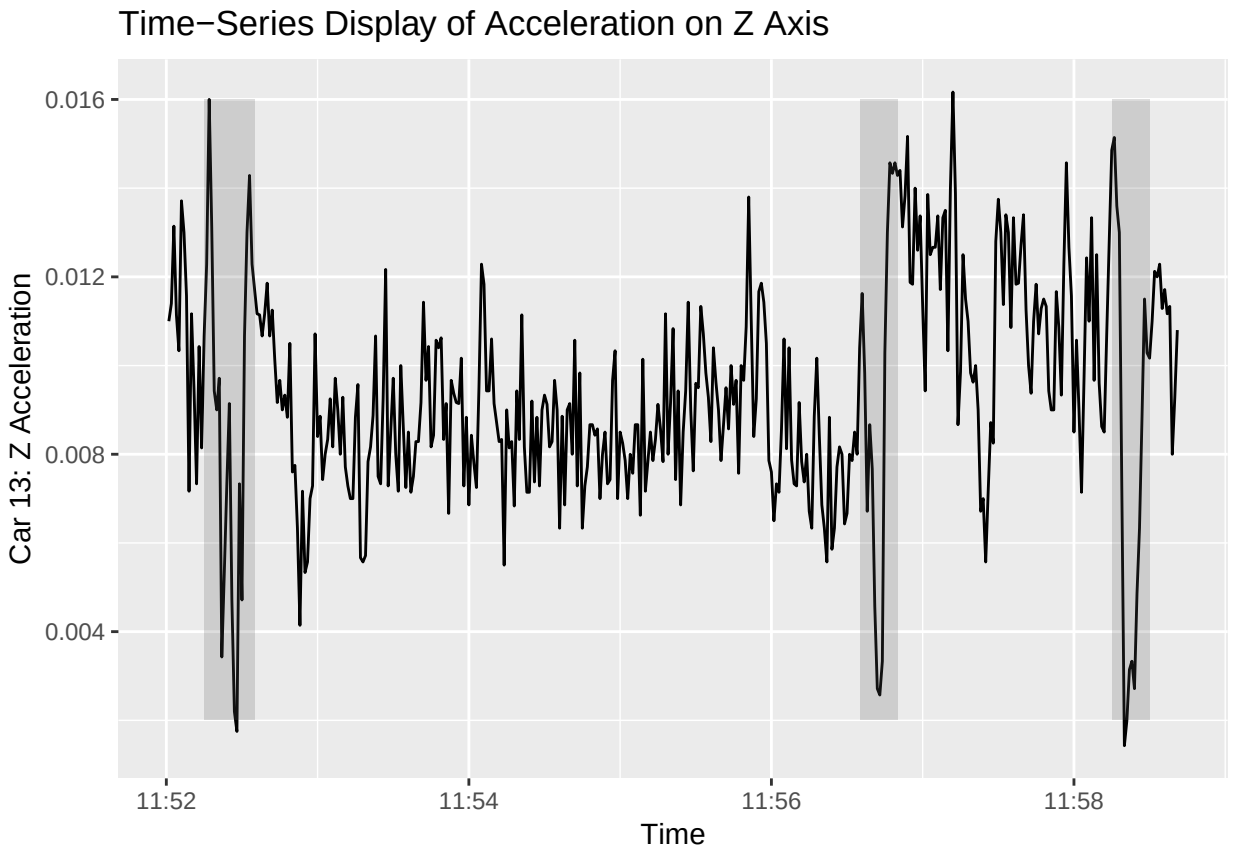
```
##      Batch.id                time Longitude Latitude     x     y     z
## 2134       13 2017-08-09 11:52:01  74.58395 42.80653 0.044 0.003 0.011
## 2135       13 2017-08-09 11:52:02  74.58395 42.80653 0.056 0.001 0.012
## 2136       13 2017-08-09 11:52:02  74.58395 42.80653 0.058 0.003 0.012
## 2137       13 2017-08-09 11:52:02  74.58395 42.80653 0.050 0.001 0.012
## 2138       13 2017-08-09 11:52:02  74.58395 42.80653 0.048 0.003 0.009
## 2139       13 2017-08-09 11:52:02  74.58395 42.80653 0.043 0.002 0.012
## 2140       13 2017-08-09 11:52:03  74.58395 42.80639 0.043 0.002 0.012
## 2141       13 2017-08-09 11:52:03  74.58395 42.80639 0.069 0.004 0.012
## 2142       13 2017-08-09 11:52:03  74.58395 42.80639 0.081 0.010 0.015
## 2143       13 2017-08-09 11:52:03  74.58395 42.80639 0.067 0.006 0.011
```

Section 4: Exploratory Data Analysis

```
# divide dataset into two groups based on Batch.id (levels = {12, 13})
KG.batch12 <- KG.valid[KG.valid$Batch.id==12, ]
KG.batch13 <- KG.valid[KG.valid$Batch.id==13, ]
# calcualte average accelerations on xyz axis for each second
KG.x12 <- aggregate(x ~ time, data = KG.batch12, mean)
KG.y12 <- aggregate(y ~ time, data = KG.batch12, mean)
KG.z12 <- aggregate(z ~ time, data = KG.batch12, mean)
KG.x13 <- aggregate(x ~ time, data = KG.batch13, mean)
KG.y13 <- aggregate(y ~ time, data = KG.batch13, mean)
KG.z13 <- aggregate(z ~ time, data = KG.batch13, mean)
# visualize time-series distribution of xyz-axis accelerations
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
plot.x12 <- ggplot(KG.x12, aes(time, x)) + geom_line() + xlab("Time") + ylab("Car 12: X Acceleration") +
plot.y12 <- ggplot(KG.y12, aes(time, y)) + geom_line() + xlab("Time") + ylab("Car 12: Y Acceleration") +
plot.z12 <- ggplot(KG.z12, aes(time, z)) + geom_line() + xlab("Time") + ylab("Car 12: Z Acceleration") +
plot.x13 <- ggplot(KG.x13, aes(time, x)) + geom_line() + xlab("Time") + ylab("Car 13: X Acceleration") +
plot.y13 <- ggplot(KG.y13, aes(time, y)) + geom_line() + xlab("Time") + ylab("Car 13: Y Acceleration") +
plot.z13 <- ggplot(KG.z13, aes(time, z)) + geom_line() + xlab("Time") + ylab("Car 13: Z Acceleration") +
plot.z13 + annotate("rect", xmin = as.POSIXct("2017-08-09 11:52:15", format="%Y-%m-%d %H:%M:%OS"),
                            xmax = as.POSIXct("2017-08-09 11:52:35", format="%Y-%m-%d %H:%M:%OS"),
                            ymin = 0.002, ymax = 0.016, alpha = .2) +
        annotate("rect", xmin = as.POSIXct("2017-08-09 11:56:35", format="%Y-%m-%d %H:%M:%OS"),
                            xmax = as.POSIXct("2017-08-09 11:56:50", format="%Y-%m-%d %H:%M:%OS"),
                            ymin = 0.002, ymax = 0.016, alpha = .2) +
        annotate("rect", xmin = as.POSIXct("2017-08-09 11:58:15", format="%Y-%m-%d %H:%M:%OS"),
                            xmax = as.POSIXct("2017-08-09 11:58:30", format="%Y-%m-%d %H:%M:%OS"),
                            ymin = 0.002, ymax = 0.016, alpha = .2)
```



We can see from the above graph **Time-Series Display of Acceleration on Z Axis** that there are three shaded fractions which strikes out. The overall similar pattern in these three fractions are a sudden decrease of z-axis accelration followed by a sudden increase of z-axis acceleration back to normal reading. These fractions maybe the scenarios when the vehicle drives through a pothole.

Section 5: Recommendations

Based on the findings from the above Section 4, We recommend to calculate the **time-series jolt** (first derivative of acceleration) for each Batch.id on three axis. When the vehicle is in constant linear motion, the

time-seris jolt should be slightly fluctuating around a normal reading. If there appears a sudden anormaly reading, then it is probable a pothole/speedbump (z-axis anormaly), a curvature (y-axis anormaly), or an inclination (x-axis and z-axis anormly).