# Capstone Project 1 – Data Wrangling

## Data files

There are 62 files, one for each day from Nov 1,2013 – Jan 1, 2014. Each file is of size ~300MB and has 4 million rows. These are tab delimited files with columns **gridID, timeInterval, countryCode, smsIn, smsOut, callIn, callOut, internetActivity** in the same order. They represent activities proportional to the amount of SMS/Calls/Internet-activity inside a given Grid id and during a given Time interval [10 min]. SMS & Calls are sent to or received from a nation with phone code **countryCode**.
Data files are downloaded from the link,
https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/EGZHFV

## Checking for Null values

Each file is read iteratively, and checked for null values column wise. We notice large numbers of null values in columns **smsIn, smsOut, callIn, callOut, internetActivity**. This can be ignored, as a given start interval time occurs in multiple rows consisting of different combinations of telecommunication activities recorded, with left over columns having null values. These are aggregated together while resampling the data.
It is also noted that none of the **gridID, timeInterval, countryCode** columns contain any null values.

```
Data/sms-call-internet-mi-2013-11-01.txt
gridID                 0
timeInterval           0
countryCode            0
smsIn            1981177
smsOut           3210070
callIn           3329423
callOut          2611016
internetActivity 2488626
dtype: int64
```

## Reading and merging data

Reading 20GB of data from 62 files iteratively and merging into a single data frame takes about 30 min and utilizes very high system memory. Alternatively, after reading each file, performing operations like sampling [daily, hourly], grouping & indexing reduces the number of rows drastically. Combining these individual data frames into a single data frame results in a faster and efficient loading operation.

## Handling datetime values

timeInterval column is represented in milliseconds, as Unix time. It is the number of milliseconds passed since 00:00:00 UTC Thursday, 1 January 1970. This column value is converted to pandas Datetime object and stored in a new column named **startTime**.

## Dropping unwanted columns

**timeInterval** column now has redundant values and **countryCode** column will not be used in this project. They are both dropped from the data frame.

## Resampling, grouping data & creating indexes

Several rows of data, with 10 min time interval are aggregated into two separate data frames, **dailyGridActivity** & **hourlyGridActivity**, with daily and hourly time intervals. They are grouped and indexed by **gridID** & **startTime** columns. Total volume of each activity over 2months for every grid is stored in data frame **totalGridActivity** and indexed by gridID.

## Visualizing the Grids

GridId can be mapped to the city of Milan using packages geopandas & geojsonio and loading the milano-grid.geojson file.