Dataset processing

**Processing code**

* **Taxiproc**: concerts raw csv to csv with formatted datetime
* **taxiproc\_linux**: same with min/max long/lat detected
* **writecounts**: converts fabio datasets to
  + coordinates.csv (for each grid size)
  + counts1110.csv
* **satscanproc**: converts counts1110.csv and coordinates.csv to satscan-compatible counts and coordinates files. Those files will still need to be converted to .cas and .geo files by SaTScan wizard

**Datasets grid circular trip**

1. NYU/Data/raw/taxi1110.csv: extracted from TaxiVis
   * cols: pickup\_time, pickup\_long, pickup\_lat
   * # lines: 491329 = total number of counts → *it should not be higher!*
   * So there were 491329 pickups in October 2011 in raw monitored data
2. NYU/Data/raw/taxi1110b.csv: formatting of datetime
   * taxi1110.csv with formatting cols
   * code: taxiproc.py
   * area: raw monitored area
     + min/max long: -74.0278 / -73.9437
     + min/max lat: 40.7003 / 40.7904
3. Aggregation by Fabio
   * Time: October 2011, 1 file per day, 31 files
   * No Smoothing
   * Area covered: 800Monitored Area
     + Grid has its own dimensions, so the file are sparse → All the cells that are not covered by taxi1110b.csv have a count = 0
     + Small differences between grids because grid step
   * Format of day files taxi2\_DAY\_1.scalars
     + 1st line: xs (number of cells along x), ys (number of cells along y)
     + 2nd line: lat/long of left bottom, lat/long of right top
     + <empty line>
     + xs \* ys lines, with a float value in each line representing the aggregated value at that cell
     + **ROW MAJOR ORDER**
   * Remarks
     + 800 have some 10^-27 values that need to be set to 0 → gaussian smoothing → *solved with Fabio’s correction*
     + SaTScan indicates that total counts = 24M for 200x200 → *problem solved*
4. Coordinates file: coordinates800.csv
   * code: writecounts.py
   * from Fabio taxi2\_DAY\_1.scalars → take one day and extract all coordinates
   * formatting ###### for ID locations
   * CHECK
5. Count files: counts1110\_800.csv
   * code: writecounts.py
   * aggregation of all FABIO\_DAY\_YY.scalars files with YY = 1..31
6. SaTScan Data Wizard
   * counts1110\_800 → Cases1110\_800.cas
   * coordinates800.csv → Coord800.geo
7. Create SaTScan parameter file from SaTScan interface

**Datasets ellipse trip**

1. taxi1110.csv
2. taxi1110b.csv
3. Aggregation Fabio
4. Count files: counts1110\_800.csv
5. Latlong coordinates file: coordinates800.csv
6. ***Conversion of latlongcoordinates to utm coordinates ← only change from circular***
   * python NYU/Code/latlong.py /path/to/latlongcoor.csv /path/to/output/utmcoord.csv
7. SaTScan Data Wizard
   * counts1110\_800 → Cases1110\_800.cas
   * coordinates800\_utm.csv → Coord800\_utm.geo
8. Create SaTScan parameter file from SaTScan interface

**Datasets** **road intersections circular processing**

1. raw harish data NYU/Data/harish/
   * coord\_graph.txt for the coordinates of the 3874 road intersections
   * oct-2011/scalar-2011-10-dd-hh.txt for the counts per node, day, hour
2. writing coordinates file for SaTScan
   * input: coord\_graph.txt
   * output: NYU/satscan/xp\_roads/coord\_graph.csv
   * code: writecountcoord.py
   * area: ***NOT A SQUARE !!***
     + min/max long: -74.01809 / -73.90906
     + min/max lat: 40.70178 / 40.87784
3. writing coord file for SaTScan
   1. input: coord\_graph.txt
   2. output: NYU/satscan/xp\_roads/coord\_graph.csv
   3. *code: writecountcoord.py*
4. counts file:
   * code: oct-2011/scalar-2011-10-dd-hh.txt all files
   * output:NYU/satscan/xp\_roads/
     + ctgraph1110\_d for daily aggregation
     + ctgraph1110\_h for hourly aggregation
5. SaTScan Data Wizard
   * ctgraph1110\_d → Casgraph1110\_d.cas
   * coord\_graph.csv → Coordgraph.geo
6. Create SaTScan parameter file from SaTScan interface

**Datasets road intersections ellipses processing**

1. raw harish data NYU/Data/harish/
2. writing coordinates file for SaTScan
3. writing counts file for SaTScan
4. coordinates file:
5. ***Conversion of latlongcoordinates to utm coordinates ← only change from circular***
   1. python NYU/Code/latlong\_utm\_graph.py /path/to/latlongcoord.csv /path/to/output/utmcoord.csv
6. SaTScan Data Wizard
   * ctgraph1110\_d → Casgraph1110\_d.cas
   * coord\_graph.csv → Coordgraph\_utm.geo
7. Create SaTScan parameter file from SaTScan interface

**Configurate aggregation code**

* Fabio code after total counts correction : NYU/Code/Fabio
* Install Qt 5.4, boost, glew (with synaptic) → OK
* Open Urbane.pro with Qt Creator
* File Sources/Pulse/GenerateScalarFiles.cpp → function taxi
* Put nice csv formatted file in rawdata/nyc/taxi2\_table.csv
* Questions : why create empty folders data, datastore, nyc… ?

**Aggregation by myself with Fabio code**

* Format csv taxi1110b.csv → taxi1110preagg.csv → renamed to taxi2\_table.csv
  + Format required by Fabio’s code
  + taxi2\_table.csv must be put in the new folder rawdata/nyc/, where rawdata has same level as src
* Create empty new folders: cf email: data same level as src, datastore, nyc/datastore
* Change Bounds and cellSizeInMeters
* Compile and run

**New data aggregated files**

* Location with count bug fix : Data/Fabio
* Grids : 20, 50, 100, 200, 300, 500, 1000
* Area: raw monitored area
  + min/max long: -74.0278 / -73.9437
  + min/max lat: 40.7003 / 40.7904

**Area Monitored – coherence check**

* NOT COHERENT
  + <https://www.google.com/maps/d/edit?mid=1dHivcnDx3W_noB3pC3RjIGxNU1s>
* Grid files have a larger monitored area than raw file…
* Sparse grid

**Lat/Long to UTM coordinates**

* One formula for conversion
* Need only to convert coordinatexxxx.csv file
* Need to choose one datum (reference)
  + <http://www.uwgb.edu/dutchs/usefuldata/utmformulas.htm>
  + NAD83/WGS84 indicated as ‘Global’ reference
  + What is used by Google earth ?
* Packages
  + Python : utm0.4.0 <https://pypi.python.org/pypi/utm>
    - installing with pip → **RESUME HERE**
  + pyproj
* Harish code → Implemented in ***latlong\_utm.py***

**Visualizing ellipses on ggearth**

1. convert coordinates1000.csv → coordinates\_utm.csv with latlong\_utm.py
2. run SaTScan with ellipse parameters
   * Compactness constraint : none / mid / strong
3. run SaTScan with ellipse parameters
4. write locations in kml file from SaTScan result file with vis\_ellipse.py
   * Extract cluster Ids in list
   * find ids locations with coordinates lat/long files
   * write list in kml SaTScan framework //circular kml files
5. Visualize locations
6. visualize polygons : failure

* write locations POLYGONS in kml file from SaTScan result file with vis\_ellipse.py
  + Create list of polygon bounds : take locations ids bounds +1/2 step longi/lati
  + order points by polar angle
  + writke in kml

**Checking column order & other possible causes of incoherent counts (river...)**

* Writecounts.py is OK
* Fabio ?
  + Column major, number of cells is #x, #y ?
* Checked everything, problem may come from taxi2\_DAYXX.scalars files...

**Counts in the river**

* On 50b\_05 → counts!=0 in the rivers :
  + locations
    - 045029:16 on 10 /04
    - 046030 : 8 on 10 /13
    - 144107:17 on 10/01, line n20557 in counts1110\_50
      * line 20559 in Fabio/50/taxi2\_DAY1.scalars
      * Problem from lat/longi step/ I do not think so...
  + clusters
    - cluster 6 → 121 cases ! On 1 day !!
    - Cluster 4 : 161 cases with almost all the cluster in the water
    - Expected cases:645 ??? **SATSCAN EXPECTED CASES ??**

**Using SaTScan with HOUR time precision**

i) Formulas

* Switch to generic time precision
  + Calculate absolute time with formula
  + Run SaTScan
  + Transform back the time results in yy-mm-dd-hh
* Formulas of absolute generic time g, from a yy-mm-dd-hh
  + g =
    - y : year, with origin y0
    - m : month, with origin m0
    - d : day
    - h : hour
* From g to yy-mm-dd-hh : cf notebook
* Origins : y0 = 2011, m0 = 10

ii) Processing input

* input : ctgraph1110\_h.csv
* output : cfgraph1110\_h\_gen.csv
  + Max unit : 743 (=24\*31 - 1)
* code : writecountcoord.py
* command : python writecount.py ….///

iii) Processing output

* input : resg1110\_h\_01
* output : resg1110\_h\_01\_ymdh
  + Same output format, just the datetime changes
* code : write\_res\_ymdh.py
* Warning : kml file also needs to be changed (classic writing kml procedure for ellipses)
* command : python write\_res\_ymdh.py inputpath outputpath

iv) Visualization

* input : resg1110\_h\_01\_ymdh
* output : resg1110\_h\_01\_ymdh.kml
* code:vis\_ellipse\_graph2.py
* command: python vis\_ellipse\_graph\_h2.py inputpath /home/ferdinand/Documents/NYU/satscan/xp\_roads/coord\_graph.csv graphe\_h\_elp\_2k\_none\_7

**Preprocessing for Telang experiment : see Telang\_Preprocessing.doc**

**Pickups or dropoffs ?**

* Pickup will detect : street blocages, accidents, road network
  + Will it detect an event with big affluence ?
  + Good for low counts analysis
* Dropoff will detect : street blocages, accidents, road network, beginning of an event with big affluence
  + Good for high counts analysis : many people go to the same place
  + Also good for low counts analysis
* Experiments with SaTScan made with pickups
  + Would not need a lot of time to recompute
  + Will need to recompute anyway with count replacement and iterative SaTScan
* Experiment topology Harish technique ?

⇒ **Conclusion**

1. **For now, use pickups temporarily**
2. **Wait for topology answer, but dropoffs seem better**