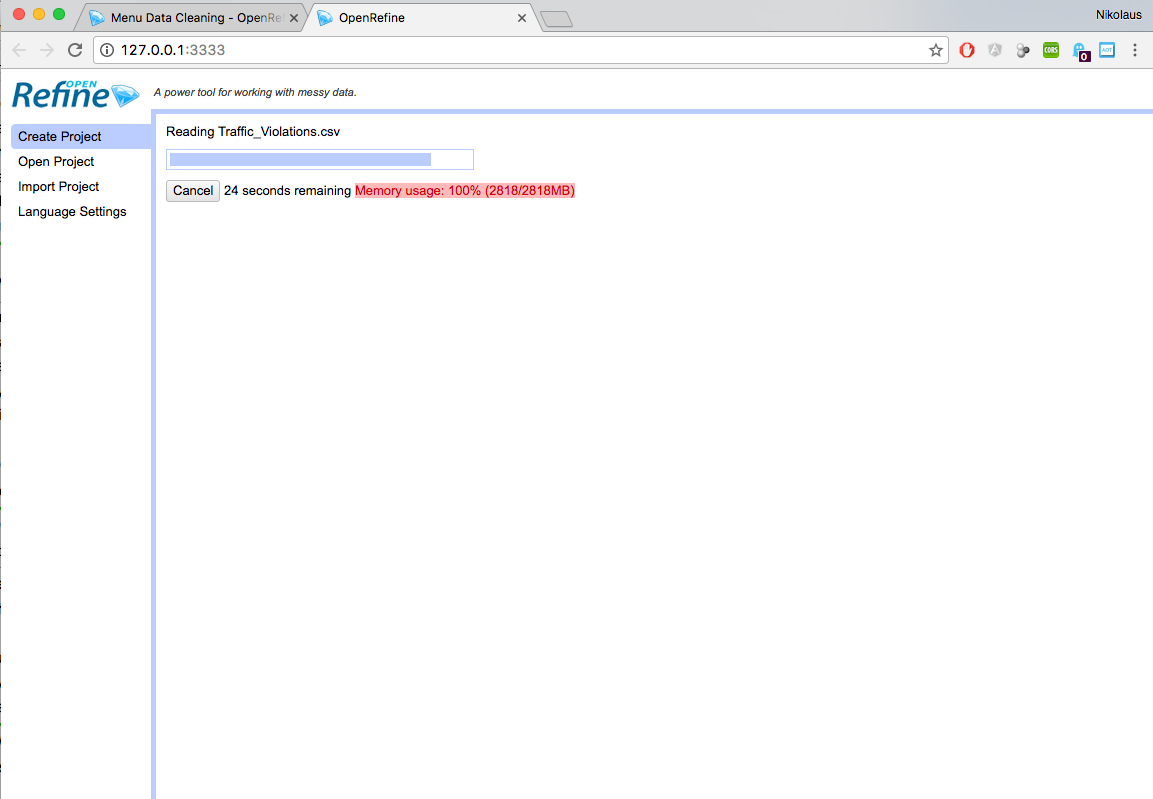
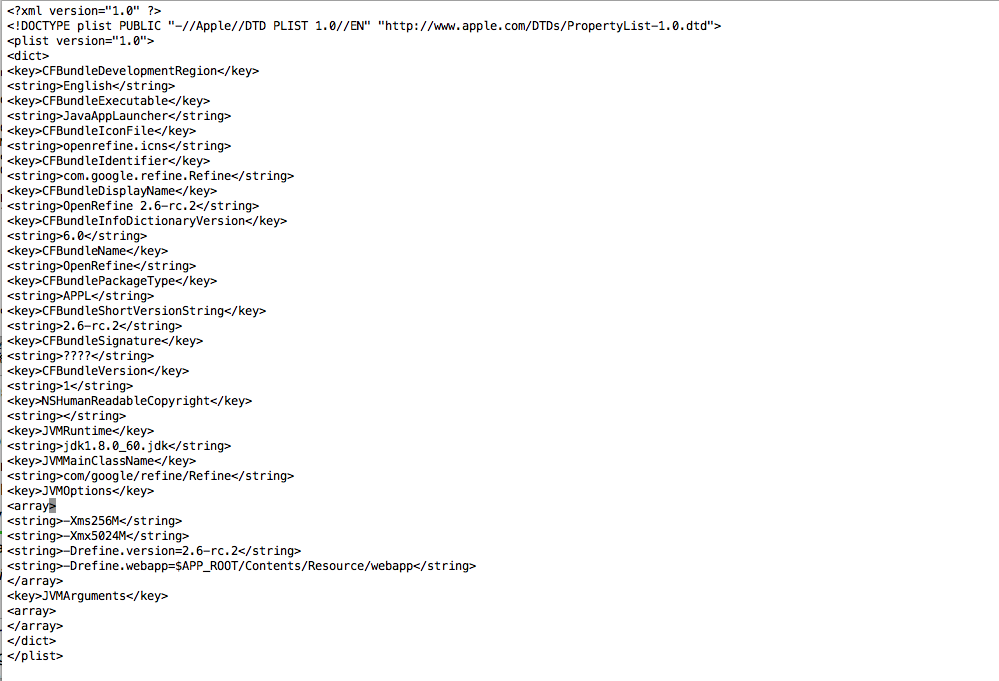
Just opening the csv file with openrefine we already faced Out of Memory problem,

Turn out there is limitation on how much memory openrefine can use. This one reason is why we consider using python / programming to clean up the data. However, to give more understanding about what we can clean in the dataset, openrefine may be a good start.



Therefore, we need to find a solution for this out of memory problem first and we found out that because Openrefine runs on java platform, we can configure memory usage by doing a little tweak in the openrefine configuration file (mac version)

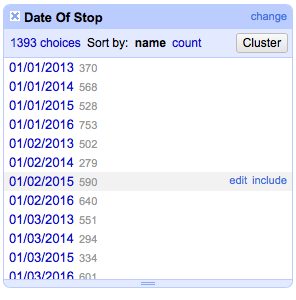
/Applications/OpenRefine.app/Contents/Info.plist



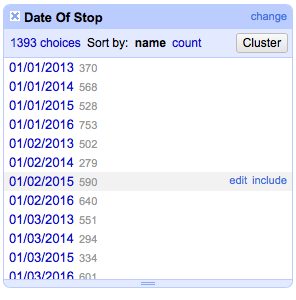
We change the memory usage into 5024M (5GB), be careful to do this one must have at least 10GB memory.

* Potential null value checking: Date of Stop, Time of Stop

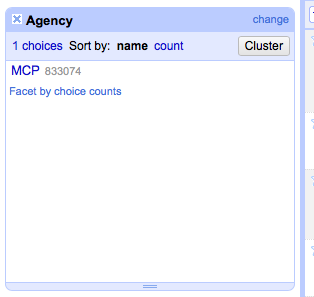
Date of Stop Field



Date of stop

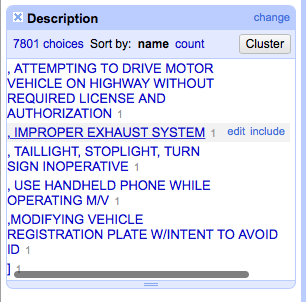


Agency

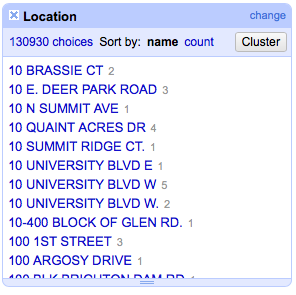


* Potential regular expression / string cleaning:

Description



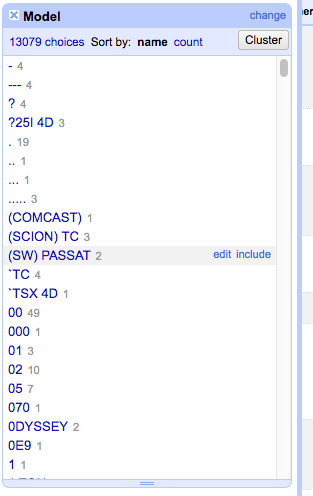
Location



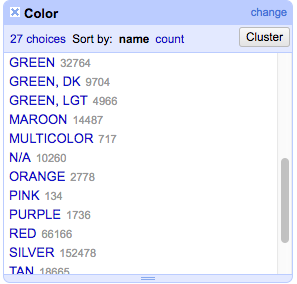
Make (unwanted values) : in addition we can do merging as well



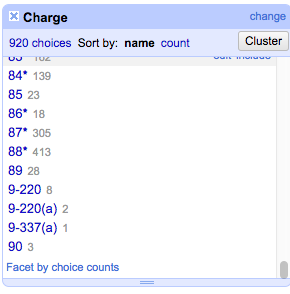
Model (Unwanted values)



Color (NA value and additional description with ,)



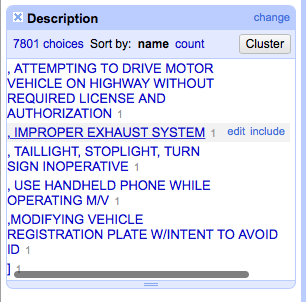
Charge (Unwanted value \*)



Driver City

* Potential clustering + merging operation:

Description

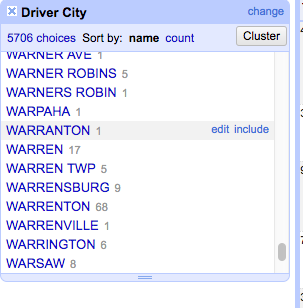


Location,

Make,

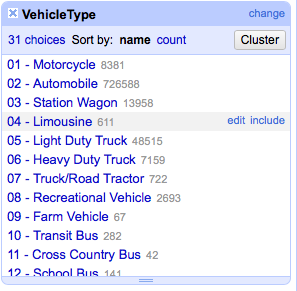
Model,

Driver City

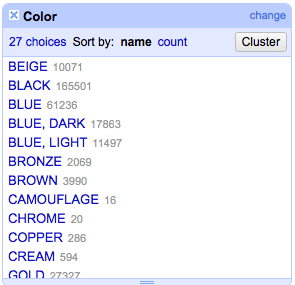


* Potential lookup value: State, SubAgency,

VehicleType,



Color,



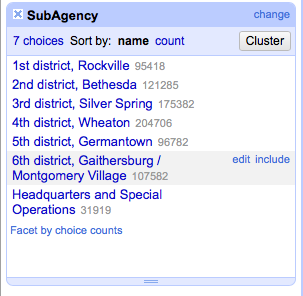
Charge (this should define the charge for the driver but somehow, the variance is so big, probably we can divide the charge as law number and section),

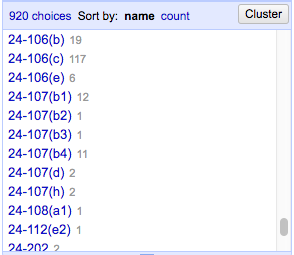
Article,

Race,

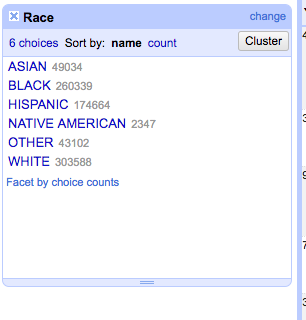
and Fields that have yes / no value (Accidents – Workzone, Contributed to Accident)

SubAgency:

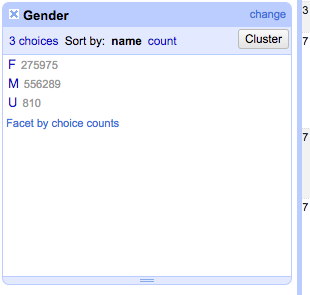




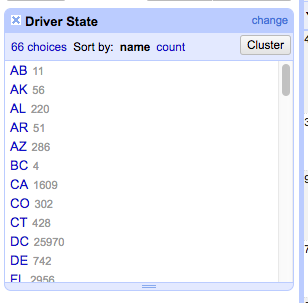
Race



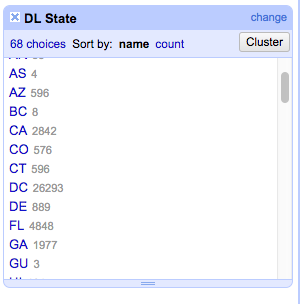
Gender (What is U mean -> unknown?) we accept these 3 values



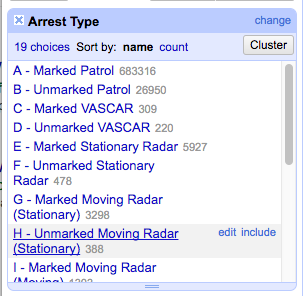
Driver State, beware of XX value -> what state is this?



DL State

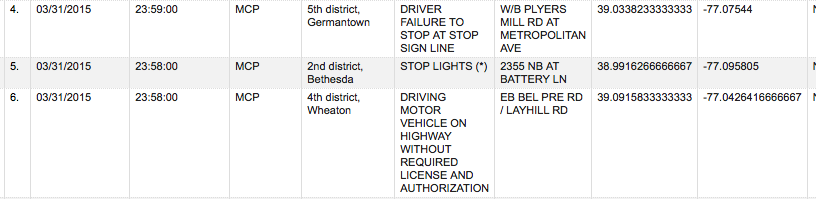


Arrest Type



Potential format checking: Date of stop, Time of Stop,

Latitude, Longitude => we can clean the value for at max 6 digits precision.



Potential Numeric Constraint Checking: Year (ridiculous number), we clean value for year more than this year (2016) and less than 1882



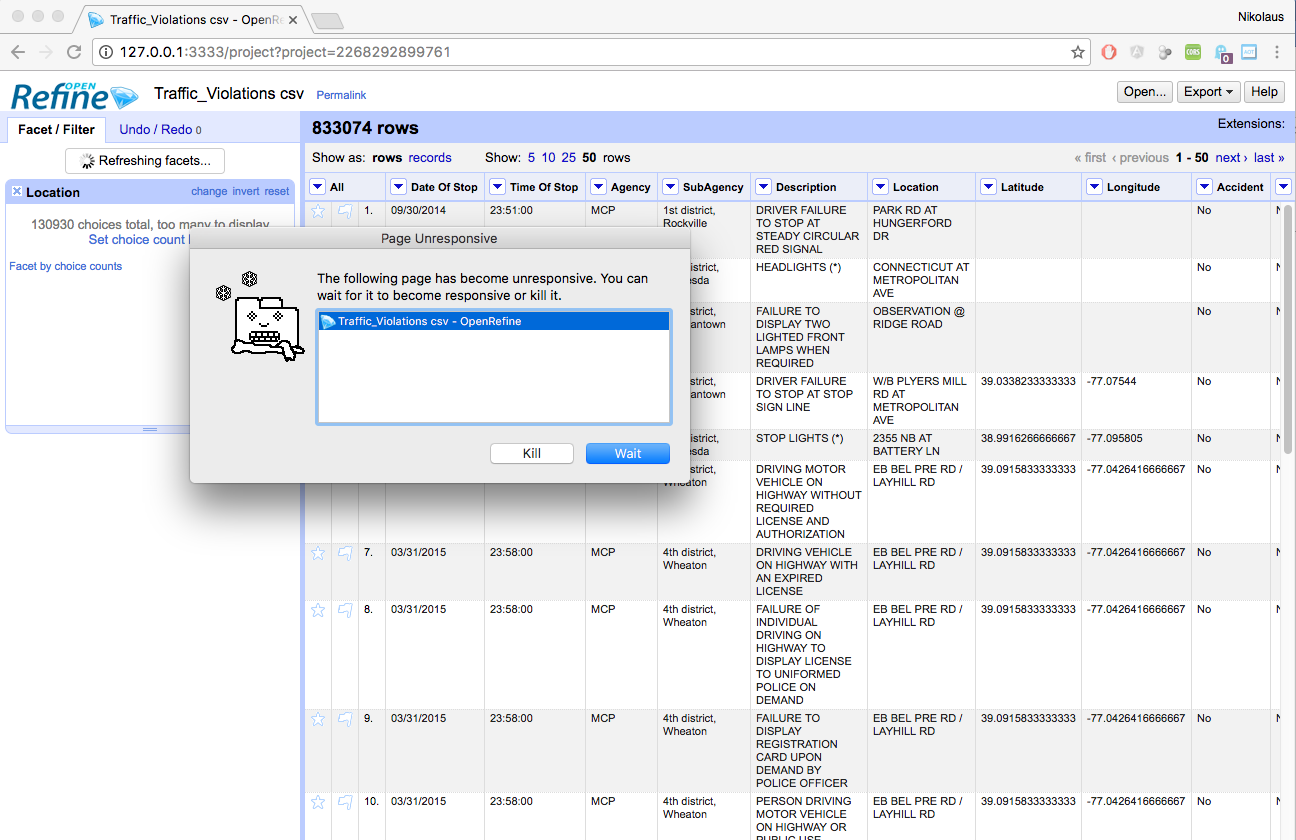


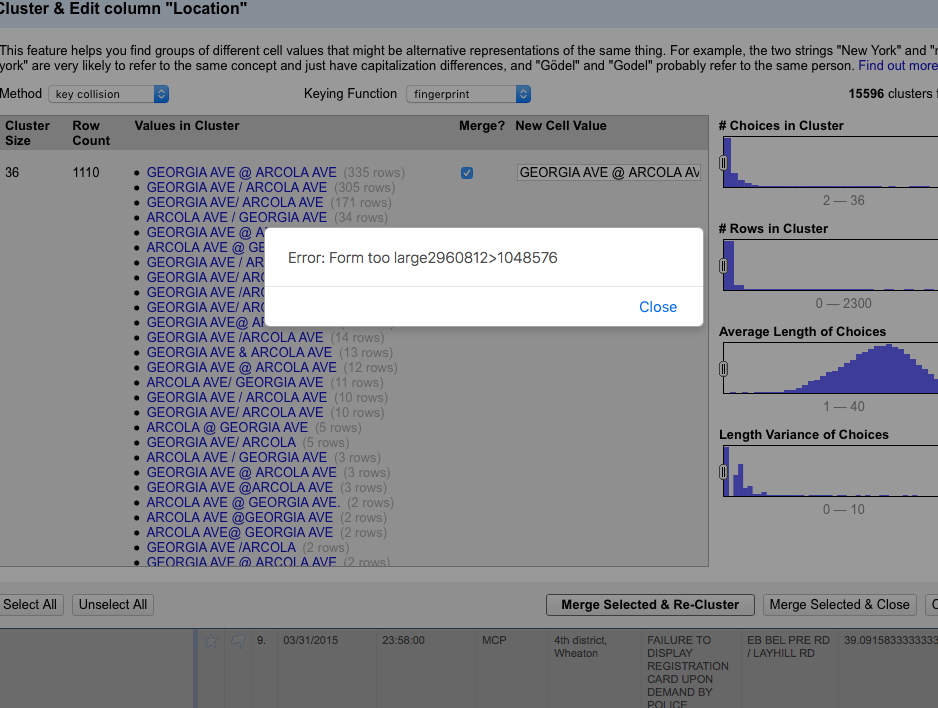
Article has blank values



Geolocation is duplicate value for latitude and longitude

OpenRefine hang when trying to make facet for Location





Need to find better way / technique

Function: clean / replace, custom function, ltrim, rtrim, uppercase, lowercase, lookup

Github project for source code and documentation:

<https://github.com/nikolausn/lab-data-cleaning-project>

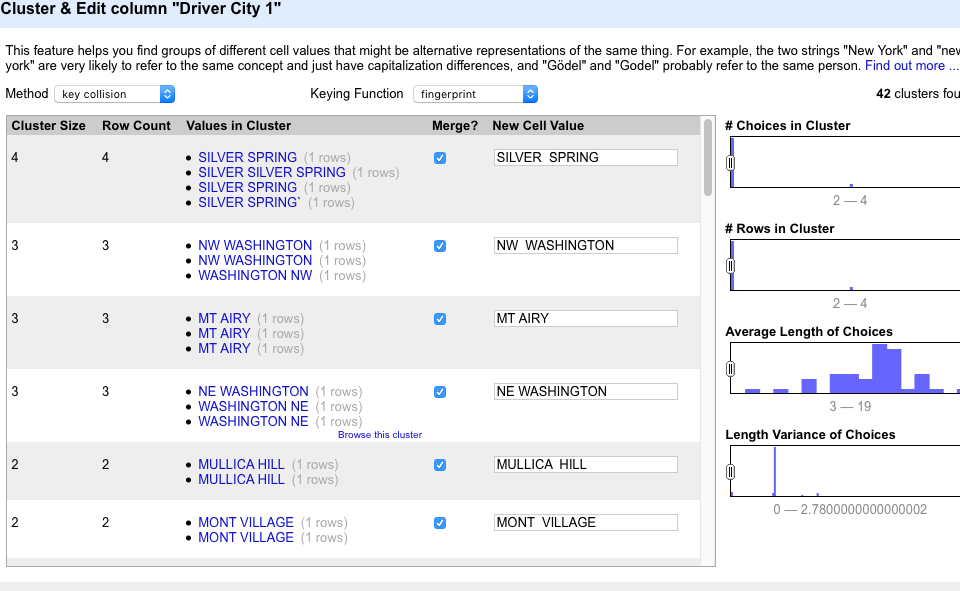
Action:

Field description

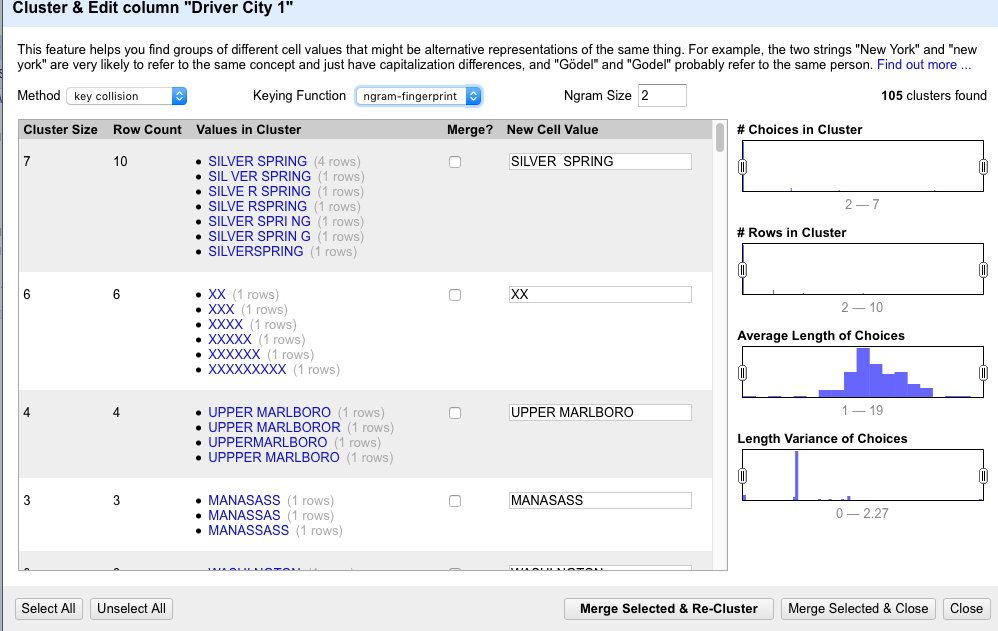
* Replace improper characters: , ] ( ) \* to space
* Trim the value
* Collapse white space
* Uppercase the value

Merge and replace

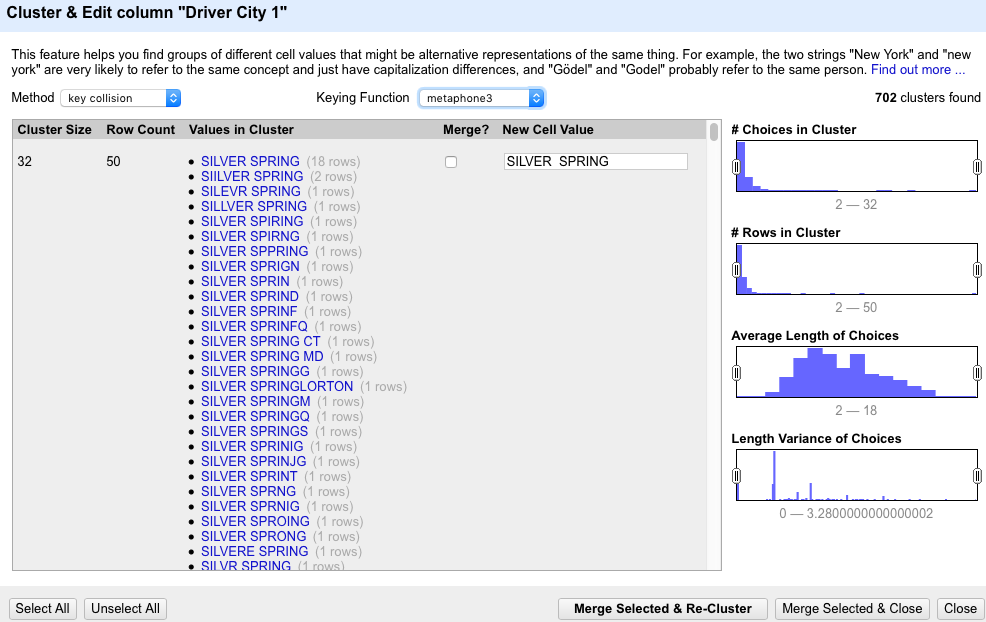
* driver city
* fingerprint

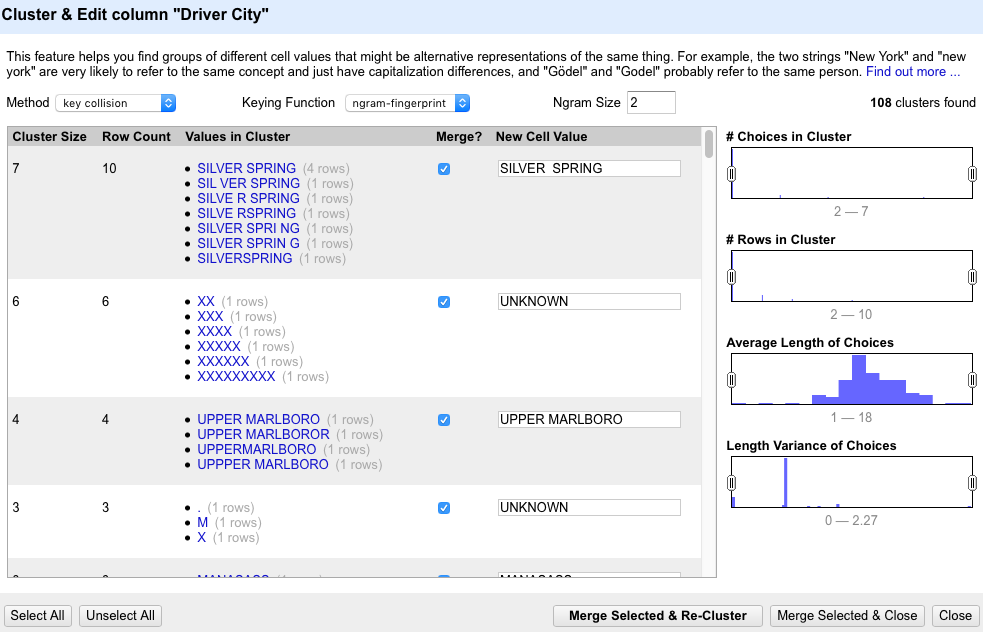


* ngram-fingerprint

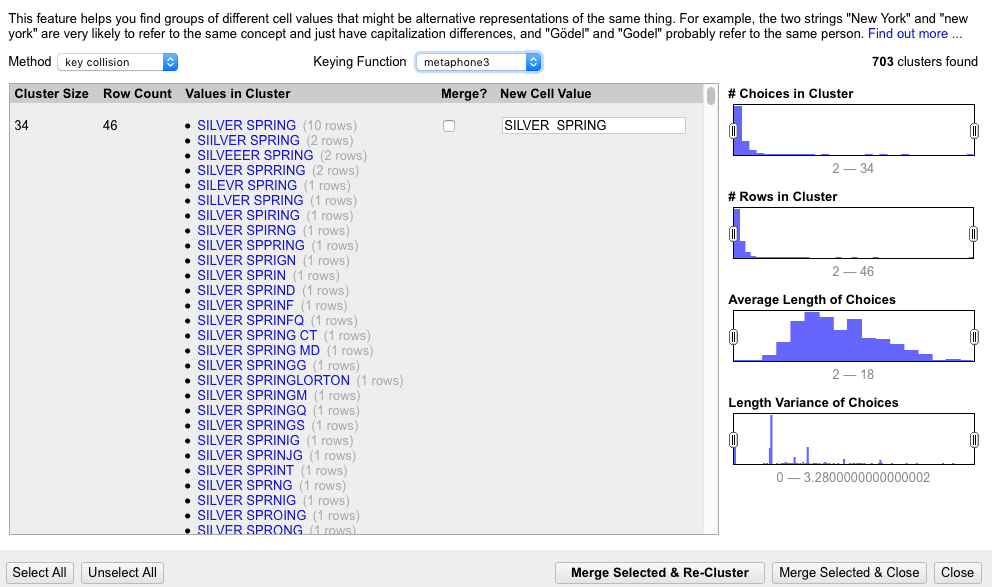


* METAPHONE

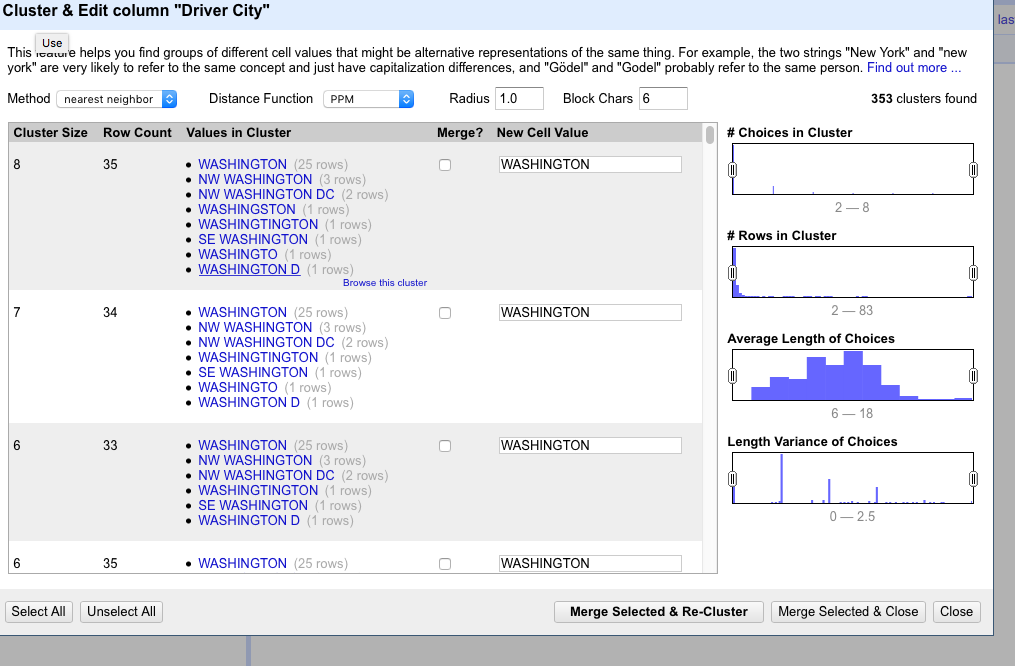




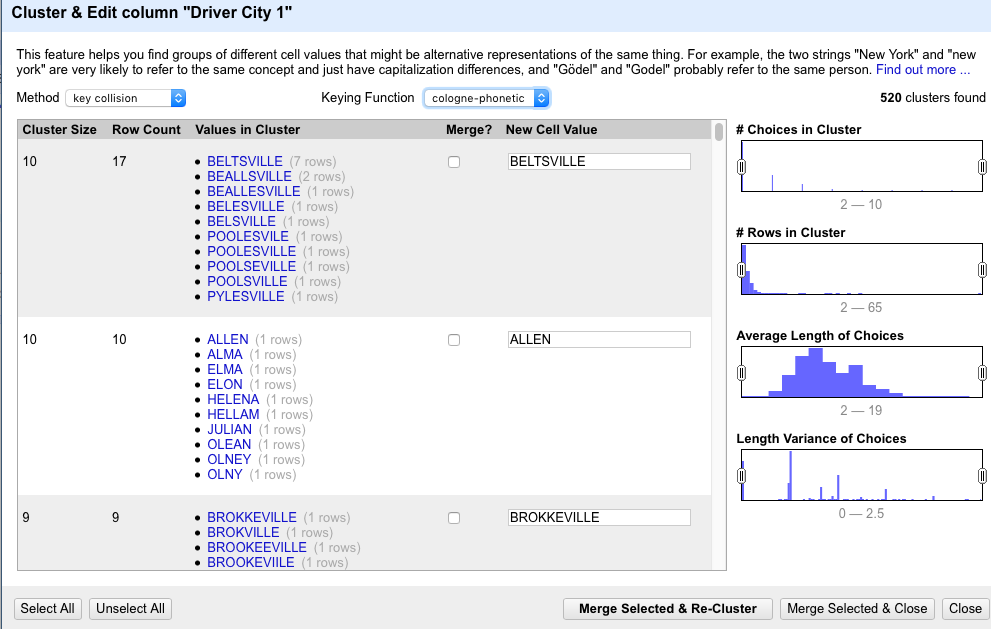
* metaphone3



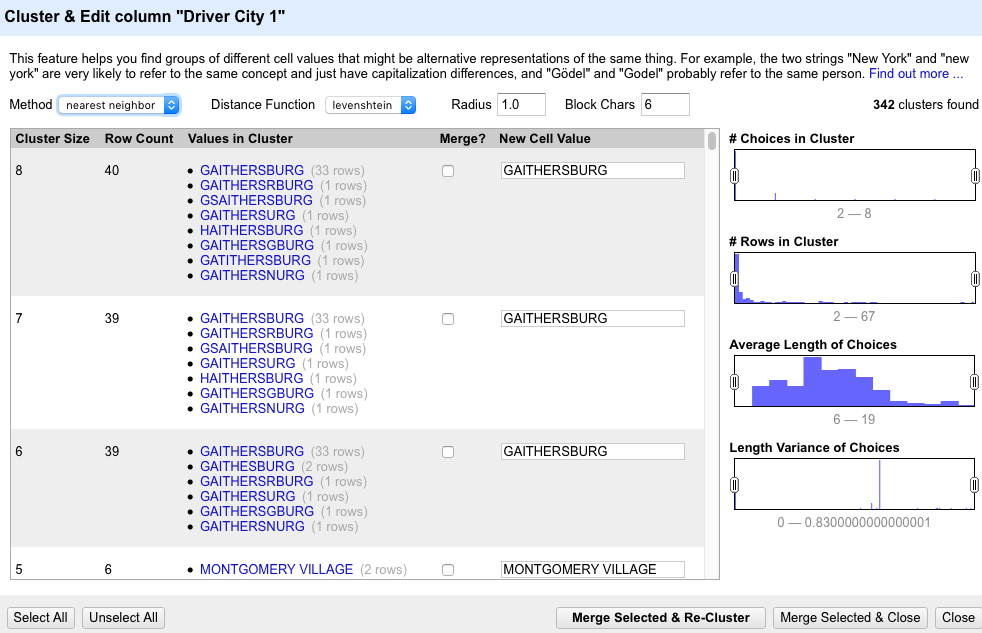
* Nearest Neighbor – PPM



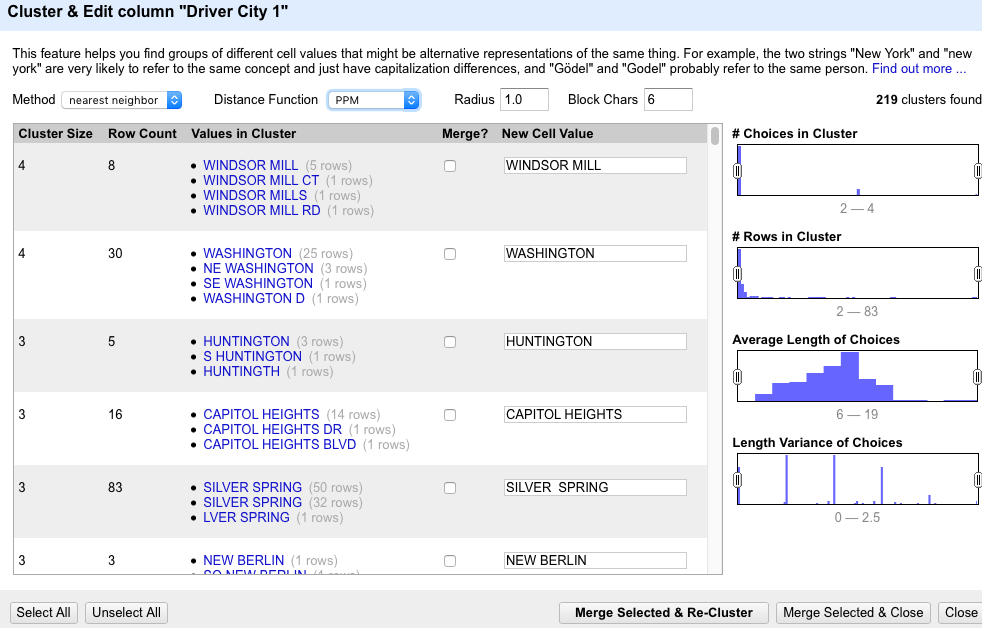
* COLOGNE PHOENETIC

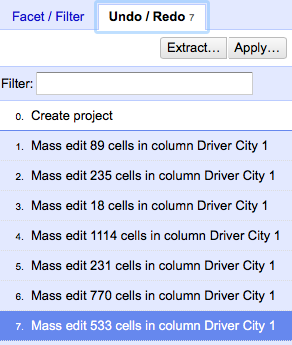


* NEAREST NEIGHBOR – LEIVENHSTEIN

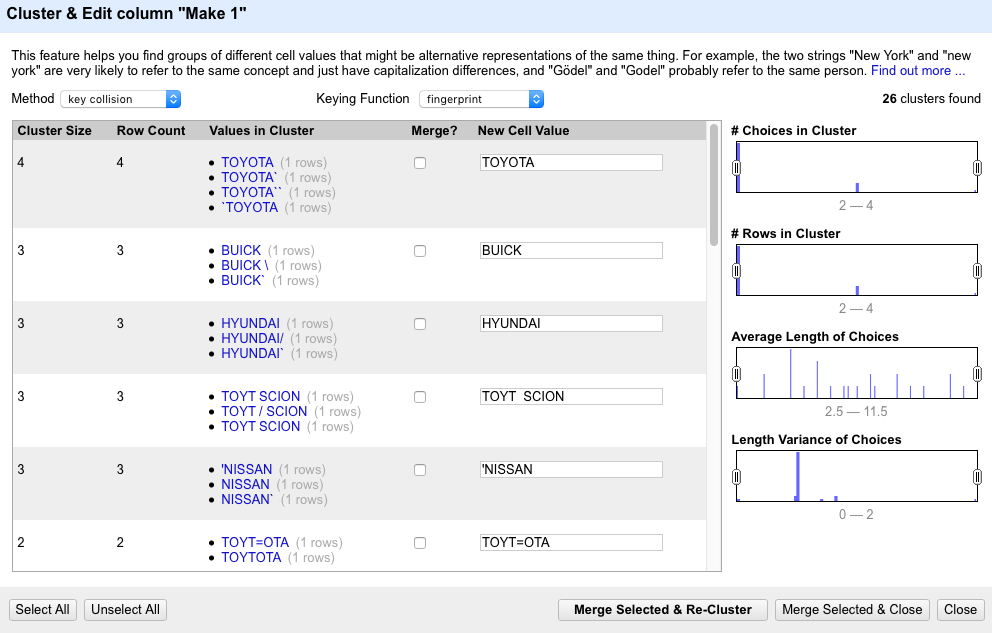


* NEAREST NEIGHBOR – PPM

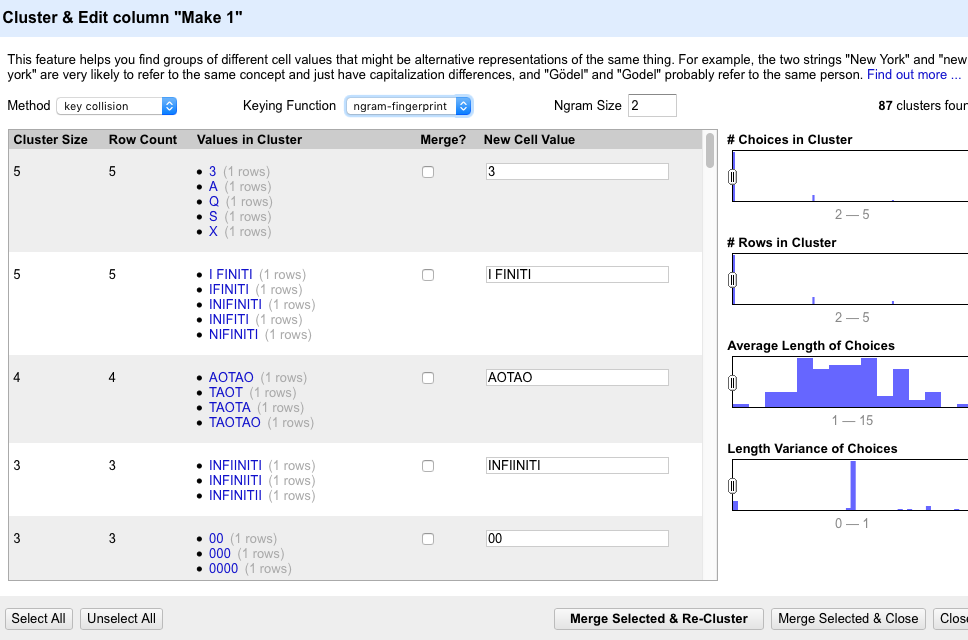




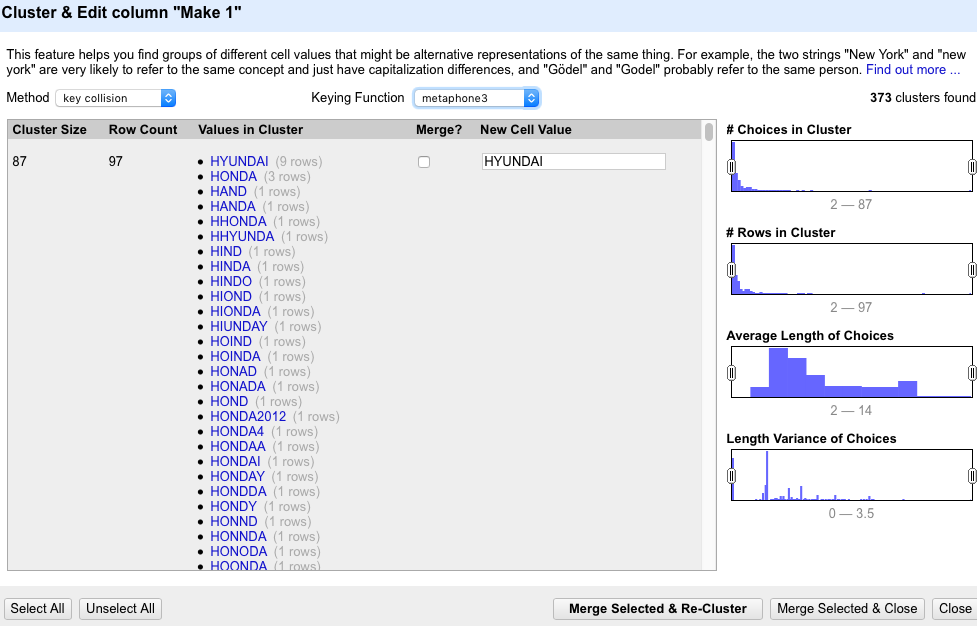
* Column Make
* KEY COLLISION – FINGERPRINT



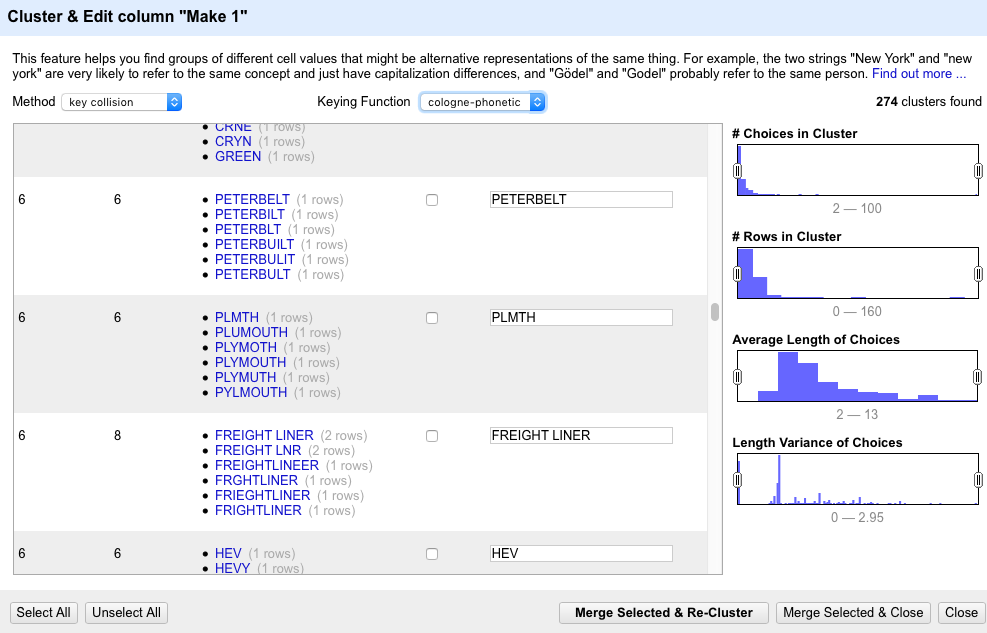
* KEY COLLISION – NGRAM FINGERPRINT



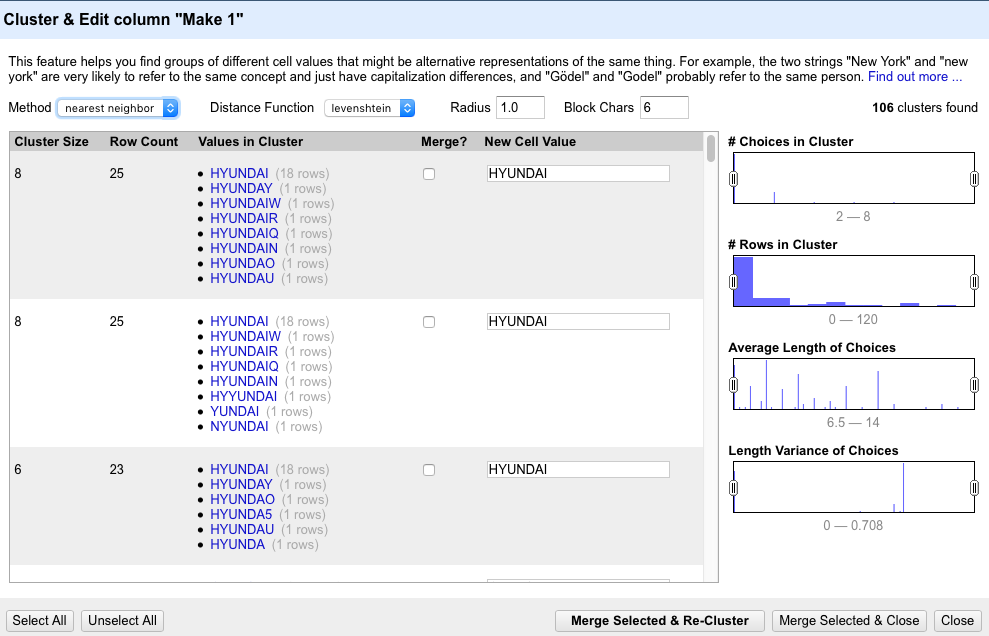
* KEY COLLISION - METAPHONE 3

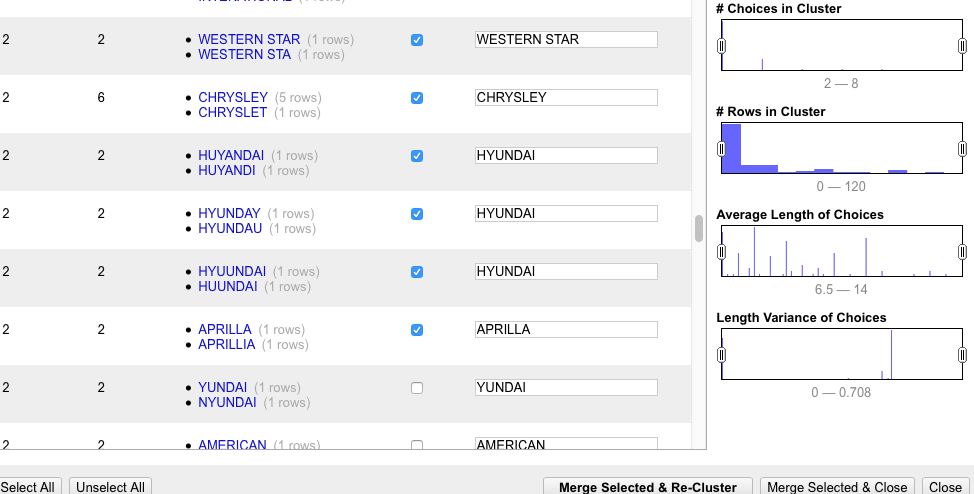


* KEY COLLISION – COLOGNE PHONETIC

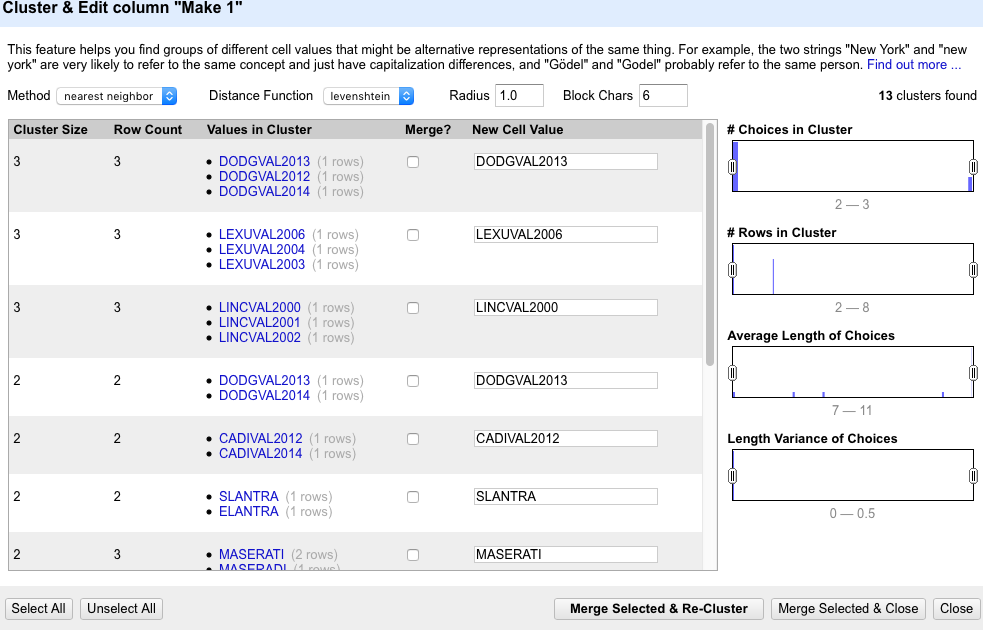


* NEAREST NEIGHBOR – LEVEINHSTEIN

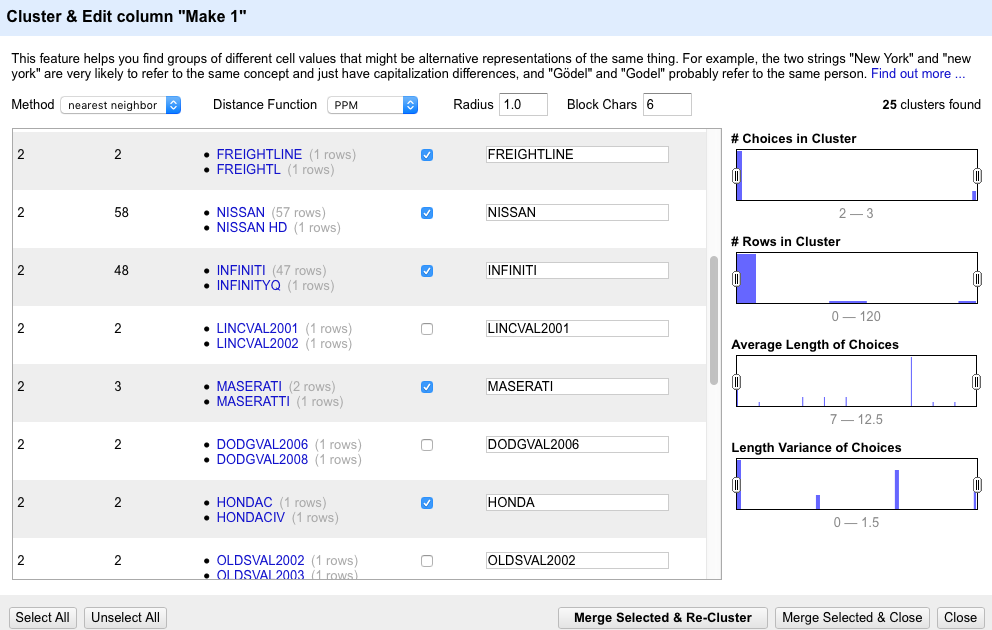




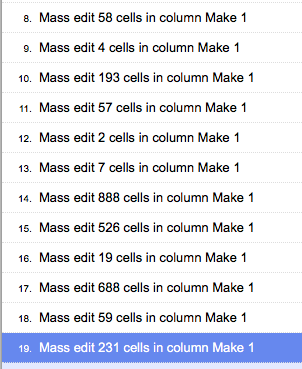
* NEAREST NEIGHBOR – LEVENHSTEIN



* NEAREST NEIGHBOR – PPM



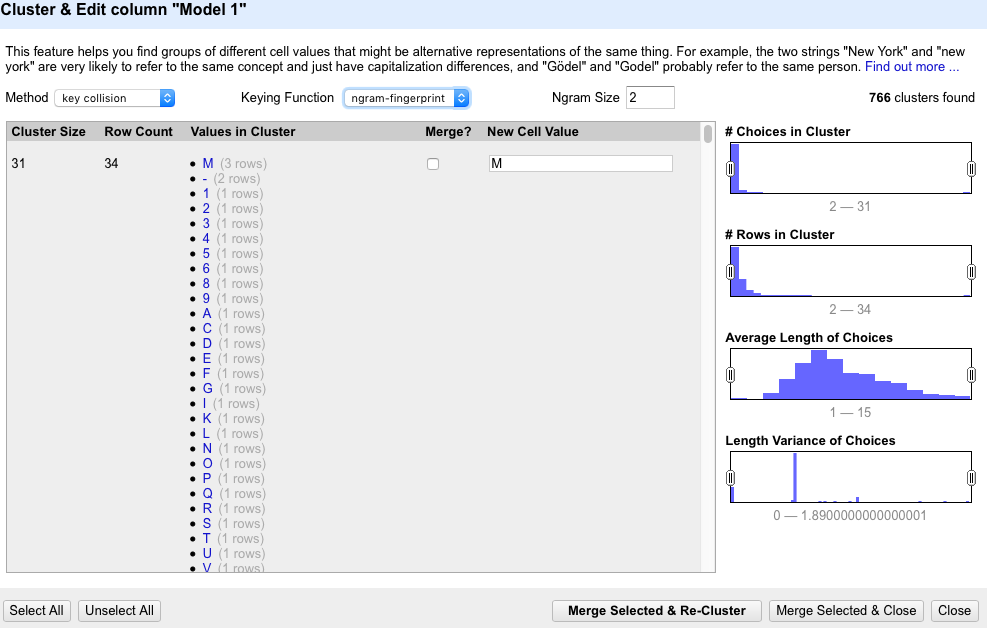
* REPLACEMENT SUMMARY



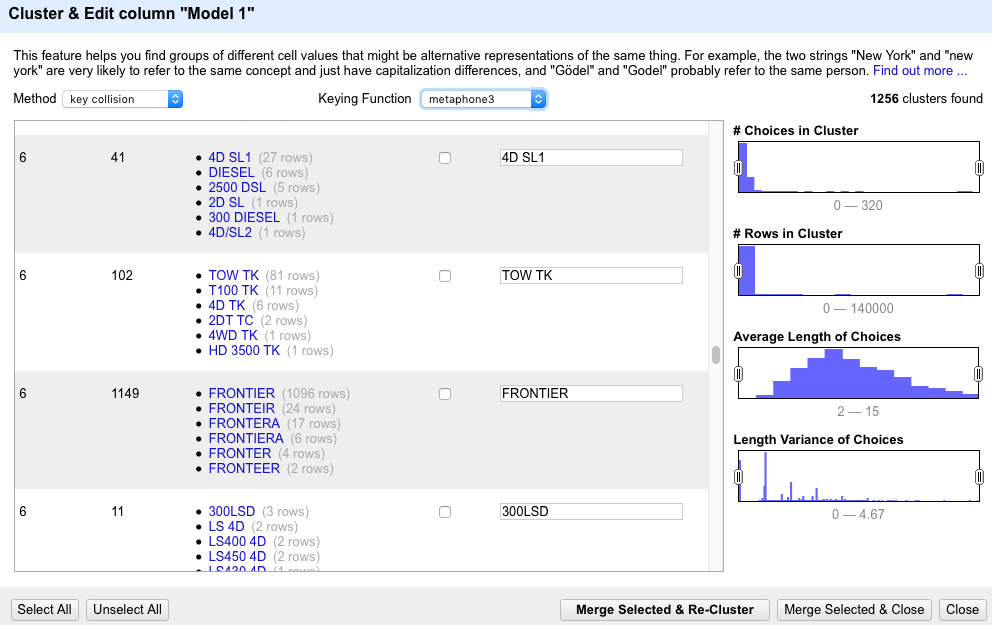
* MODEL 1
* KEY COLLISION – FINGERPRINT



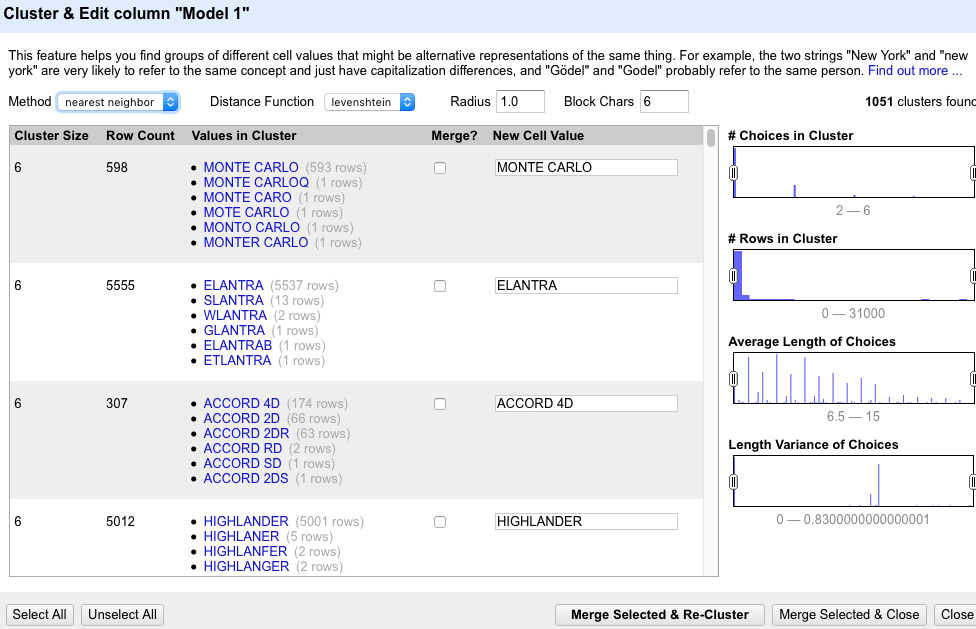
* KEY COLLISION – NGRAM FINGERPRINT



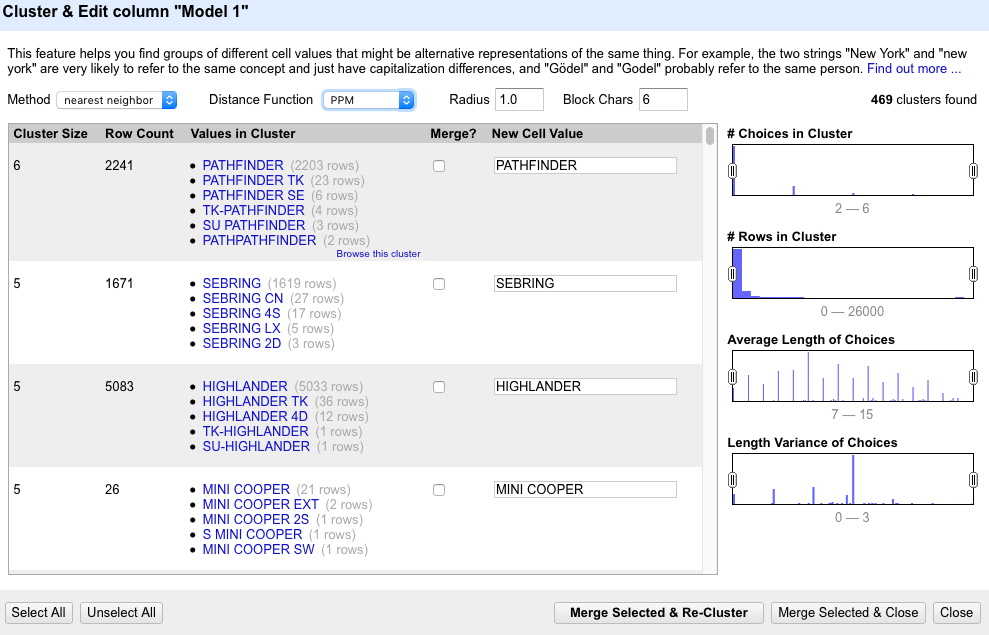
* KEY COLLISION – METAPHONE 3



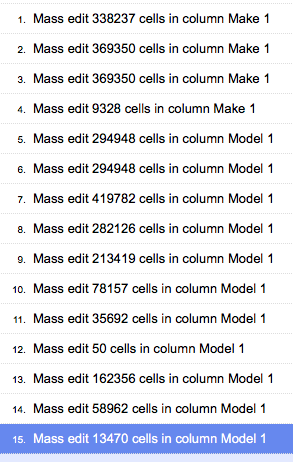
* KEY COLLISION – COLOGNE PHONETIC (NOTHING SPECIAL, MOSTLY THE SAME WITH METAPHONE 3)
* NEAREST NEIGHBOR – LEVENSHTEIN



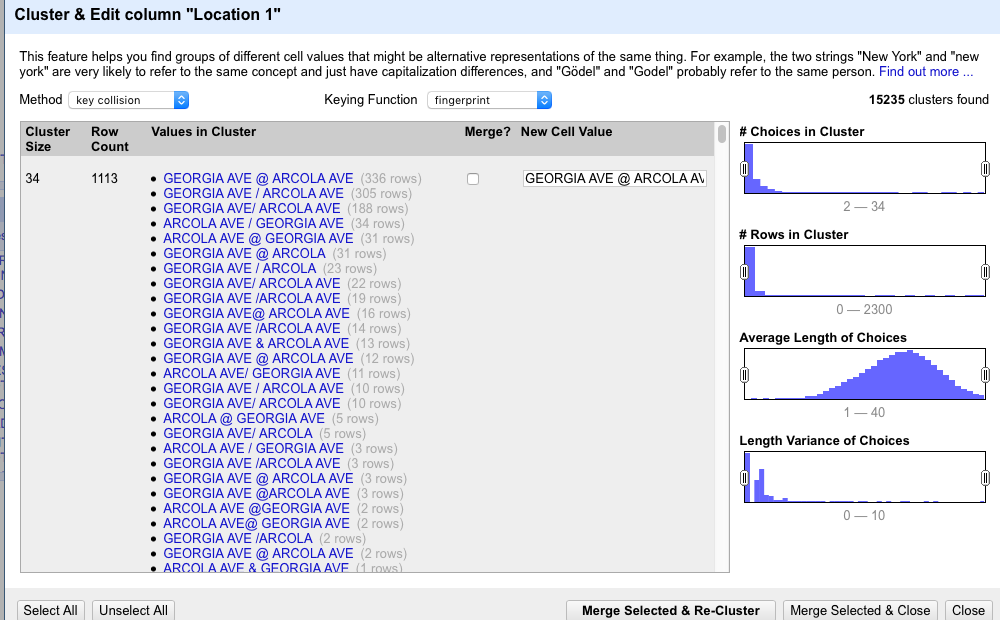
* NEAREST NEIGHBOR – PPM



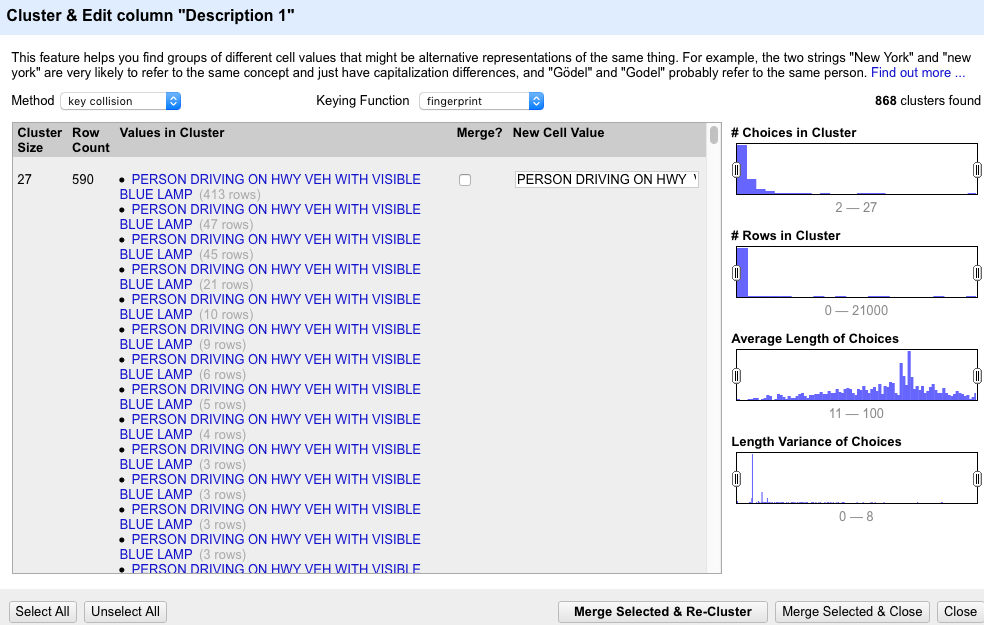
* SUMMARY REPLACE



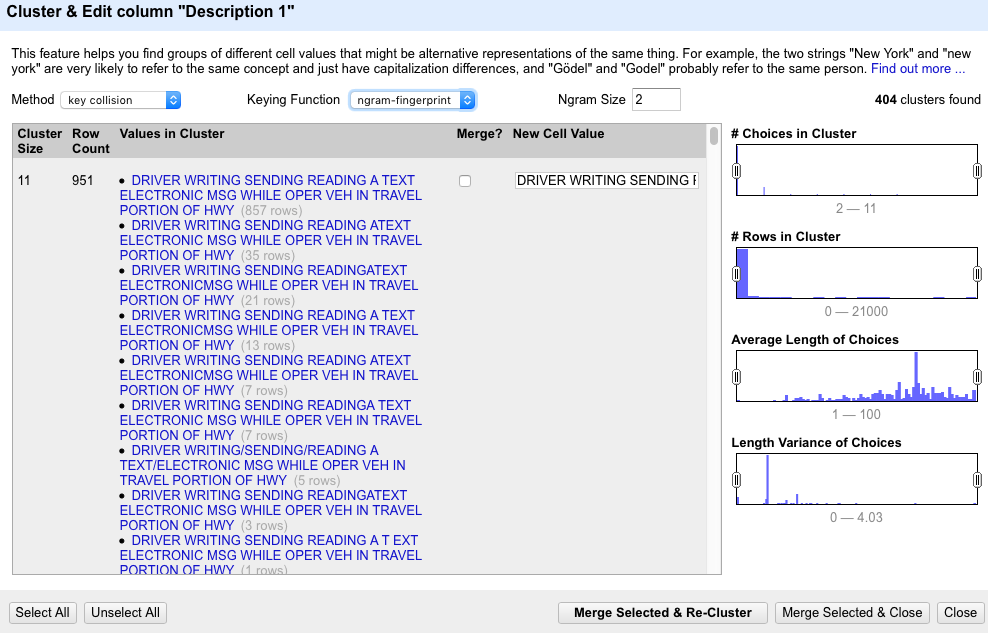
* LOCATION
* KEY COLLISION – FINGERPRINT



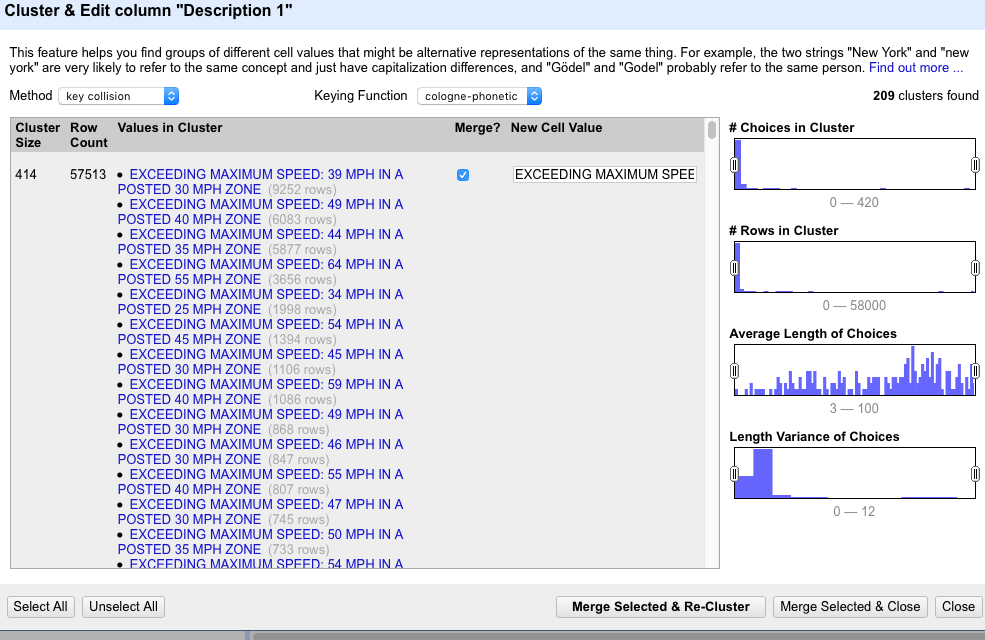
* Aa
* Description
* Key COLLISION – FINGERPRINT



* KEY COLLISION – NGRAM FINGERPRINT



* KEY COLLISION – METAPHONE3: NOTHING GOOD, TO MANY VARIANCES IN ONE CLUSTER
* KEY COLLISION – COLOGNE PHONETIC



* NEAREST NEIGHBOR – LEVENSHTEIN

Lookup:

**SubAgency**

"4th district, Wheaton","204706"

"3rd district, Silver Spring","175382"

"2nd district, Bethesda","121285"

"6th district, Gaithersburg / Montgomery Village","107582"

"5th district, Germantown","96782"

"1st district, Rockville","95418"

"Headquarters and Special Operations","31919"

**Accident,Belts,Personal Injury,Property Damage,Fatal,Commercial License, HAZMAT,Commercial Vehicle, Alcohol, Work Zone,** **Contributed To Accident**

"No","802292"

"Yes","30782"

**State**

"MD","731226"

"VA","33901"

"DC","20686"

"XX","6799"

"PA","5885"

"FL","3672"

"NC","3049"

"WV","2937"

"NY","2817"

"NJ","2382"

"TX","2298"

"GA","1863"

"DE","1503"

"OH","1430"

"CA","1388"

"MA","1103"

"IL","908"

"SC","894"

"TN","749"

"MI","679"

"IN","576"

"CT","522"

"AZ","512"

"AL","388"

"US","340"

"KY","311"

"WA","291"

"OK","290"

"MO","267"

"LA","256"

"MS","256"

"ME","242"

"CO","239"

"MN","228"

"WI","220"

"RI","178"

"NH","165"

"NV","144"

"IA","133"

"NM","128"

"OR","126"

"UT","126"

"KS","116"

"AR","102"

"ON","94"

"VT","84"

"AK","80"

"ID","72"

"MT","68"

"ND","53"

"MB","46"

"HI","41"

"NE","36"

"WY","35"

"NB","34"

"SD","32"

"QC","17"

"VI","15"

"AB","11"

"PR","11"

"GU","6"

"BC","4"

"PQ","4"

"NS","3"

"IT","1"

"MH","1"

"PE","1"

**VehicleType**

"02 - Automobile","726588"

"05 - Light Duty Truck","48515"

"28 - Other","15767"

"03 - Station Wagon","13958"

"01 - Motorcycle","8381"

"06 - Heavy Duty Truck","7159"

"29 - Unknown","5574"

"08 - Recreational Vehicle","2693"

"25 - Utility Trailer","1024"

"19 - Moped","848"

"07 - Truck/Road Tractor","722"

"04 - Limousine","611"

"20 - Commercial Rig","399"

"10 - Transit Bus","282"

"12 - School Bus","141"

"27 - Farm Equipment","106"

"09 - Farm Vehicle","67"

"21 - Tandem Trailer","58"

"26 - Boat Trailer","48"

"11 - Cross Country Bus","42"

"23 - Travel/Home Trailer","20"

"22 - Mobile Home","16"

"13 - Ambulance(Emerg)","13"

"18 - Police(Non-Emerg)","10"

"14 - Ambulance(Non-Emerg)","8"

"24 - Camper","7"

"15 - Fire(Emerg)","4"

"16 - Fire(Non-Emerg)","4"

"18 - Police Vehicle","4"

"17 - Police(Emerg)","3"

"14 - Ambulance","2"

**Violation Type**

"Citation","397017"

"Warning","387198"

"ESERO","48833"

"SERO","26"

**Article**

"Transportation Article","777648"

"","48869"

"Maryland Rules","6557"

"<3,0>","<3,1>"

**Race**

"WHITE","303588"

"BLACK","260339"

"HISPANIC","174664"

"ASIAN","49034"

"OTHER","43102"

"NATIVE AMERICAN","2347"

**Gender**

"M","556289"

"F","275975"

"U","810"

Vehicle Type and Arrest Type can be separated by dash -

Init cleaning

ontangs-i7:lab-data-cleaning-project nikolausn$ ./data-cleaning-framework.py init -in ../Traffic\_Violations.csv -out testTraffic1 -c data-cleaning-config-2.json

make merge openrefine ready

#select particular fields for merging

./data-cleaning-framework.py merge -in testTraffic1 -out testTraffic2 –f ‘Model 1,Make 1,Description 1,Driver City 1,Location 1’

#massedit using config from openrefine

./data-cleaning-framework.py massedit -in test2Traffic1 -out test2Traffic3 -c testproduction.json

#select particular field for loading

./data-cleaning-framework.py select -in testTraffic1 -out testTraffic2 –f ‘Date Of Stop,Time Of Stop,Agency,SubAgency,Latitude,Longitude,Accident,Belts,Personal Injury,Property Damage,Fatal,Commercial License,HAZMAT,Commercial Vehicle,Alcohol,Work Zone,State,Year,Violation Type,Article,Contributed To Accident,Race,Gender,Driver State,DL State,Geolocation,Location 1,Color 1,Charge 1,Arrest Type 1,Arrest Type 2,VehicleType 1,VehicleType 2,Description 1 1,Make 1 1,Model 1 1,Driver City 1 1’

#load to database