ETL Project

Navya Roy

Ashley Oakes

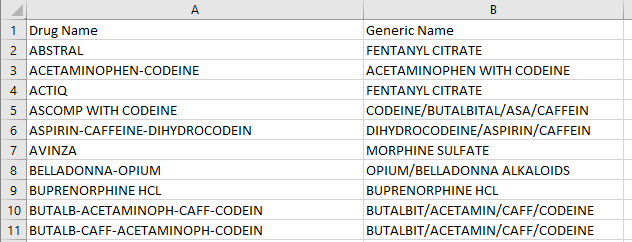
Joe Comeaux

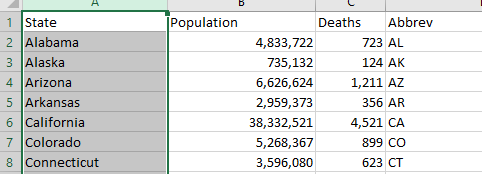
August 24, 2019

**U.S. Opiate Prescriptions/Overdoses**

**Introduction**

Our group decided we wanted to do something within the medical field, so accessed a dataset on the U.S. Opiate Prescriptions/Overdoses from the Kaggle (kaggle.com) web site. This dataset, which is for 2014, is meant to be used for predictive modeling. There are 3 csv files in the collection:

1. Prescriber-info.csv - list of 25,000 rows, 1 per doctor, that provides some info, like National Practitioner Identifier, the state, gender, credentials and specialty. The also lists the number of prescriptions the doctor has written for over 250 drugs. Please see the list of columns at the end of this report and the original file has been uploaded with this report.
2. Opioids.csv - provides a list of drugs that are opioids. This list provides the generic name and the names used by the company that created the drug. Here is a screen shop of the file, plus we uploaded the original
3. Overdoses.csv - contains the number of deaths from opioids and total population for each state. Below is a screen shot of the file



**Database Choice**

Examination of the prescriber-info.csv data shows that the drug list has over 250 entries and most of the time, for a given doctor, most of the entries are 0 (which makes sense, since most doctors would not prescribe that many drugs). Our plan is to use a cloud-based database, so this many columns might cause issues with the free cloud-based options, plus, this is an inefficient way to store the data if most of the columns are usually 0. It is also the case that none of the columns are always zero, so we could not eliminate any columns. This leads to the conclusion that a MongoDB database would best fit our needs. The advantage of Mongo is that we could eliminate the columns that are zero for each doctor. As our data is relatively small (25,000 records), we do not have to worry about query speed under Mongo. Our records will also be fairly small (usually less than 50 items which will mostly be a key and an integer), so this also means query speed should not be an issue using Mongo. Another advantage is that Mongo also has a free cloud-based DB server we can use.

**Cleaning**

*(Note that all steps are well documented in our program)*

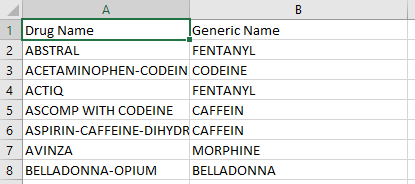
Our next step was to determine what data from each of the tables we needed to include in our database. Since we are using Mongo, we will need to have a self-contained collection without any secondary cross-reference tables, which in turn means we have to design a document that will contain all needed info.

Our initial dataset, the prescriber-info file, does not indicate if any of the drugs are opioids. This info is contained in the opioids.csv file, so our next step is to reconcile these 2 files. We started by reading both into a pandas dataframe and tried to match drug names between the two files. This was a disaster as 0 drugs matched. The first problem was that the opioids list was right justified, which means there was extra white space to the left of each name, so we eliminated this using the python lstrip method for strings. The drug name in the presciber-info file also contained periods instead of spaces or dashes for word breaks, so we changed the periods to dashes in the program.

Our opioids file also contains both a generic name and an industry name given by the company that developed said drug. Upon inspection of the list, it was clear that the industry name would be very complicated to reconcile between the two lists because of inconsistent naming conventions. Plus, we really only want to use the generic opioid names as they are way more descriptive as what is happening, so we decided to use the generic names. It was clear that even the generic names would be difficult to match as is and would need attention.

We discussed handling this step using code, but decided to manually edit the file as there were only about 100 entries and most of the generic names were pretty straight forward, with a base name, like oxycontin, and then a secondary name appended to the end, which usually did not match what was used in the prescriber-info. However, there were enough differences in the special cases to likely lead to several need special case rules for coding, which we could easily manually handled in a more timely fashion.

By shortening the opioid list to only the ‘generic’ drug name meant that this was often a subset of the longer names used in the prescriber-info file, so problem solved… we could now indicate which drugs on the list were opioids. Below is a screen shot of the cleaned up opioids-clean.csv file.



The third file contains data on the number of deaths from opioid overdoses (overdoses.csb) by state, and also provides the total population for each state.

The overdose file was slurped into a dataframe and compared to the prescriber-info data. This indicated several issues where several state abbreviations in the prescriber-info file that were not in the overdoses file. Inspection of these showed that they were for Puerto Rico, Washington DC and several branches of the military. We felt that the state info was somewhat of a secondary goal of the project, so instead of eliminating these records, we would set the state data to a missing value of -1, which would be easy to filter out if need be. This could have been done by editing the original overdoses file and adding the state codes with -1 values for all data, however, in this case, this was easy to do in the program. The advantage of this option is that if more state codes are found in the future, we would not have to edit the file again, just add new rules to the code.

At this time, we had reconciled our 3 files and decided what data we needed from each, however, we needed to decide how to indicate which drugs were opioids in our database. One option was to append ‘opioid’ to the drug name or capitalize only the opioid drugs. We decided the best thing would be to split the drug list into 2… one for opioid drugs and one for non-opioid drugs. This means that documents will have 2 sub-dictionaries.

One other concern was that the drug list also contained a flag that indicated if each doctor prescribed opioids. This was included in our drug dictionary, but we wanted this info in the primary info part of our document, so this was moved from the drug list into the primary part so that we could easily query and get all doctors that prescribed opioids.

Inspection of our document lead to the conclusion that it would be nice to add 2 computed fields to our primary info which would track the total number of opioid prescriptions each doctor wrote and the total number of non-opioid prescriptions written by the doctor. We felt this would be extremely useful when examining how many doctors may be leading to the opioid prescription abuse.

We now have a document we felt meant our goals. Please see below for a screenshot of one of the documents in the mongodb collection.

**The keys in our document**

**NPI, Gender, State, Credentials** and **Specialty** are taken from the *prescriber-info.csv* file, as is.

The **Deaths** and **Population** come from the *overoses.csv* file.

The **NumberofOpioids** and **NumberofNonopiods** are *computed* by us and are taken from a combination of the *prescriber-info.csv* and *opioids.csv* files.

The **Drugs** list comes from the *prescriber-info.csv* file, where we eliminated all entries that were 0. 

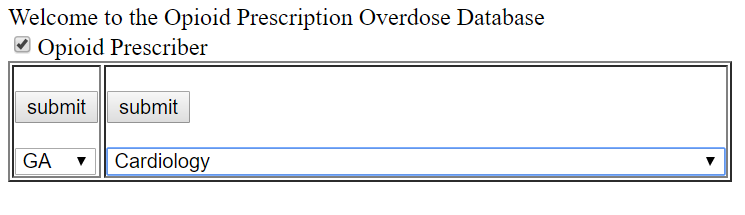
The **Opioids** dictionary is combined using the *prescriber-info.csv* file and the *opiods.csv* file.

It should be noted that in any case where we manually changed the original data files, we kept an original version and changed a copy, as should always be done.

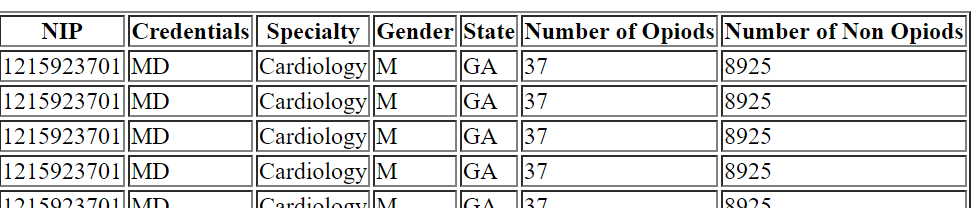
**Web Interface**

We also created a web interface to our database. This will read the db and create queries based on user input and show the results. This has been done using flash with templates and pymongo.

Here is a screenshot of the front page: where we query all doctors that have prescribed opioids from Georgia with a specialty of Cardiology.



And the results look like:



Appendix I - List of all columns in the prescriber-info.csv file

NPI

Gender

State

Credentials

Specialty

ABILIFY

CODEINE

ACYCLOVIR

ADVAIR.DISKUS

AGGRENOX

ALENDRONATE.SODIUM

ALLOPURINOL

ALPRAZOLAM

AMIODARONE.HCL

AMITRIPTYLINE.HCL

AMLODIPINE.BESYLATE

AMLODIPINE.BESYLATE.BENAZEPRIL

AMOXICILLIN

AMOX.TR.POTASSIUM.CLAVULANATE

AMPHETAMINE.SALT.COMBO

ATENOLOL

ATORVASTATIN.CALCIUM

AVODART

AZITHROMYCIN

BACLOFEN

BD.ULTRA.FINE.PEN.NEEDLE

BENAZEPRIL.HCL

BENICAR

BENICAR.HCT

BENZTROPINE.MESYLATE

BISOPROLOL.HYDROCHLOROTHIAZIDE

BRIMONIDINE.TARTRATE

BUMETANIDE

BUPROPION.HCL.SR

BUPROPION.XL

BUSPIRONE.HCL

BYSTOLIC

CARBAMAZEPINE

CARBIDOPA.LEVODOPA

CARISOPRODOL

CARTIA.XT

CARVEDILOL

CEFUROXIME

CELEBREX

CEPHALEXIN

CHLORHEXIDINE.GLUCONATE

CHLORTHALIDONE

CILOSTAZOL

CIPROFLOXACIN.HCL

CITALOPRAM.HBR

CLINDAMYCIN.HCL

CLOBETASOL.PROPIONATE

CLONAZEPAM

CLONIDINE.HCL

CLOPIDOGREL

CLOTRIMAZOLE.BETAMETHASONE

COLCRYS

COMBIVENT.RESPIMAT

CRESTOR

CYCLOBENZAPRINE.HCL

DEXILANT

DIAZEPAM

DICLOFENAC.SODIUM

DICYCLOMINE.HCL

DIGOX

DIGOXIN

DILTIAZEM.24HR.CD

DILTIAZEM.24HR.ER

DILTIAZEM.ER

DILTIAZEM.HCL

DIOVAN

DIPHENOXYLATE.ATROPINE

DIVALPROEX.SODIUM

DIVALPROEX.SODIUM.ER

DONEPEZIL.HCL

DORZOLAMIDE.TIMOLOL

DOXAZOSIN.MESYLATE

DOXEPIN.HCL

DOXYCYCLINE.HYCLATE

DULOXETINE.HCL

ENALAPRIL.MALEATE

ESCITALOPRAM.OXALATE

ESTRADIOL

EXELON

FAMOTIDINE

FELODIPINE.ER

FENOFIBRATE

FENTANYL

FINASTERIDE

FLOVENT.HFA

FLUCONAZOLE

FLUOXETINE.HCL

FLUTICASONE.PROPIONATE

FUROSEMIDE

GABAPENTIN

GEMFIBROZIL

GLIMEPIRIDE

GLIPIZIDE

GLIPIZIDE.ER

GLIPIZIDE.XL

GLYBURIDE

HALOPERIDOL

HUMALOG

HYDRALAZINE.HCL

HYDROCHLOROTHIAZIDE

ACETAMIN

HYDROCORTISONE

HYDROMORPHONE

HYDROXYZINE.HCL

IBANDRONATE.SODIUM

IBUPROFEN

INSULIN.SYRINGE

OPIUM

IRBESARTAN

ISOSORBIDE.MONONITRATE.ER

JANTOVEN

JANUMET

JANUVIA

KETOCONAZOLE

KLOR.CON.10

KLOR.CON.M10

KLOR.CON.M20

LABETALOL.HCL

LACTULOSE

LAMOTRIGINE

LANSOPRAZOLE

LANTUS

LANTUS.SOLOSTAR

LATANOPROST

LEVEMIR

LEVEMIR.FLEXPEN

LEVETIRACETAM

LEVOFLOXACIN

LEVOTHYROXINE.SODIUM

LIDOCAINE

LISINOPRIL

LISINOPRIL.HYDROCHLOROTHIAZIDE

LITHIUM.CARBONATE

LORAZEPAM

LOSARTAN.HYDROCHLOROTHIAZIDE

LOSARTAN.POTASSIUM

LOVASTATIN

LOVAZA

LUMIGAN

LYRICA

MECLIZINE.HCL

MELOXICAM

METFORMIN.HCL

METFORMIN.HCL.ER

METHADONE

METHOCARBAMOL

METHOTREXATE

METHYLPREDNISOLONE

METOCLOPRAMIDE.HCL

METOLAZONE

METOPROLOL.SUCCINATE

METOPROLOL.TARTRATE

METRONIDAZOLE

MIRTAZAPINE

MONTELUKAST.SODIUM

MORPHINE

MORPHINE

MUPIROCIN

NABUMETONE

NAMENDA

NAMENDA.XR

NAPROXEN

NASONEX

NEXIUM

NIACIN.ER

NIFEDICAL.XL

NIFEDIPINE.ER

NITROFURANTOIN.MONO.MACRO

NITROSTAT

NORTRIPTYLINE.HCL

NOVOLOG

NOVOLOG.FLEXPEN

NYSTATIN

OLANZAPINE

OMEPRAZOLE

ONDANSETRON.HCL

ONDANSETRON.ODT

ONGLYZA

OXCARBAZEPINE

OXYBUTYNIN.CHLORIDE

OXYBUTYNIN.CHLORIDE.ER

ACETAMIN

OXYCODONE

OXYCONTIN

PANTOPRAZOLE.SODIUM

PAROXETINE.HCL

PHENOBARBITAL

PHENYTOIN.SODIUM.EXTENDED

PIOGLITAZONE.HCL

POLYETHYLENE.GLYCOL.3350

POTASSIUM.CHLORIDE

PRADAXA

PRAMIPEXOLE.DIHYDROCHLORIDE

PRAVASTATIN.SODIUM

PREDNISONE

PREMARIN

PRIMIDONE

PROAIR.HFA

PROMETHAZINE.HCL

PROPRANOLOL.HCL

PROPRANOLOL.HCL.ER

QUETIAPINE.FUMARATE

QUINAPRIL.HCL

RALOXIFENE.HCL

RAMIPRIL

RANEXA

RANITIDINE.HCL

RESTASIS

RISPERIDONE

ROPINIROLE.HCL

SEROQUEL.XR

SERTRALINE.HCL

SIMVASTATIN

SOTALOL

SPIRIVA

SPIRONOLACTONE

SUCRALFATE

SULFAMETHOXAZOLE.TRIMETHOPRIM

SUMATRIPTAN.SUCCINATE

SYMBICORT

SYNTHROID

TAMSULOSIN.HCL

TEMAZEPAM

TERAZOSIN.HCL

TIMOLOL.MALEATE

TIZANIDINE.HCL

TOLTERODINE.TARTRATE.ER

TOPIRAMATE

TOPROL.XL

TORSEMIDE

TRAMADOL.HCL

TRAVATAN.Z

TRAZODONE.HCL

TRIAMCINOLONE.ACETONIDE

TRIAMTERENE.HYDROCHLOROTHIAZID

VALACYCLOVIR

VALSARTAN

VALSARTAN.HYDROCHLOROTHIAZIDE

VENLAFAXINE.HCL

VENLAFAXINE.HCL.ER

VENTOLIN.HFA

VERAPAMIL.ER

VESICARE

VOLTAREN

VYTORIN

WARFARIN.SODIUM

XARELTO

ZETIA

ZIPRASIDONE.HCL

ZOLPIDEM.TARTRATE

Opioid.Prescriber

# U.S. Opiate Prescriptions/Overdoses

# U.S. Opiate Prescriptions/Overdoses

# U.S. Opiate Prescriptions/Over