Predicting Opioid Drug Prescriptions among Prescribing Specialists

Jimmy Smart

1. Introduction and Data background

Accidental death by fatal drug overdose is a rising trend in the United States. What can we do to help?

Below is a dataframe that I will be using to dive into the opioid crisis as I wrangle through and provide visuals that pertain to the csv dataframes I have available.

The 1st one, opioid.csv, lists the known opioid drug names and their generic names. The Overdoses.csv contains data provided from the 50 states in regards to their population and opioid related deaths. Finally, the Prescriber-info.csv provides data regarding a specified prescriber. their gender, state, and speciality are provided, as well as the drugs they prescribed to their patients. Opioids and non opioid drugs.

This dataset contains summaries of prescription records for 250 common opioid and non-opioid drugs written by nearly 25,000 unique licensed medical professionals in 2014 in the United States for citizens covered under Class D Medicare as well as some metadata about the doctors themselves. This is only a small subset of data that was sourced from a much larger file: cms.gov.

The data was acquired from Kaggle: U.S. Opiate Prescriptions/Overdoses by Alan “AJ” Pryor, Ph.D. The full dataset contains almost 24 million prescription instances in long format. In the Kaggle form, the data has already been previously cleaned and compiled here in a format with 1 row per prescriber and limited the approximately 1 million total unique prescribers down to 25,000 to keep it manageable.

2. Key for reading the dataset

NPI – unique National Provider Identifier number
Gender - (M/F)
State - US State by abbreviation
Credentials - set of initials indicative of medical degree
Specialty - description of type of medicinal practice (109 unique specialties)
Opioid.Prescriber - a boolean label indicating whether or not that individual prescribed opiate drugs more than 10 times in the year

3. Data wrangling and cleaning

The original datasets called *opioid.csv*, *overdoses.csv* and *prescriber-info.csv* were loaded in as 3 dataframes into a Jupyter Notebook. Once the data was loaded, the shape of the main dataset (prescriber-info aka prescriber) was confirmed to be 24759 rows by 256 columns. The datasets (opioid and overdoses) were confirmed to be 113 rows × 2 columns and 50 rows × 2 columns respectively.

a. Selecting initial variables

An initial glance at the dataset revealed certain variables to be redundant and some containing missing values or contain many unique categories, all of which would provide little information during further analysis or during the modeling process. The following variables were excluded from our dataset with the reasoning provided beside each one:

b. Redundant variables from *prescriber* dataset

- Credentials: provided a set of initials indicative of medical degree. They were deemed redundant due to the more important *Specialty* variable
- Within the *State* variable we removed certain US territories such as (PR, AA, GU, AE, ZZ) as we just want to focus on the 50 states. DC was merged into VA

c. Finding any null or missing values

The number of missing values for each feature was determined using the prescriber.isnull().sum() method in Python which provided a list of all the variables along with the number of non-null values in each as well as the data type of each variable providing a good overview of the structure of the dataset. *Credentials* variable was the only variable with any null returns. Since we are dropping *Credentials* anyway from our dataset, we can continue on.

d. Identifying opioid drug names

Now that we have cleaned up our dataframe a bit we can start looking at the data a bit better. We will start by identifying opioids by comparing drug names found in the *prescriber* dataframe and drug names found in the *opioid* dataframe. Identifying the opioids will help us with our various comparisons between regularly prescribed drugs and opioids.

List of opioids:

MORPHINE.SULFATE.ER
METHADONE.HCL
HYDROCODONE.ACETAMINOPHEN
ACETAMINOPHEN.CODEINE
HYDROMORPHONE.HCL

MORPHINE.SULFATE
TRAMADOL.HCL
FENTANYL
OXYCODONE.HCL
OXYCODONE.ACETAMINOPHEN
OXYCONTIN

e. Identifying sum of opioids and non opioids prescribed by specialists

Using the code we just used to identify the opioid drug names, we were able to create a code to find the sum of opioids prescribed by each specialist in our dataset.

```
prescriber['SumOpi'] = prescriber[opi_presc].sum(axis=1)
```

As well as the sum total number of prescriptions written by each specialists:

```
prescriber['TotPresc'] = (prescriber.iloc[:,4:254]).sum(axis=1)
```

With ‘SumOpi and ‘TotPresc’ we were able to create a 3rd column consisting of the non opioids:

```
prescriber['NonOpi'] = prescriber['TotPresc'] - prescriber[opi_presc].sum(axis=1)
```

f. Looking at the data with a focus on the States

We will create a bar graph representing the 50 states and compare total prescriptions vs total opioids prescribed. **Fig 1**. shows our results. Red represents the total amount of prescriptions while the blue represents the total number of opioids prescribed.

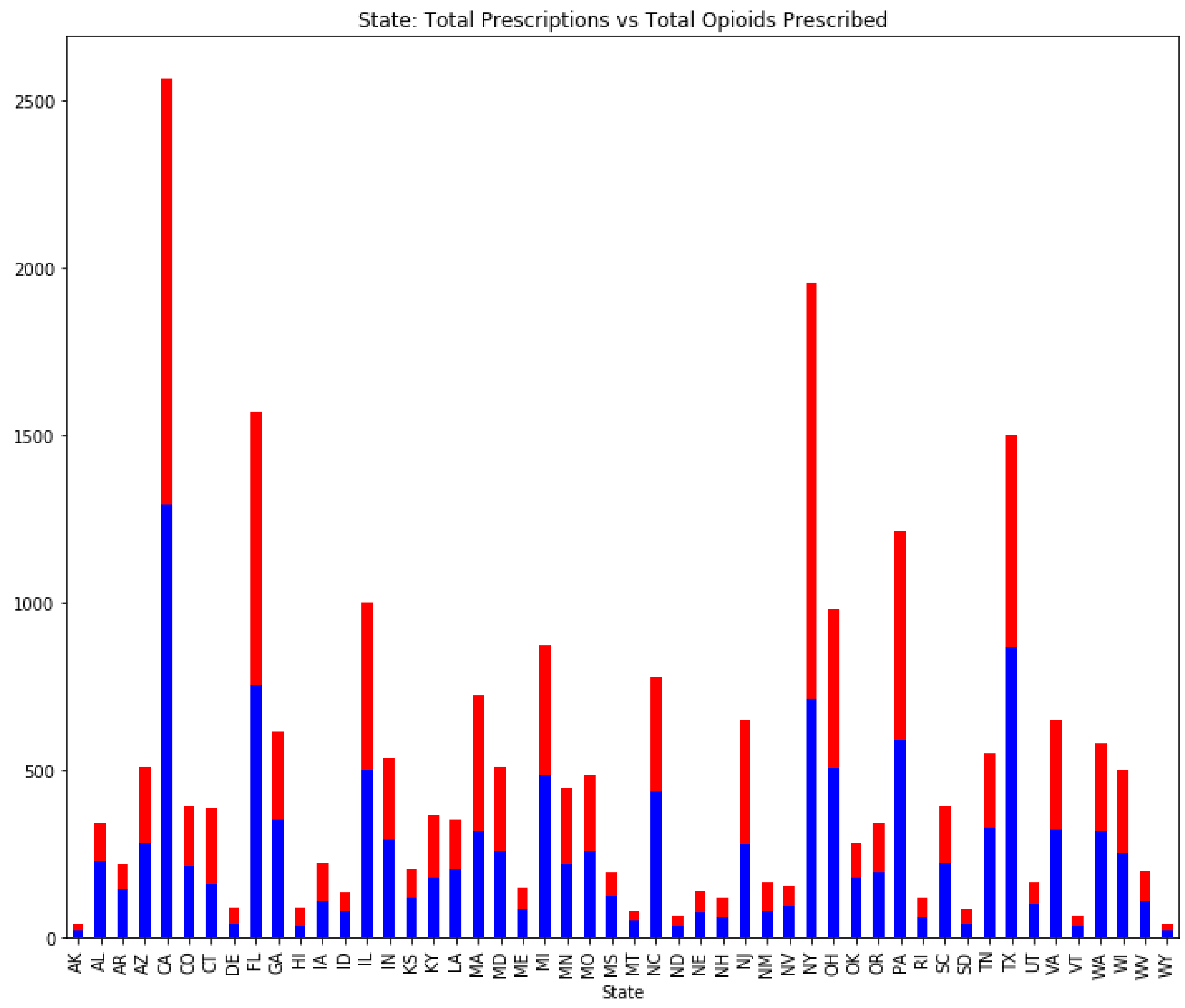
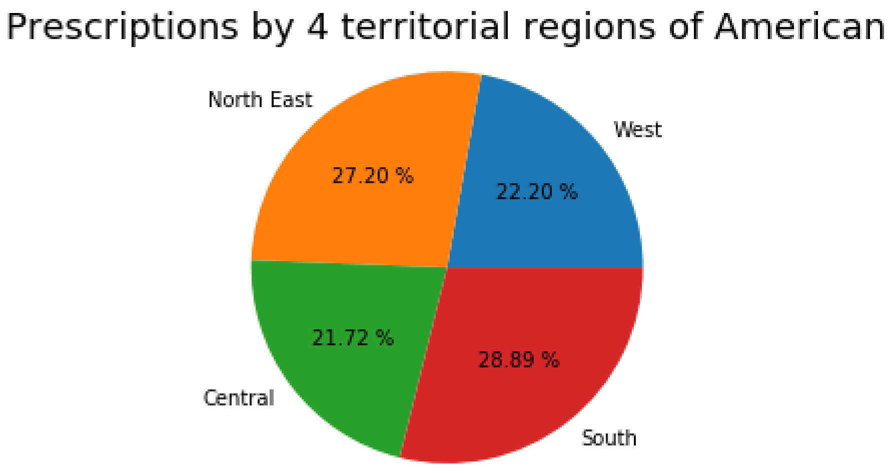


Fig 1. Total Prescriptions vs Total Opioids Prescribed

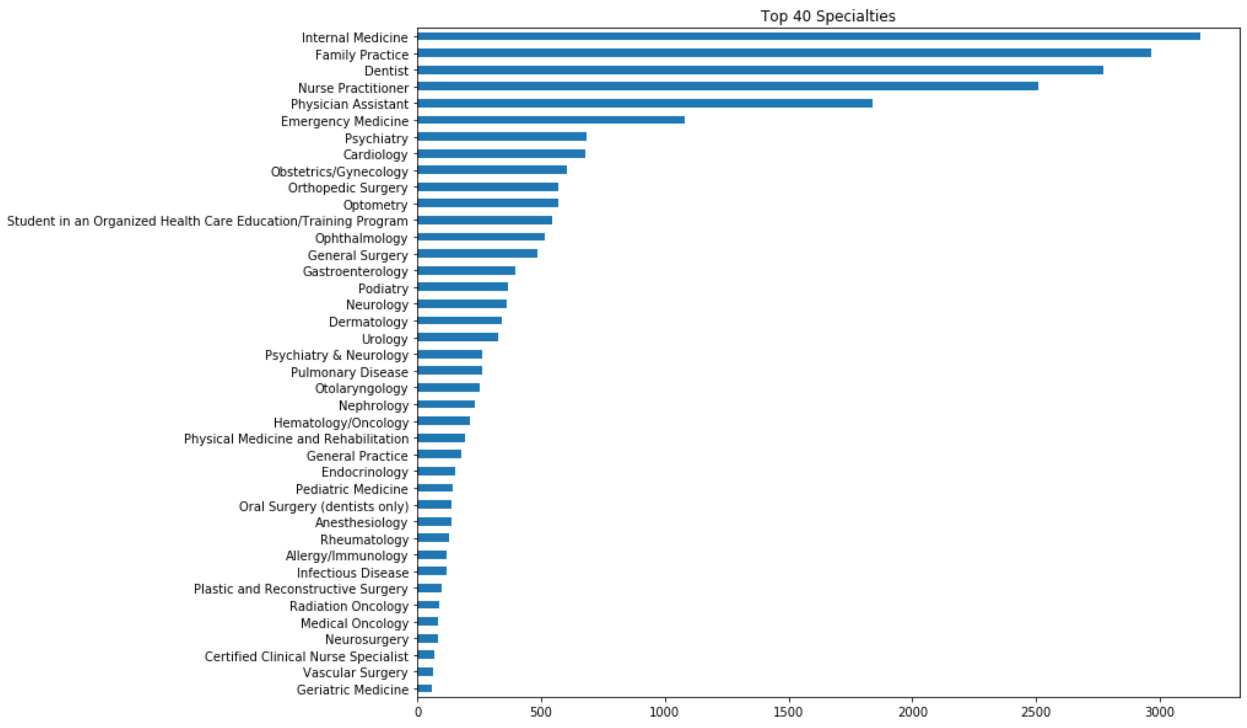
We then grouped the states into 4 geographical categories represented by: North East, Central, South and West. Each region consists of either 12-13 states within their groups. **Fig 2.** shows a pie chart of the breakdown of our results.



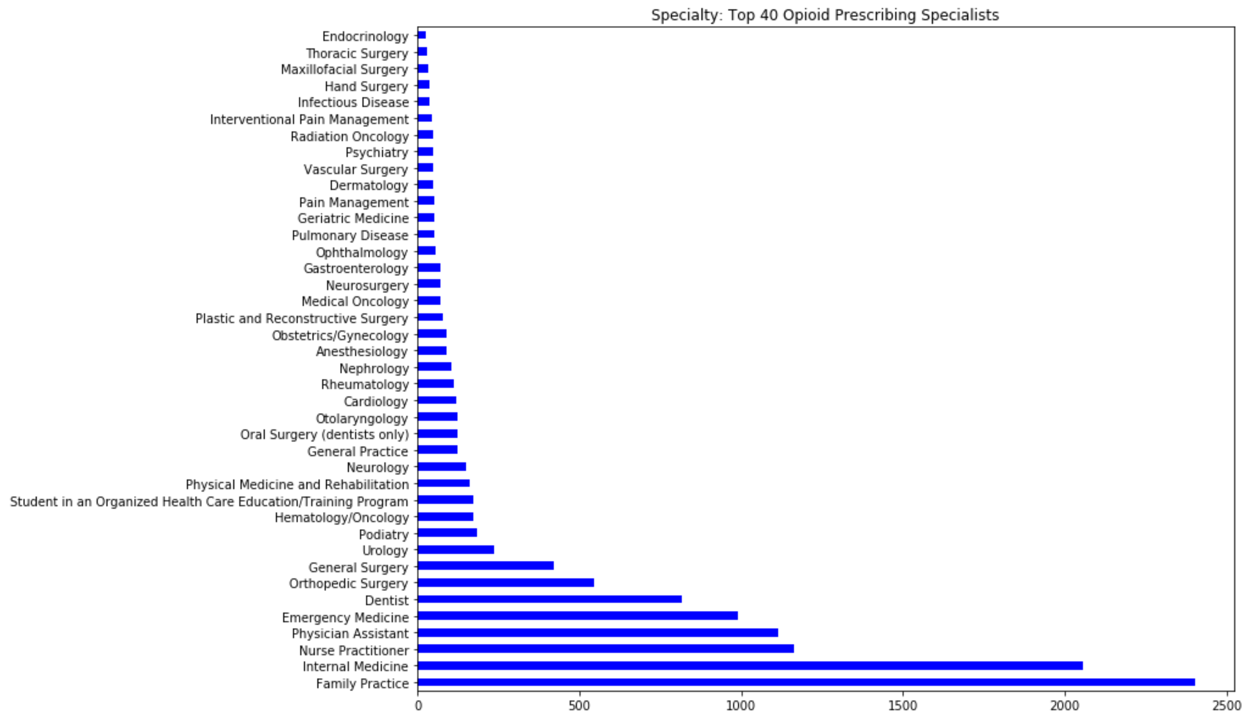
When comparing **Fig 1.** and **Fig 2.** Individually CA and NY lead the way but overall the South carries a bigger percentage of the pie.

g. Specialties

Specialties are an important piece to our dataset as they are the main focus of this Capstone in trying to find a trend within the various specialties. For this part of our data modeling we 1st took a look at the value counts. **Fig 3.** Shows the graph we created to show the value counts for the Top 40 various specialties.



With **Fig 4.** We wanted to show the value counts for the Top 40 specialties who prescribed opioids.



As we can see in **Fig 3.** and **Fig 4.** there is some difference between the respected Top 40s of each graph. Internal Medicine, Nurse Practitioner and Family Practice are at or near the top of each figure.

We created a chart for the top 5 specialists who prescribed opioids and their percentages in reference to the rest of the specialties:

Percentage of Prescriptions from the top 5 prescribers

	Value Opi	**Percent Opi**
Family Practice	2402	18.93
Internal Medicine	2057	16.21
Nurse Practitioner	1165	9.18
Physician Assistant	1113	8.77
Emergency Medicine	990	7.80

h. Gender

First and foremost we wanted to see the percentage differences there were present between Males and Female specialists who didn't prescribe opioids as well as a pie chart of total differences between Males and Females within our dataset. **Fig 5.** represents a pie chart showing the results.

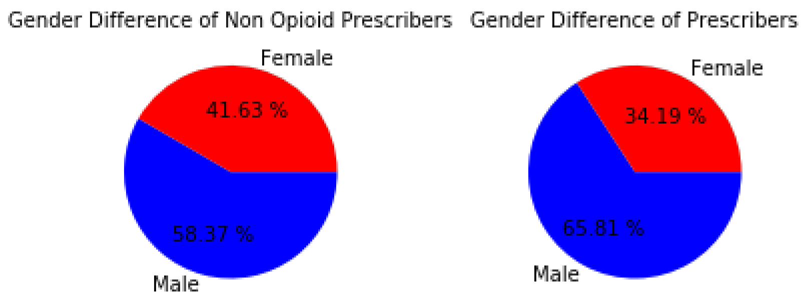
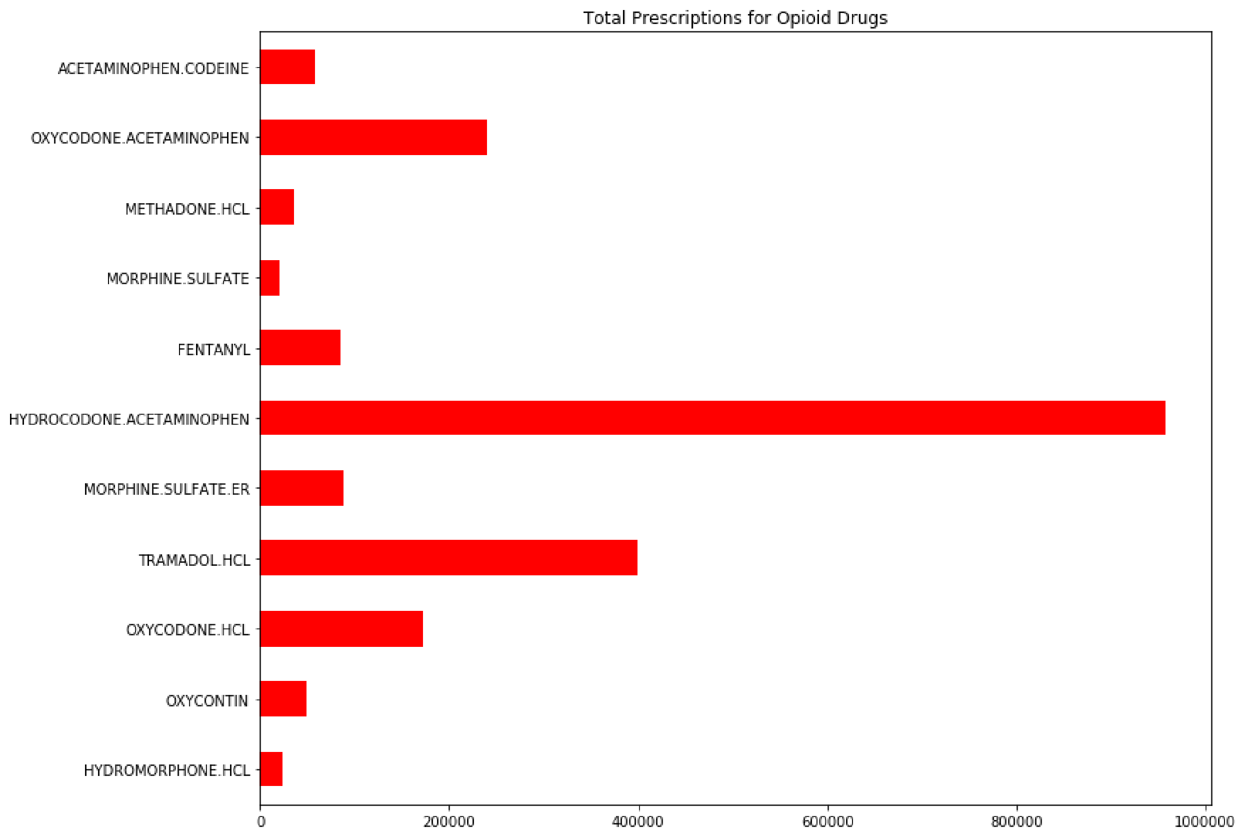


Fig 5. Gender Difference of Non Opioid Prescribers and Gender Difference of Prescribers

Overall there were 5025 Females who didn't prescribe opioids and 7046 Males who also didn't prescribe opioids.

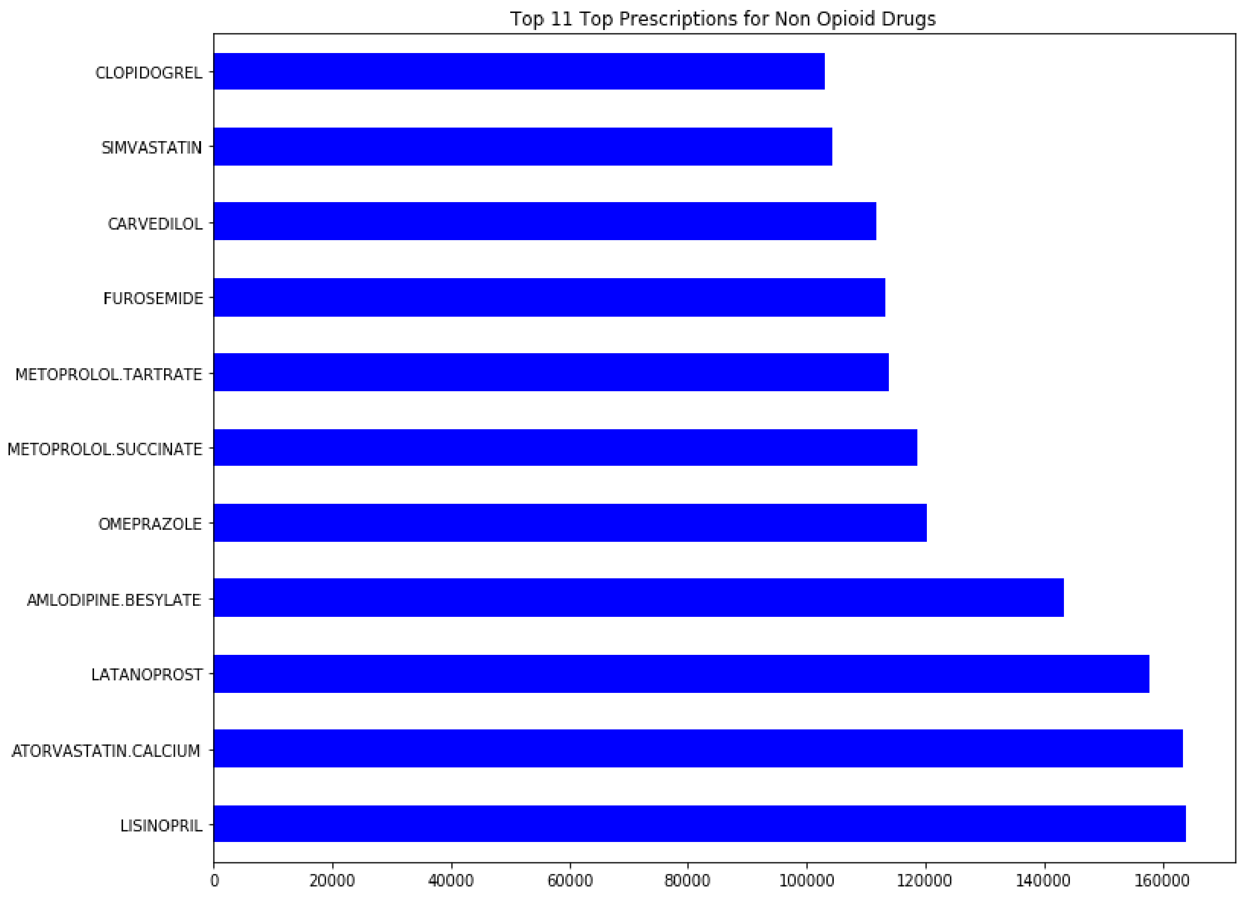
i. Drugs

We took a look at the various drugs available in our dataset. As mentioned previously, we were able to identify the known opioids found within our dataset. We 1st looked at the value counts of these drugs. **Fig 6.** shows our results.



HYDROCODONE.ACETAMINOPHEN	957439
TRAMADOL.HCL	398734
OXYCODONE.ACETAMINOPHEN	240134
OXYCODONE.HCL	172286
MORPHINE.SULFATE.ER	88190
FENTANYL	85344
ACETAMINOPHEN.CODEINE	58477
OXYCONTIN	49727
METHADONE.HCL	35789
HYDROMORPHONE.HCL	23159
MORPHINE.SULFATE	20707

Fig 7. shows the top 11 non opioid prescribed drugs.



4. Summary

We just wanted to use this section to get to know our datasets. As mentioned at the beginning, we inherited a fairly clean dataset so we didn't have to spend too much time in the wrangling and cleaning area. We did want to make the dataset a bit of our own so we created a few new variables as well as took some variables away. As for data visualization, it was helpful to start seeing some of the trends that presented themselves as we started grouping certain variables together. We can see a trend in which specialists seem to be prescribing the most drugs in general, as well as which ones seem to be prescribing the most opioid related drugs. Considering that most opioid drugs are based on pain relief, it's not surprising to see that certain specialists who deal in medical fields that interact with patients coming to them looking for some sort of pain relief, seem to be the common opioid prescribers. As well as prescribers of non opioid pain relief medications.

5. Statistical Analysis

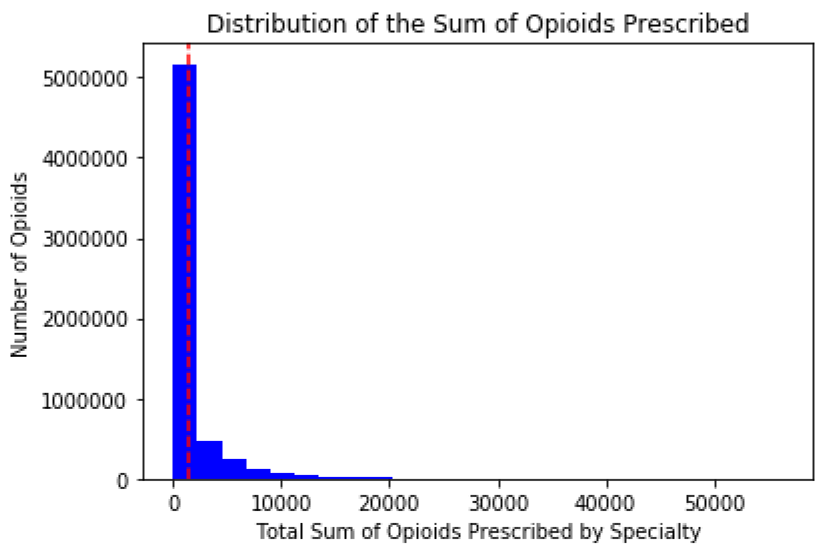
Now that we've spent some time on data wrangling and creating some visuals to better see what our dataset has, we can dive deeper into the statistical information. Our null hypothesis is that certain specialists are prescribing more opioid drugs.

j. Frequentism

We will focus on the Total Prescriptions (TotPresc) variable within our dataset.

mean:	137.205864
Std:	3097.55013

Fig 8. Shows the total sum of opioid prescribed



We have a probability value of (**0.999**) and a critical value of (**1.959**) and confidence intervals of (**1274.135** and **1468.2768**)

k. Creating opioid and non opioid prescribing specialist variables

We separated our specialists from those who prescribed opioid drugs and the specialists who didn't prescribe any opioid drugs.

```
opi_presc = prescriber.TotPresc.loc[prescriber['SumOpi'] >= 1]
non_opi_presc = prescriber.TotPresc.loc[prescriber['SumOpi'] == 0]
```

Results:

Number of Opioids Prescribed:	12688
Number of Specialists who did not prescribe opioids:	12071
Mean of Opioids Prescribed:	2251.877
Standard Deviation of Opioids Prescribed:	4018.83

l. Calculating t-test

To calculate the t-test, 1st we went with a manual approach by calculating the value of the test statistic and then its probability (the p-value). We also used ttest_ind then verified that we got the same results from both.

```
T-test: 47.947716278114676
```

```
Ttest_ind Result(statistic= 47.94582121833751, pvalue=0.0)
```

When we set the equal_var to False we got the result:

```
Ttest_indResult (statistic= 49.00902592890073, pvalue=0.0)
```

We were able to obtain the same t-test results and noticed a slight change when set to False.

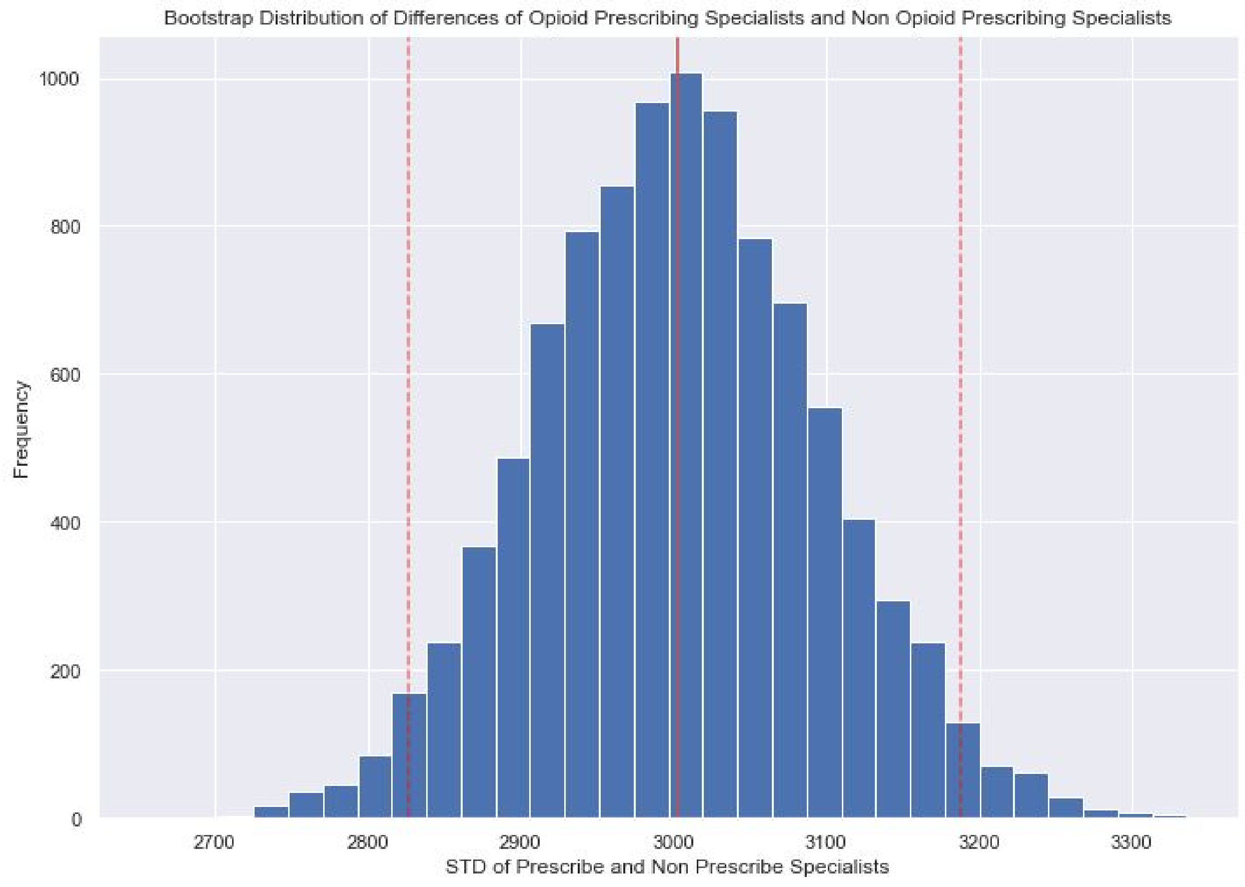
m. Bootstrapping

We'll create a bootstrap sampling to estimate a 95% confidence interval lower limit.

Null Hypothesis: There isn't a difference in the standard deviation between the 2 groups (specialists prescribing opioids and specialists who didn't prescribe opioids)

Alternate Hypothesis: There will be a differ in the standard deviation between the 2 groups (specialists prescribing opioids and specialists who didn't prescribe opioids)

Fig 9. Shows our the Bootstrap replicate with a confidence interval of (**2826.542** and **3187.763**)



Difference of STD for opioid and non opioid prescriptions: *3003.2483421030547*
Difference of STD for bootstrap samples: *3003.302123387791*
The 95% confidence interval for the difference between the standard deviations of opioid and non opioid prescriptions is: [*2826.54281741 3187.76267874*]

n. Confidence interval and p-value

We'll perform a bootstrapped hypothesis test at the 5% significance level (α=0.05) to calculate the p-value of the observed difference between opioid prescribing specialist and non-opioid prescribing specialist.

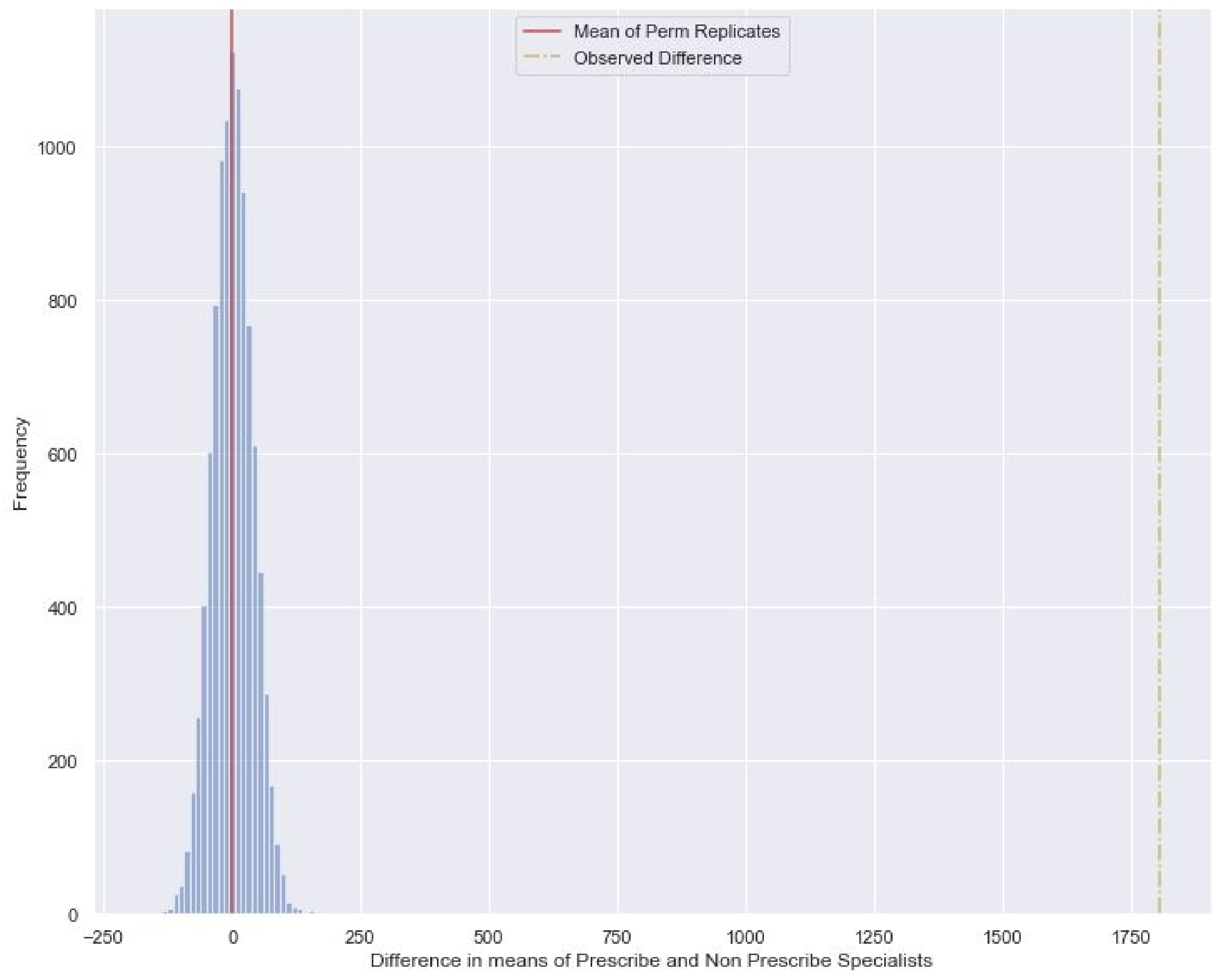
```
def diff_of_means(data_1, data_2):
    diff = np.mean(data_1) - np.mean(data_2)
    return diff

# Compute difference of mean insured and uninsured: empirical_diff_means
empirical_diff_means = diff_of_means(opi_presc, non_opi_presc)

# Draw 10,000 permutation replicates: perm_replicates
perm_replicates = draw_perm_reps(opi_presc, non_opi_presc, diff_of_means,
size=N_rep)

# Compute permutation p-value: perm_p
perm_p = np.sum(perm_replicates >= empirical_diff_means) / len(perm_replicates)

#Compute p-value: p
p = np.sum(bs_perm_rep >= diff_means)/len(bs_perm_rep)


print('Perm p-value = ', perm_p)
print('P-value= ', p)

                Perm p-value =  0.0
                P-value=  0.6619
```

Null hypothesis: There isn't a difference in the mean charges between the 2 groups (specialists prescribing opioids and specialists who didn't prescribe opioids)

Alternate hypothesis: There will be a difference in the mean charges between the 2 groups (specialists prescribing opioids and specialists who didn't prescribe opioids)

With the p-value from the bootstrap being 0.6619, I can reject the alternate hypothesis and accept the null hypothesis

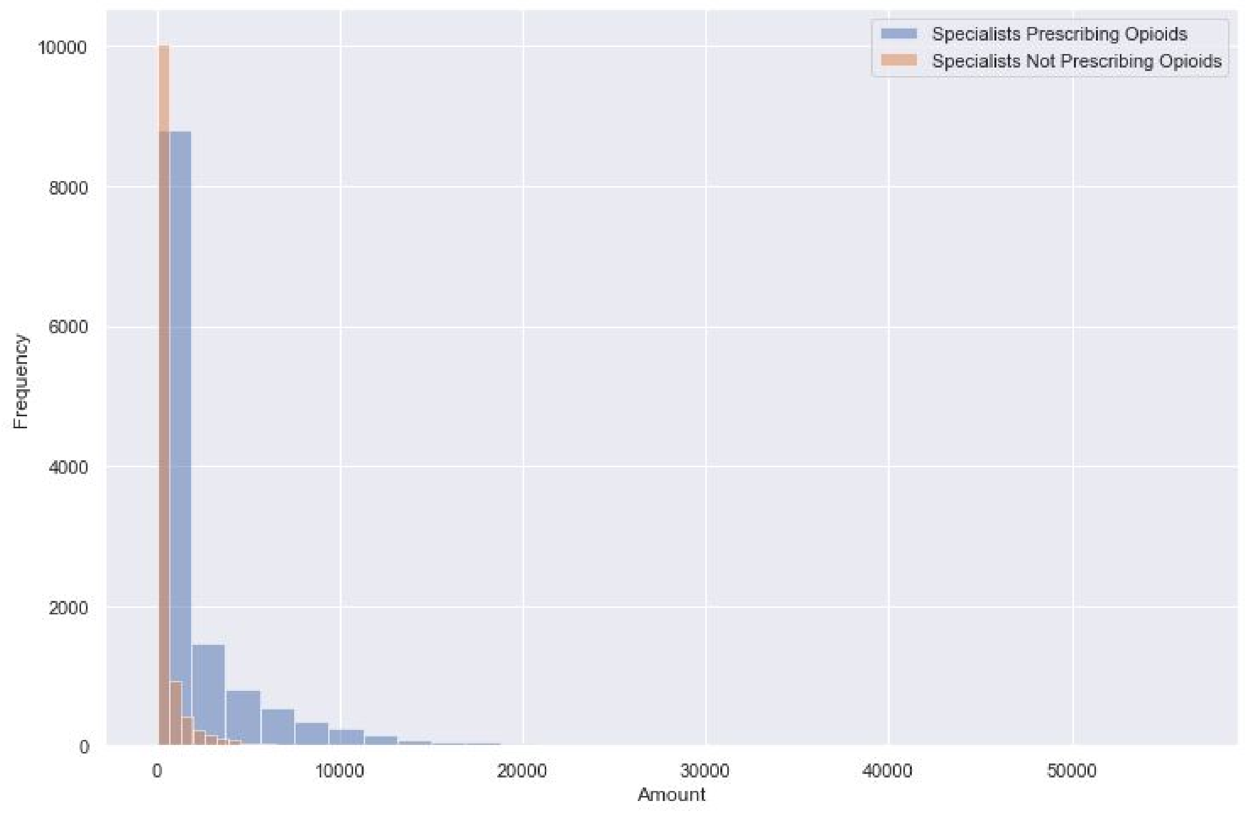
Fig 10. Difference in means of Prescribe and Non Prescribe Specialists



o. Bayesian Inference - Creating a PyMC3 Model

We may suspect from our models that there is some sort of exponential-like distribution at play here. The gamma distribution may be applicable and we could test this for the distribution of prescriptions. Developing our new method for the easiest looking case first is a common and sound approach that can demonstrate a minimum viable solution.

Fig 11. Shows our Specialist who prescribed opioid drugs and non opioids



We created a PyMC3 Model

```
with pm.Model() as model:
    alpha_ = pm.Exponential('alpha', 1/alpha_est)
    rate_ = pm.Exponential('rate', 1/rate_est)
    prescribed_opioids = pm.Gamma('Specialists Prescribing Opioids', alpha= alpha_,
beta= rate_, observed = opi_presc)
    trace= pm.sample(1000, tune=2000, cores=2)

Alpha_chi = [0.40736891 0.42284083]
Rate_ci = [0.00019885 0.00021156]
```

Fig 12. and **Fig 13.** Show our credible intervals

Fig 12.

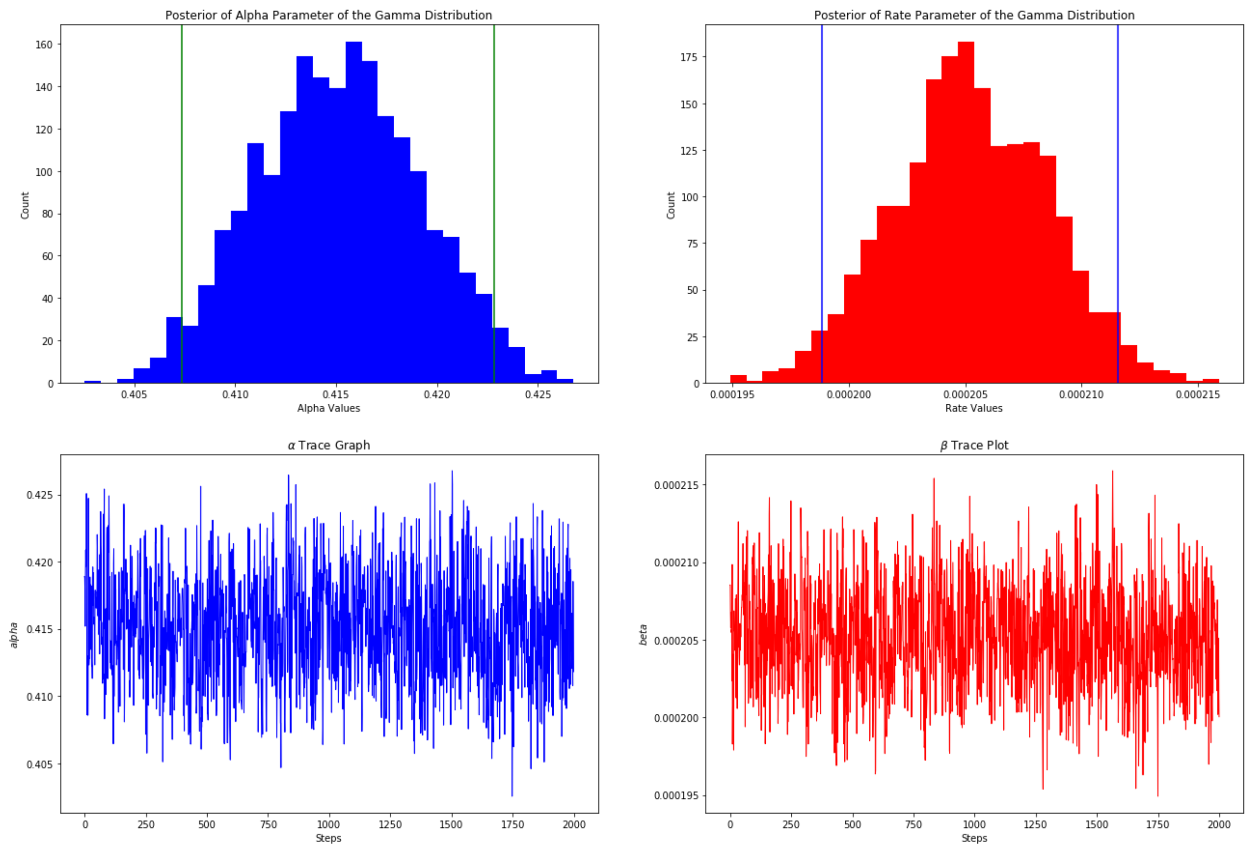
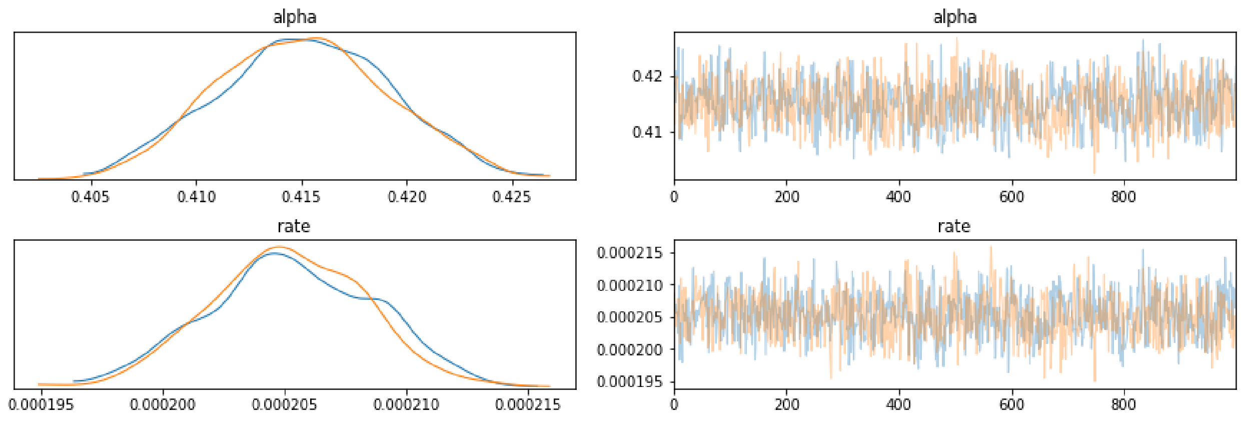


Fig 13.



p. Summary

We have postulated a distribution to describe the individual amounts for opioid prescriptions cases. This distribution has two required parameters, which we do not know, but we used PyMC3 to perform Bayesian inference to find our level of "belief" in a range of values for them. We then used the average parameter values to create one simulated data set of the same size as the original, but the distribution of our posteriors for these parameters will allow us to perform simulations of any sample size we desire and for a range of scenarios of different a and b.