FinalProject520YoungDeborah

Deborah Young

2023-02-28

Milestone 1:

You will be working on a research paper for your final project. This project will include identifying a topic/problem that you want to solve using data science. While the final solution to the problem does not need to be provided via programming – you will be doing some exploratory data analysis, transformations, and summary statistics on the data via R. You are welcome to create a model based on what you have learned in this course to solve the problem, but this is not required. Instead, a recommendation is required for a model or method you would implement to solve the problem. There are 3 steps to this project, with the final deliverable being a formal paper (completed in R Markdown) that outlines the problem, shows the analysis done with the data, and concludes with your recommendation for next steps. Each step provides detailed information that you must include in each phase of the project.

- *Identify a topic or a problem that you want to research. Provide an introduction that explains the problem statement or topic you are addressing. Why would someone be interested in this? How is it a data science problem?*

I'm choosing to use data science via machine learning models to help determine hotspots of opioid prescription and overdoses that would benefit from more opioid-care facilities or opioid-related education.

According to the CDC:

- "Millions of people in the U.S. are living with opioid use disorder (OUD).
- Opioid use disorder (OUD) occurs when opioid use causes significant impairment and distress.
- A diagnosis of OUD is based on specific criteria such as unsuccessful efforts to cut down or control use or use resulting in a failure to fulfill obligations at work, school, or home, among other criteria.
- About 2.7 million people in the United States report suffering from OUD. Overdoses are a leading injury-related cause of death in the United States and appear to have accelerated during the COVID-19 pandemic.
- OUD is a medical condition that can affect anyone – regardless of race, gender, income level, or social class.
- Common treatment options for OUD include medications for opioid use disorder (MOUD) (including methadone, buprenorphine, naltrexone)."

- *Draft 5-10 Research questions that focus on the problem statement/topic.*

- Which opioid medications are most frequently prescribed?
- Which medication prescriptions occur adjacent to opioids?
- What providers/specialties are prescribing opioids?

- Where the providers located that are making the most prescriptions?
- Where are the highest rates of opioid-related overdoses?

- *Provide a concise explanation of how you plan to address this problem statement.*

Using publicly available data about prescriptions and overdoses, I can manipulate the data to find correlations between rates of prescriptions and overdoses and build a model that locates where the hotspots are.

- *Discuss how your proposed approach will address (fully or partially) this problem.*

Although I'm not positive that I have the skills (nor most recent data) to develop a model that could actually be implemented, or if my correlations will be meaningful, gaining the experience in finding relationships between data like this is very beneficial for being able to recognize ways that machine learning can inform public health practices.

- *Do some digging and find at least 3 datasets that you can use to address the issue. (There is not a required number of fields or rows for these datasets)*

o Original source where the data was obtained is cited and, if possible, hyperlinked.

I will be using subsets of data from Centers for Medicare & Medicaid Service provided by Alan Pryor via Kaggle.

o Source data is thoroughly explained (i.e. what was the original purpose of the data, when was it collected, how many variables did the original have, explain any peculiarities of the source data such as how missing values are recorded, or how data was imputed, etc.).

According to the author, Alan Pryor, the original dataset contains "summaries of prescription records for 250 common opioid and non-opioid drugs written by 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 a small subset of data that was sourced from cms.gov. The full dataset contains almost 24 million prescription instances in long format. I have cleaned and compiled this data 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. If you are interested in more data, you can get the script I used to assemble the dataset here and run it yourself. The main data is in `prescriber-info.csv`. There is also `opioids.csv` that contains the names of all opioid drugs included in the data and `overdoses.csv` that contains information on opioid related drug overdose fatalities.

I will be using the aforementioned subsets: `prescriber-info.csv`, `opioids.csv,andoverdoses.csv.`

I will also include here the disclaimer provided by the author: "I am absolutely not suggesting that doctors who prescribe opiates are culpable for overdoses. These are drugs with true medical value when used appropriately. The idea is rather that a systematic way for identifying sources may reveal trends in particular practices, fields, or regions of the country that could be used effectively to combat the problem."

- *Identify the packages that are needed for your project.*

Although I'm sure I'll add more packages throughout, I will likely be using some of the following: "ggplot2", "dplyr", "caTools", "hmisc", "ggm", "tidyverse", "QuantPsyc", and/or "car".

- *What types of plots and tables will help you to illustrate the findings to your research questions?*

Scatter plots, histograms, tibbles, bar charts, and/or confusion matrices may be beneficial throughout this process.

- *What do you not know how to do right now that you need to learn to answer your research questions?*

I will need to figure out how I want to mutate the data, possibly using dplyr. I will want to use plots to gain perspective on potential relationships. I need to figure out what correlations I can calculate and what their significance (if any) will indicate. I want to use k-nearest-neighbors and k-means algorithm to build a model to visualize hotspots, but I'm not sure if I'm talented enough for that yet. :)

Milestone 2:

At this point you should have framed your problem/topic, described the data, and how you plan to solve the problem. Now you need to move on to the next step of analyzing and preparing the data.

Some additional questions you may want to consider asking yourself as you work through this section of the project: 1. What features could you filter on? 2. How could arranging your data in different ways help? 3. Can you reduce your data by selecting only certain variables? 4. Could creating new variables add new insights? 5. Could summary statistics at different categorical levels tell you more? 6. How can you incorporate the pipe (%>%) operator to make your code more efficient?

Adding on to the draft you started in Step 1:

- *Data importing and cleaning steps are explained in the text and follow a logical process. Outline your data preparation and cleansing steps.*

First I will import and review the datasets.

```
provider_df <- read.csv("/Users/debane/Documents/MS Data Science/520 Statistics for DS/Final Project/pr

# check provider df for na values and make new df
nrow(provider_df)
```

```
## [1] 25000
```

```
ncol(provider_df)
```

```
## [1] 256
```

```
provider <- na.omit(provider_df)
nrow(provider)
```

```
## [1] 25000
```

```
ncol(provider)
```

```
## [1] 256
```

```
opioids_df<- read.csv("/Users/debane/Documents/MS Data Science/520 Statistics for DS/Final Project/opio

# check opioids df for na values
nrow(opioids_df)
```

```
## [1] 113
```

```
ncol(opioids_df)
```

```
## [1] 2
```

```
opioids <- na.omit(opioids_df)
nrow(opioids)
```

```
## [1] 113
```

```
ncol(opioids)
```

```
## [1] 2
```

```
od_df <- read.csv("/Users/debane/Documents/MS Data Science/520 Statistics for DS/Final Project/overdoses

# check od for na values
nrow(od_df)
```

```
## [1] 50
```

```
ncol(od_df)
```

```
## [1] 4
```

```
overdose <- na.omit(od_df)
nrow(overdose)
```

```
## [1] 50
```

```
ncol(overdose)
```

```
## [1] 4
```

```
colnames(provider)
```

I want to only keep the providers who are opioid prescribers, so I am filtering for rows that have a true value in the opioid prescriber column. Then I'll follow Alan Pryor (*and I will be returning to his code when needed throughout this project*) in converting the character columns to factors. I also want to edit the Opioid Dataset so that the names of the drugs are the same as they are listed in the Provider dataset so that I can compare them later. Then I'll review the columns with information regarding the providers.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(stringr)

#change provider dataframe name for simplicity
df <- provider

#Convert character columns to factors.
char_cols <- c("NPI",names(df)[vapply(df,is.character,TRUE)])
df[,char_cols] <- lapply(df[,char_cols],as.factor)

# First column contains the names of the opiates
opioids <- as.character(opioids[,1])

# replace hyphens and spaces with periods to match the dataset
opioids <- gsub("\ |-",".",opioids)
opioids
```

```
##  [1] "ABSTRAL"                        "ACETAMINOPHEN.CODEINE"
##  [3] "ACTIQ"                          "ASCOMP.WITH.CODEINE"
##  [5] "ASPIRIN.CAFFEINE.DIHYDROCODEIN" "AVINZA"
##  [7] "BELLADONNA.OPIUM"               "BUPRENORPHINE.HCL"
##  [9] "BUTALB.ACETAMINOPH.CAFF.CODEIN" "BUTALB.CAFF.ACETAMINOPH.CODEIN"
## [11] "BUTALBITAL.COMPOUND.CODEINE"    "BUTORPHANOL.TARTRATE"
## [13] "BUTRANS"                        "CAPITAL.W.CODEINE"
## [15] "CARISOPRODOL.COMPOUND.CODEINE"  "CARISOPRODOL.ASPIRIN.CODEINE"
## [17] "CODEINE.SULFATE"                "CO.GESIC"
## [19] "CONZIP"                         "DEMEROL"
## [21] "DEMEROL"                        "DILAUDID"
## [23] "DILAUDID"                       "DILAUDID.HP"
## [25] "DISKETS"                        "DOLOPHINE.HCL"
## [27] "DURAGESIC"                      "DURAMORPH"
## [29] "ENDOCET"                        "ENDODAN"
## [31] "EXALGO"                         "FENTANYL"
## [33] "FENTANYL.CITRATE"               "FENTORA"
## [35] "FIORICET.WITH.CODEINE"          "FIORINAL.WITH.CODEINE.#3"
## [37] "HYCET"                          "HYDROCODONE.ACETAMINOPHEN"
## [39] "HYDROCODONE.IBUPROFEN"          "HYDROMORPHONE.ER"
## [41] "HYDROMORPHONE.HCL"              "HYDROMORPHONE.HCL"
## [43] "IBUDONE"                        "INFUMORPH"
## [45] "KADIAN"                         "LAZANDA"
## [47] "LEVORPHANOL.TARTRATE"           "LORCET"
## [49] "LORCET.10.650"                  "LORCET.HD"
## [51] "LORCET.PLUS"                    "LORTAB"
## [53] "MAGNACET"                       "MEPERIDINE.HCL"
## [55] "MEPERIDINE.HCL"                 "MEPERITAB"
## [57] "METHADONE.HCL"                  "METHADONE.INTENSOL"
## [59] "METHADOSE"                      "MORPHINE.SULFATE"
## [61] "MORPHINE.SULFATE"               "MORPHINE.SULFATE.ER"
## [63] "MS.CONTIN"                      "NALBUPHINE.HCL"
## [65] "NORCO"                          "NUCYNTA"
## [67] "NUCYNTA.ER"                     "OPANA"
## [69] "OPANA.ER"                       "OPIUM.TINCTURE"
## [71] "OXECTA"                         "OXYCODONE.HCL"
## [73] "OXYCODONE.HCL.ER"               "OXYCODONE.HCL.ASPIRIN"
##  [75] "OXYCODONE.HCL.IBUPROFEN"        "OXYCODONE.ACETAMINOPHEN"
##  [77] "OXYCONTIN"                      "OXYMORPHONE.HCL"
##  [79] "OXYMORPHONE.HCL.ER"             "PENTAZOCINE.ACETAMINOPHEN"
##  [81] "PENTAZOCINE.NALOXONE.HCL"       "PERCOCET"
##  [83] "PERCODAN"                       "PRIMLEV"
##  [85] "REPREXAIN"                      "ROXICET"
##  [87] "ROXICODONE"                     "RYBIX.ODT"
##  [89] "STAGESIC"                       "SUBSYS"
##  [91] "SYNALGOS.DC"                    "TALWIN"
##  [93] "TRAMADOL.HCL"                   "TRAMADOL.HCL.ER"
##  [95] "TRAMADOL.HCL.ACETAMINOPHEN"     "TREZIX"
##  [97] "TYLENOL.CODEINE.NO.3"           "TYLENOL.CODEINE.NO.4"
##  [99] "ULTRACET"                       "ULTRAM"
## [101] "ULTRAM.ER"                      "VICODIN"
## [103] "VICODIN.ES"                     "VICODIN.HP"
## [105] "VICOPROFEN"                     "XARTEMIS.XR"
## [107] "XODOL.10.300"                   "XODOL.5.300"
## [109] "XODOL.7.5.300"                  "XYLON.10"
## [111] "ZAMICET"                        "ZOHYDRO.ER"
## [113] "ZOLVIT"
```

```
#View information for the first few columns with the provider information
str(df[,1:5])
```

```
## ’data.frame’:    25000 obs. of  5 variables:
##  $ NPI        : Factor w/ 25000 levels "1003002320","1003004771",..: 18016 6106 10728 16695 16924 138
##  $ Gender     : Factor w/ 2 levels "F","M": 2 1 1 2 2 2 2 1 2 1 ...
##  $ State      : Factor w/ 57 levels "AA","AE","AK",..: 48 4 38 6 37 42 34 42 48 54 ...
##  $ Credentials: Factor w/ 888 levels "","(DMD)","A.N.P.",..: 244 507 403 507 403 314 507 839 697 507
##  $ Specialty  : Factor w/ 109 levels "Addiction Medicine",..: 19 29 28 42 36 29 25 63 66 42 ...
```

There are too many states (56) listed, so I need to verify them.

```
df %>%
  group_by(State) %>%
  dplyr::summarise(state.counts = n()) %>%
  arrange(state.counts)
```

```
## # A tibble: 57 x 2
##    State state.counts
##    <fct>        <int>
##  1 AA               1
##  2 AE               2
##  3 GU               2
##  4 ZZ               2
##  5 VI               3
##  6 WY              38
##  7 AK              39
##  8 VT              65
##  9 ND              66
## 10 MT              77
## # ... with 47 more rows
```

We can review the state abbreviations on this site. I can create an "other" variable for the states that don't occur frequently.

```
#Select states that have infrequent occurences
rare.abbrev <- df %>%
  group_by(State) %>%
  dplyr::summarise(state.counts = n()) %>%
  arrange(state.counts) %>%
  filter(state.counts < 10) %>%
  select(State)

# Insert a new level into the factor, then remove the old names
levels(df$State) <- c(levels(df$State),"other")
df$State[df$State %in% rare.abbrev$State] <- "other"
df$State <- droplevels(df$State)
```

Reviewed credentials, but since they are all over the place, I'm removing them.

```
#Select credentials by occurrence

df %>%
  group_by(Credentials) %>%
  dplyr::summarise(credential.counts = n()) %>%
  arrange(credential.counts) %>%
  data.frame() %>%
  glimpse()

#Remove credentials column
df2 <- select(df, -Credentials)
df <- df2
```

Now I want to look at the specialties and edit them.

```
#select specialty and sort by specialty count
df %>%
  group_by(Specialty) %>%
  dplyr::summarise(specialty.counts = n()) %>%
  arrange(desc(specialty.counts)) %>%
  data.frame() %>%
  glimpse()
```

```
## Rows: 109
## Columns: 2
## $ Specialty        <fct> "Internal Medicine", "Family Practice", "Dentist", "N~
## $ specialty.counts <int> 3194, 2975, 2800, 2512, 1839, 1087, 691, 688, 615, 57~
```

The specialties have a lot of common occurrences, so I want to keep those. Then the specialties with uncommon names can be re-categorized. I will pull out any specialties with the word fragment "surg" and call it "Surgeon", "pain" and call it "Pain.Management", and create an "Other" variable as well. Then we can view the categories.

```r
#group by specialty, save common specialities and work on the remaining
common.specialties <- df %>%
  group_by(Specialty) %>%
  dplyr::summarise(specialty.counts = n()) %>%
  arrange(desc(specialty.counts)) %>%
  filter(specialty.counts > 50) %>%
  select(Specialty)
common.specialties <- levels(droplevels(common.specialties$Specialty))

#create variables for uncommon specialties
new.specialties <- factor(x=rep("other",nrow(df)),levels=c(common.specialties,"Surgeon","other","Pain.Ma
new.specialties[df$Specialty %in% common.specialties] <- df$Specialty[df$Specialty %in% common.specialti
new.specialties[grepl("surg",df$Specialty,ignore.case=TRUE)] <- "Surgeon"
new.specialties[grepl("pain",df$Specialty,ignore.case=TRUE)] <- "Pain.Management"
new.specialties <- droplevels(new.specialties)
df$Specialty <- new.specialties

#view specialites
df %>%
  group_by(Specialty) %>%
  dplyr::summarise(specialty.counts = n()) %>%
  arrange(desc(specialty.counts)) %>%
  data.frame() %>%
  glimpse()
```

```
## Rows: 39
## Columns: 2
## $ Specialty        <fct> Internal Medicine, Family Practice, Dentist, Nurse Pr~
## $ specialty.counts <int> 3194, 2975, 2800, 2512, 1839, 1689, 1087, 691, 688, 6~
```

Finally, because this is a subset of a larger dataset, some of the prescriptions there are some medications that won't have prescribed values, so I want to remove those variables. This is also helpful for avoiding superfluous multicollinearity.

```r
#use vapply to remove occurences of 0
df <- df[vapply(df,function(x) if (is.numeric(x)){sum(x)>0}else{TRUE},FUN.VALUE=TRUE)]
```

- *With a clean dataset, show what the final data set looks like. However, do not print off a data frame with 200+ rows; show me the data in the most condensed form possible.*

We can compare the information from the beginning of this exploration to the current layout.

Original:

‘data.frame’: 14688 obs. of 5 variables: $ NPI : Factor w/ 14688 levels “1003002320”,“1003008475”,..: 10595 3607 9805 9943 8130 6454 12172 8954 9562 2104 … $ Gender : Factor w/ 2 levels “F”,“M”: 2 1 2 2 2 2 1 1 2 2 … $ State : Factor w/ 56 levels “AE”,“AK”,“AL”,..: 47 3 5 36 41 33 53 42 47 7 … $ Credentials: Factor w/ 526 levels ““,”A.N.P.”,“A.P.N.”,..: 154 309 309 253 200 309 309 253 309 309 … $ Specialty : Factor w/ 94 levels “Addiction Medicine”,..: 16 26 37 33 26 22 37 25 26 22 …

```r
str(df[,1:4])
```

```
## 'data.frame':    25000 obs. of  4 variables:
```

```
##  $ NPI      : Factor w/ 25000 levels "1003002320","1003004771",..: 18016 6106 10728 16695 16924 13862
##  $ Gender   : Factor w/ 2 levels "F","M": 2 1 1 2 2 2 2 1 2 1 ...
##  $ State    : Factor w/ 53 levels "AK","AL","AR",..: 45 2 35 4 34 39 31 39 45 50 ...
##  $ Specialty: Factor w/ 39 levels "Allergy/Immunology",..: 5 37 11 15 13 37 9 20 23 15 ...
```

- *What do you not know how to do right now that you need to learn to import and cleanup your dataset?*

I need to see if I want to filter on other variables or use other data transformation tools (dplyr). I will need to change some of the named variables into categorical variables for scoring. I tried making dummy variables but haven't quite figured it out. I am also wondering if I should create an alternate dataframe by removing all medications that are not opioids for further exploration.

- *Discuss how you plan to uncover new information in the data that is not self-evident. What are different ways you could look at this data to answer the questions you want to answer?*

I could find correlations between specialties and prescriptions, location and prescriptions, locations and specialties, other drug prescriptions to inform likelihood of opioid prescription.

- *Do you plan to slice and dice the data in different ways, create new variables, or join separate data frames to create new summary information? Explain.*

.I'm considering pulling out only the opioid drugs and working on that dataset. I might also be able to merge the overdoses and providers datasets to work on correlations from there. Packages I could use are "dplyr".

- *How could you summarize your data to answer key questions?*

I'll use summary statistics via packages to summarize ("car" "Hmisc", "pastecs", "quantpsyc").

- *What types of plots and tables will help you to illustrate the findings to your questions? Ensure that all graph plots have axis titles, legend if necessary, scales are appropriate, appropriate geoms used, etc.).*

I could potentially use linear or logistic regressions, scatterplots/histograms/bar charts (plot(), hist(), ggplot2, glm(), lm(), ggm).

Do you plan on incorporating any machine learning techniques to answer your research questions? Explain.

I am hoping to use k-nearest neighbors and/or kmeans clusters to see where the overlap of providers and overdoses are located. (Clutsters, ClusterR, caTools). We will see!

- *What do you not know how to do right now that you need to learn to answer your questions?*

I want to use machine learning so I will have to figure out what I want to use and how to complete it!

Milestone 3:

You are now on to the final phase of your research paper. While this step does not require you build a model, you are welcome to do so if you feel you have the time. Instead, you need to make a recommendation for the approach you would take and what the remaining steps would be using the information you have learned in this course to take this project from simply being an analysis exercise to proposed implementation of a solution.

- *Overall, write a coherent narrative that tells a story with the data as you complete this section.*

I'm going to start by making a new dataframe containing only the opioid medications. I can do this because the Opioid dataset contains all of the names of the opioid drugs. I've cleaned it up in the second step of this project so that the names are formatted in the same way as the Provider (df) dataset. I took way too long trying to figure this out because I'm not skilled enough with data transformation and somehow kept doing something wrong. Please excuse my weird process below.

From here, I'm going to abandon my original hope to use k-means clusters (I think that was a bit ambitious of me to want to do with these datasets) and focus on a regression model to predict opioid prescription…

Dataset reconfiuration:

```
opioids_df2 <- as.data.frame(opioids)

provider_notopioids <- df %>% select(-any_of(dput(as.character(opioids))))

notopioids <- select(provider_notopioids, -c('NPI', 'Gender', 'State', 'Specialty', 'Opioid.Prescriber'

notopioids_df <- as.data.frame(notopioids)

colnames <- colnames(notopioids_df)
colnames_df <- as.data.frame(colnames)

not_opioids <- as.character(colnames_df[,1])

df3 <- df

#New dataframe!!!
provider_opioidsonly <- df3 %>% select(-any_of(dput(as.character(not_opioids))))
```

Next I'm going to use my code from the week 10 assignment to build models and predict Opioid Prescription & check accuracy.

I am using poisson because many of the values are counts… not sure if this is correct but I'm going for it.

```
#NPI doesn't have a lot of added value here, so I'm going to remove it from the dataframe

colnames(df)

df_mod <- select(df, -NPI)
```

```
# Split the data
library(caTools)

set.seed(250)  #makes replicatable

split <- sample.split(df_mod, SplitRatio = .8)
train <- subset(df_mod, split == "TRUE")
test <- subset(df_mod, split == "FALSE")

# Train the data

prescribe.mod.1 <- glm(Opioid.Prescriber ~ ., data=train, family = poisson(),
                       na.action = na.omit)
# summary(prescribe.mod.1)
```

```
# Run test through the model

res <- predict(prescribe.mod.1, test, type="response")
res <- predict(prescribe.mod.1, train, type ="response")

# Validate the model

confmatrix.1 <- table(Actual_Value=train$Opioid.Prescriber,
                      Predicted_Value = res >0.5)
confmatrix.1
```

```
##              Predicted_Value
## Actual_Value FALSE TRUE
##            0  6001 2252
##            1  2130 9595
```

```
# Calculate accuracy

(confmatrix.1[[1,1]] + confmatrix.1[[2,2]]) / sum(confmatrix.1)
```

```
## [1] 0.7806587
```

With this model, we're reporting 78% accuracy. That's not too bad, but not great. As we may have guessed, the specialties have a strong correlation with the prediction of opioid prescription (**this was identified in the giant Summary - not printed here for the sake of the reader**). I'm going to try removing some variables and see if that changes the accuracy.

```
set.seed(250) #makes replicatable

split <- sample.split(df_mod, SplitRatio = .8)
train <- subset(df_mod, split == "TRUE")
test <- subset(df_mod, split == "FALSE")

# Train the data

prescribe.mod.2 <- glm(Opioid.Prescriber ~ . -Gender, data=train,
                       family = poisson(), na.action = na.omit)
# summary(prescribe.mod.2)
```

```
# Run test through the model

res <- predict(prescribe.mod.2, test, type="response")
res <- predict(prescribe.mod.2, train, type ="response")

# Validate the model

confmatrix.2 <- table(Actual_Value=train$Opioid.Prescriber,
                      Predicted_Value = res >0.5)
confmatrix.2
```

```
##              Predicted_Value
## Actual_Value FALSE TRUE
##            0  6000 2253
##            1  2116 9609
```

```
# Calculate accuracy

(confmatrix.2[[1,1]] + confmatrix.2[[2,2]]) / sum(confmatrix.2)
```

```
## [1] 0.7813094
```

Removing the gender makes it slightly more accurate, but not significantly.

I'm going to try again, this time removing state.

```
set.seed(250) #makes replicatable

split <- sample.split(df_mod, SplitRatio = .8)
train <- subset(df_mod, split == "TRUE")
test <- subset(df_mod, split == "FALSE")

# Train the data, using poisson because many of the values are low number integers

prescribe.mod.3 <- glm(Opioid.Prescriber ~ . -State, data=train,
                       family = poisson(), na.action = na.omit)
# summary(prescribe.mod.1)
```

```
# Run test through the model

res <- predict(prescribe.mod.3, test, type="response")
res <- predict(prescribe.mod.3, train, type ="response")

# Validate the model

confmatrix.3 <- table(Actual_Value=train$Opioid.Prescriber,
                      Predicted_Value = res >0.5)
confmatrix.3
```

```
##              Predicted_Value
## Actual_Value FALSE TRUE
##            0  5685 2568
##            1  1939 9786
```

```
# Calculate accuracy

(confmatrix.3[[1,1]] + confmatrix.3[[2,2]]) / sum(confmatrix.3)
```

```
## [1] 0.7744018
```

Ok so removing state makes it less accurate, which makes sense.

I'm going to try without using the prescriptions.

```
set.seed(250) #makes replicatable

split <- sample.split(df_mod, SplitRatio = .8)
train <- subset(df_mod, split == "TRUE")
test <- subset(df_mod, split == "FALSE")

# Train the data

prescribe.mod.4 <- glm(Opioid.Prescriber ~ Gender + State + Specialty,
                       data=train, family = poisson(), na.action = na.omit)
# summary(prescribe.mod.1)
# Run test through the model

res <- predict(prescribe.mod.4, test, type="response")
res <- predict(prescribe.mod.4, train, type ="response")

# Validate the model

confmatrix.4 <- table(Actual_Value=train$Opioid.Prescriber,
                      Predicted_Value = res >0.5)
confmatrix.4
```

```
##              Predicted_Value
## Actual_Value FALSE TRUE
##            0  5492 2761
##            1  2072 9653
```

```
# Calculate accuracy

(confmatrix.4[[1,1]] + confmatrix.4[[2,2]]) / sum(confmatrix.4)
```

```
## [1] 0.7580839
```

Now it's even less accurate. Alright, good to know!

Now I'm going to try a model with my datasets that contain only non-opioid prescriptions.

```
nonopioid_mod_df <- select(provider_notopioids, -NPI)

set.seed(250)  #makes replicatable

split <- sample.split(nonopioid_mod_df, SplitRatio = .8)
train <- subset(df_mod, split == "TRUE")
test <- subset(df_mod, split == "FALSE")

# Train the data

nonopioid.prescribe.mod.1 <- glm(Opioid.Prescriber ~ ., data=train,
                                 family = poisson(), na.action = na.omit)
# summary(prescribe.mod.1)
```

```
# Run test through the model

res <- predict(nonopioid.prescribe.mod.1, test, type="response")
res <- predict(nonopioid.prescribe.mod.1, train, type ="response")

# Validate the model

confmatrix.6 <- table(Actual_Value=train$Opioid.Prescriber,
                      Predicted_Value = res >0.5)
confmatrix.6
```

```
##              Predicted_Value
## Actual_Value FALSE TRUE
##            0  5963 2276
##            1  2150 9570
```

```
# Calculate accuracy

(confmatrix.6[[1,1]] + confmatrix.6[[2,2]]) / sum(confmatrix.6)
```

```
## [1] 0.7782454
```

Still about 78% accurate.

- *Summarize the problem statement you addressed.*

- *Summarize how you addressed this problem statement (the data used and the methodology employed, including a recommendation for a model that could be implemented).*

- *Summarize the interesting insights that your analysis provided.*

- *Summarize the implications to the consumer (target audience) of your analysis.*

- *Discuss the limitations of your analysis and how you, or someone else, could improve or build on it.*

Well… I didn't address the problem I set out to address. That is a big lesson for me. Not having the robust understanding that I thought I did is a bummer, but it is encouraging to me to continue learning and growing! The model I built instead could hopefully address a different question (that I came up with along the way).

Question: Can we predict the likelihood of an opioid prescriber based on their specialty, other medications prescribed, location, etc? **Yes, we probably can!**

Did I do that here? I'm not so confident about that. The generalized linear models (glm) I created might be completely wrong, but the calculations showed that we could be 78% accurate using this set of variables to predict opioid prescription by a provider.

This information could be used to inform best practices around opioid care, particularly prevention, and hopefully also encourage providers and communities to promte greater awareness and access to opioid-care facilities and treatments (eg. methadone, buprenorphine, naltrexone).

Because it took so long for me to clean and transform my data, I got to a point where I recognized that I would not be able to complete all of the functions I wanted to with precision.

Ideally, these are some of the other things I would do:

- Find correlations between opioid prescriptions and other prescriptions Find correlations between specialties and opioid prescriptions Find correlations between locations and opioid prescriptions Find overlaps between locations, opioid prescriptions, and specialties

- Plot each prescription based on its frequency Plot each opioid prescription based on its frequency Plot prescriptions of non-opioids against opioids

- Many other plots and summary statistics

What I really wanted to do was to use k-means!!! But, I *kmean*, maybe I don't really have the skills to get my data to the point that I could do that. Maybe I will someday!

I'm really encouraged by this class. I've enjoyed the statistical methods so much more than I thought (which is encouraging for possibly furthering my public health career) and I gained an immense amount of confidence in using R, which translates into an easier time using Python as well.

Thank you, Professor Denton, for an excellent course!