Opioid-Related Death in Massachusetts

Simulated Demo

Gabby Novak

```
#### Packages ####
library(tidyverse)
# Loads: ggplot2, dplyr, tidyr, readr, purrr, tibble, stringr, & forcats
library(tigris)
library(lubridate)
library(knitr)
library(kableExtra)
library(grid)
library(gridExtra)
library(extrafont)
library(viridis)

#### External Data ####
# base shp file for mapping
shp<-zctas(year=2010,state="Massachusetts")
# Example occupations and industries
occups<-read_csv("Occups.csv")
# ICD10 2018 codes
# Source: https://www.cms.gov/Medicare/Coding/ICD10/2018-ICD-10-CM-and-GEMs.html
icd10<-read_csv("icd10cm_order_2018.csv")
```

Data Set Unification

This project made use of publicly available death records of individuals who died in Massachusetts, USA between 2000 and 2017 with an opioid-related ICD10 code assigned to them as a cause of death. The data source presented several challenges, not least among them, errors due to manual entry of information. However, the greatest hurdle were changes to data format in mid-2014. Prior to that 2014 format change, 77 variables were available. After the format change, a staggering 843 were available. While the vast majority of the earlier variables were repeated in the newer format, both variable names and coding structures were updated. In order to fit any sort of temporal model, we needed a unified data set. Additionally, while we were interested in several individual-level covariates, much of these data were irrelevant to us. I developed the following functions as a mechanism to extract specific information from the raw vitals, recode it, and populate a data frame much more suited to the project's needs. Note that these functions create coded data sets. The full data included several hundred thousand observations. The coding was a mechanism meant to reduce file size for sharing between colleagues. The analysis was actually completed using data run through another function which converted the numeric labels to their representative values.

```
#### For deaths 2000-mid 2014 ####
vital.00.14<-function(dataset){
  temp<-list()
  state<-c("ALABAMA","ALASKA","ARIZONA","ARKANSAS","CALIFORNIA","COLORADO",
           "CONNECTICUT","DELAWARE","FLORIDA","GEORGIA","HAWAII","IDAHO",
           "ILLINOIS","INDIANA","IOWA","KANSAS","KENTUCKY","LOUISIANA",
           "MAINE","MARYLAND","MASSACHUSETTS","MICHIGAN","MINNESOTA",
```

```
          "MISSISSIPPI","MISSOURI","MONTANA","NEBRASKA","NEVADA",
          "NEW HAMPSHIRE","NEW JERSEY","NEW MEXICO","NEW YORK",
          "NORTH CAROLINA","NORTH DAKOTA","OHIO","OKLAHOMA","OREGON",
          "PENNSYLVANIA","RHODE ISLAND","SOUTH CAROLINA","SOUTH DAKOTA",
          "TENNESSEE","TEXAS","UTAH","VERMONT","VIRGINIA","WASHINGTON",
          "WASHINGTON DC","WEST VIRGINIA","WISCONSIN","WYOMING")
abbr<-c("AL","AK","AZ","AR","CA","CO","CT","DC","DE","FL","GA","HI","ID",
        "IL","IN","IA","KS","KY","LA","ME","MD","MA","MI","MN","MS","MO",
        "MT","NE","NV","NH","NJ","NM","NY","NC","ND","OH","OK","OR","PA","RI",
        "SC","SD","TN","TX","UT","VT","VA","WA","WV","WI","WY")
# batch
temp$batch<-1
for(j in 1:nrow(dataset)){
  # sfnum
  temp$sfnum[j]<-unlist(dataset[j,"CERT"])
  # ddate
  temp$ddate[j]<-paste0(str_sub(dataset[j,"DOD"],1,4),"-",
                                str_sub(dataset[j,"DOD"],5,6),"-",
                                str_sub(dataset[j,"DOD"],7,8))
  # male
  if(dataset[j,"SEX"]=="1"){temp$male[j]<-1}
  if(dataset[j,"SEX"]=="2"){temp$male[j]<-0}
  # age
  if(str_sub(dataset[j,"AGE_AT_DEATH"],1,1)==0|
     str_sub(dataset[j,"AGE_AT_DEATH"],1,1)==1)
  {temp$age[j]<-as.numeric(str_sub(dataset[j,"AGE_AT_DEATH"],-2,-1))}
  else{if(str_sub(dataset[j,"AGE_AT_DEATH"],1,1)==2|
          str_sub(dataset[j,"AGE_AT_DEATH"],1,1)==4|
          str_sub(dataset[j,"AGE_AT_DEATH"],1,1)==5|
          str_sub(dataset[j,"AGE_AT_DEATH"],1,1)==6)
    {temp$age[j]<-0}else{if(str_sub(dataset[j,"AGE_AT_DEATH"],1,1)==9){temp$age[j]<-NA}}}
  # race
  if(!(dataset[j,"DETHNIC_HISPANIC"]=="0"|dataset[j,"DETHNIC_HISPANIC"]=="9"))
    {temp$race[j]<-3}
    else{if(dataset[j,"RACE"]=="01"){temp$race[j]<-1}
         if(dataset[j,"RACE"]=="02"){temp$race[j]<-2}
         if(dataset[j,"RACE"]=="03"){temp$race[j]<-5}
         if(dataset[j,"RACE"]=="04"|
            dataset[j,"RACE"]=="05"|
            dataset[j,"RACE"]=="06"|
            dataset[j,"RACE"]=="07"|
            dataset[j,"RACE"]=="08"|
            dataset[j,"RACE"]=="09"|
            dataset[j,"RACE"]=="10"|
            dataset[j,"RACE"]=="11"|
            dataset[j,"RACE"]=="12"){temp$race[j]<-4}
         if(dataset[j,"RACE"]=="13"|
            dataset[j,"RACE"]=="14"){temp$race[j]<-7}
         if(dataset[j,"RACE"]=="99"){temp$race[j]<-NA}}
  # occup
  temp$occup[j]<-unlist(dataset[j,"OCCUP"])
  # indust
  temp$indust[j]<-unlist(dataset[j,"INDUST"])
```

```
# edu
if(as.numeric(dataset[j,"DEDUC"])<=11){temp$edu[j]<-1}
  else{if(as.numeric(dataset[j,"DEDUC"])<=13){temp$edu[j]<-2}
    else{if(as.numeric(dataset[j,"DEDUC"])<=16){temp$edu[j]<-3}
      else{if(dataset[j,"DEDUC"]=="99"){temp$edu[j]<-NA}
        else{if(as.numeric(dataset[j,"DEDUC"])>16){temp$edu[j]<-4}}}}}
# immig
if(dataset[j,"NATIVITY"]=="99"){temp$immig[j]<-NA}
  else{if(as.numeric(dataset[j,"NATIVITY"])>51){temp$immig[j]<-4}
    else{temp$immig[j]<-5}}
# pimmig
ifelse(!dataset[j,"FATHER_BSTATE"]%in%state&
       !dataset[j,"FATHER_BSTATE"]%in%abbr&
       !dataset[j,"FATHER_BSTATE"]=="UNKNOWN",
       yes=ifelse(!dataset[j,"MOTHER_BSTATE"]%in%state&
                  !dataset[j,"MOTHER_BSTATE"]%in%abbr&
                  !dataset[j,"MOTHER_BSTATE"]=="UNKNOWN",
                  yes=temp$pimmig[j]<-2,
                  no=temp$pimmig[j]<-1),
       no=ifelse(!dataset[j,"MOTHER_BSTATE"]%in%state&
                 !dataset[j,"MOTHER_BSTATE"]%in%abbr&
                 !dataset[j,"MOTHER_BSTATE"]=="UNKNOWN",
                 yes=temp$pimmig[j]<-1,
                 no=ifelse((dataset[j,"FATHER_BSTATE"]%in%state|
                     dataset[j,"FATHER_BSTATE"]%in%abbr)&
                    (dataset[j,"MOTHER_BSTATE"]%in%state|
                     dataset[j,"MOTHER_BSTATE"]%in%abbr),
                    yes=temp$pimmig[j]<-0,
                    no=temp$pimmig[j]<-NA)))
# marital
if(dataset[j,"MARITAL"]=="1"){temp$marital[j]<-5}
if(dataset[j,"MARITAL"]=="2"){temp$marital[j]<-1}
if(dataset[j,"MARITAL"]=="3"){temp$marital[j]<-3}
if(dataset[j,"MARITAL"]=="4"){temp$marital[j]<-4}
if(dataset[j,"MARITAL"]=="9"){temp$marital[j]<-NA}
# veteran
if(dataset[j,"VET_STAT"]==0){temp$veteran[j]<-0}
else{if(dataset[j,"VET_STAT"]==9){temp$veteran[j]<-NA}
  else{temp$veteran[j]<-1}}
# preg
temp$preg[j]<-NA
# resadd
temp$resadd[j]<-str_remove_all(paste(dataset[j,"RES_ADDR_NUM"],
                                     dataset[j,"RES_ADDR1"],
                                     dataset[j,"RES_STREET_DESIG"]),," NA")
# rescity
temp$rescity[j]<-unlist(dataset[j,"RES_CITY"])
# resstate
ifelse(is.na(dataset[j,"RES_CITY_CODE"]),
       yes=temp$resstate[j]<-NA,
       no=ifelse(as.numeric(dataset[j,"RES_CITY_CODE"])<=351,
                 yes=temp$resstate[j]<-"MASSACHUSETTS",
                 no=temp$resstate[j]<-"OUT OF STATE"))
```

```
# reszip
temp$reszip[j]<-unlist(dataset[j,"RES_ZIP"])
# resnat
temp$resnat[j]<-NA
# dplace
ifelse(dataset[j,"DPLACE"]==1,
       yes=temp$dplace[j]<-1,
       no=ifelse(dataset[j,"DPLACE"]==2,
                 yes=temp$dplace[j]<-2,
                 no=ifelse(dataset[j,"DPLACE"]==3,
                           yes=temp$dplace[j]<-3,
                           no=ifelse(dataset[j,"DPLACE"]==5,
                                     yes=temp$dplace[j]<-6,
                                     no=ifelse(dataset[j,"DPLACE"]==6,
                                               yes=temp$dplace[j]<-4,
                                               no=ifelse(dataset[j,"DPLACE"]==7,
                                                         yes=temp$dplace[j]<-8,
                                                         no=temp$dplace[j]<-NA))))))
# dfacilitynum
if(dataset[j,"FACCODE"]=="0000"|
   dataset[j,"FACCODE"]=="0060"|
   dataset[j,"FACCODE"]=="0070"|
   dataset[j,"FACCODE"]=="0080"|
   dataset[j,"FACCODE"]=="0090"|
   dataset[j,"FACCODE"]=="9999"){temp$dfacilitynum[j]<-NA}
else{temp$dfacilitynum[j]<-unlist(dataset[j,"FACCODE"])}
# dadd
temp$ddad[j]<-NA
# dcity
temp$dcity[j]<-unlist(dataset[j,"DNAME_CITY"])
# dstate
ifelse(dataset[j,"DSTATEL"]=="MA",
       yes=temp$dstate[j]<-"MASSACHUSETTS",
       no=ifelse(dataset[j,"DSTATEL"]=="MASSACHUSETTS",
                 yes=temp$dstate[j]<-"MASSACHUSETTS",
                 no=temp$dstate[j]<-NA))
# dzip
temp$dzip[j]<-NA
# dnat
temp$dnat[j]<-"UNITED STATES"
# travel
ifelse(!is.na(dataset[j,"RES_CITY"])&!is.na(dataset[j,"DNAME_CITY"]),
       yes=ifelse(dataset[j,"RES_CITY"]==dataset[j,"DNAME_CITY"],
                  yes=temp$travel[j]<-0,
                  no=temp$travel[j]<-1),
       no=temp$travel[j]<-NA)
# All icd variables
y<-str_trim(str_split(str_replace_all(dataset[j,"TRX_REC_AXIS_CD"],"  "," "),
                      " ",simplify=T),side="both")
x<-vector(mode="character")
l<-1
for(k in 1:length(y)){
  if(str_length(y[k])>4){
```

```
      if(str_length(y[k])<6)
      {x[l]<-str_remove(y[k],".$")
      l<-l+1}
      else
      {x[l]<-str_split(y[k],"0",simplify=T)[1]
      l<-l+1
      x[l]<-str_split(y[k],"0",simplify=T)[2]
      l<-l+1}}
    else
    {x[l]<-y[k]
    l<-l+1}}
  # icd1
  if(is.na(x[1])){temp$icd1[j]<-NA}
  else{temp$icd1[j]<-x[1]}
  # icd2
  if(is.na(x[2])){temp$icd2[j]<-NA}
  else{temp$icd2[j]<-x[2]}
  # icd3
  if(is.na(x[3])){temp$icd3[j]<-NA}
  else{temp$icd3[j]<-x[3]}
  # icd4
  if(is.na(x[4])){temp$icd4[j]<-NA}
  else{temp$icd4[j]<-x[4]}
  # icd5
  if(is.na(x[5])){temp$icd5[j]<-NA}
  else{temp$icd5[j]<-x[5]}
  # icd6
  if(is.na(x[6])){temp$icd6[j]<-NA}
  else{temp$icd6[j]<-x[6]}
  # icd7
  if(is.na(x[7])){temp$icd7[j]<-NA}
  else{temp$icd7[j]<-x[7]}
  # icd8
  if(is.na(x[8])){temp$icd8[j]<-NA}
  else{temp$icd8[j]<-x[8]}
  # icd9
  if(is.na(x[9])){temp$icd9[j]<-NA}
  else{temp$icd9[j]<-x[9]}
  # icd10
  if(is.na(x[10])){temp$icd10[j]<-NA}
  else{temp$icd10[j]<-x[10]}
  # icd11
  if(is.na(x[11])){temp$icd11[j]<-NA}
  else{temp$icd11[j]<-x[11]}
  # icd12
  if(is.na(x[12])){temp$icd12[j]<-NA}
  else{temp$icd12[j]<-x[12]}
  # icd13
  if(is.na(x[13])){temp$icd13[j]<-NA}
  else{temp$icd13[j]<-x[13]}
  #icd14
  if(is.na(x[14])){temp$icd14[j]<-NA}
  else{temp$icd14[j]<-x[14]}
```

```
    # icd15
    if(is.na(x[15])){temp$icd15[j]<-NA}
    else{temp$icd15[j]<-x[15]}
    #icd16
    if(is.na(x[16])){temp$icd16[j]<-NA}
    else{temp$icd16[j]<-x[16]}
  }
  return(as_tibble(temp))
}
```

```
#### For deaths late 2014-2017 ####
vital.14.17<-function(dataset){
  temp<-list()
  # batch
  temp$batch<-2
  for(i in 1:nrow(dataset)){
    # sfnum
    temp$sfnum[i]<-unlist(dataset[i,"SFN_NUM"])
    # ddate
    temp$ddate[i]<-paste0(str_sub(dataset[i,"DOD_4_FD"],7,10),"-",
                          str_sub(dataset[i,"DOD_4_FD"],1,2),"-",
                          str_sub(dataset[i,"DOD_4_FD"],4,5))
    # male
    ifelse(dataset[i,"SEX"]=="M",
           yes=temp$male[i]<-1,
           no=ifelse(dataset[i,"SEX"]=="F",
                     yes=temp$male[i]<-0,
                     no=temp$male[i]<-NA))
    # age
    if(dataset[i,"AGETYPE"]==1)
    {temp$age[i]<-unlist(dataset[i,"AGE1_CALC"])}
    else{if(dataset[i,"AGETYPE"]==2|
            dataset[i,"AGETYPE"]==3)
    {temp$age[i]<-0}
      else{if(dataset[i,"AGETYPE"]==8|
              dataset[i,"AGETYPE"]==9)
      {temp$age[i]<-NA}}}
    # race
    ifelse(str_count(paste0(dataset[i,"RACE1"],
                            dataset[i,"RACE_AM_NATIVE"],
                            dataset[i,"RACE_ASIAN"],
                            dataset[i,"RACE_BLACK"],
                            dataset[i,"DETHNIC4"]),"Y")>1,
           yes=temp$race[i]<-6,
           no=ifelse(dataset[i,"RACE_HISP_LAT_WHITE"]=="Y"|
                     dataset[i,"RACE_HISP_LAT_BLACK"]=="Y"|
                     dataset[i,"DETHNIC4"]=="Y",
                     yes=temp$race[i]<-3,
                     no=ifelse(dataset[i,"RACE1"]=="Y",
                               yes=temp$race[i]<-1,
                               no=ifelse(dataset[i,"RACE_BLACK"]=="Y",
                                         yes=temp$race[i]<-2,
                                         no=ifelse(dataset[i,"RACE_ASIAN"]=="Y",
                                                   yes=temp$race[i]<-4,
                                                   no=ifelse(dataset[i,"RACE_AM_NATIVE"]=="Y",
                                                             yes=temp$race[i]<-5,
                                                             no=ifelse(dataset[i,"RACE_UNK"]=="Y",
                                                                       yes=temp$race[i]<-NA,
                                                                       no=temp$race[i]<-7)))))))
    # occup
    temp$occup[i]<-unlist(dataset[i,"OCCUP"])
    # indust
    temp$indust[i]<-unlist(dataset[i,"INDUST"])
```

```
    # edu
    ifelse(dataset[i,"DEDUC"]==1|
           dataset[i,"DEDUC"]==2,
           yes=temp$edu[i]<-1,
           no=ifelse(dataset[i,"DEDUC"]==3|
                     dataset[i,"DEDUC"]==4|
                     dataset[i,"DEDUC"]==5,
                     yes=temp$edu[i]<-2,
                     no=ifelse(dataset[i,"DEDUC"]==6|
                               dataset[i,"DEDUC"]==7,
                               yes=temp$edu[i]<-3,
                               no=ifelse(dataset[i,"DEDUC"]==8|
                                         dataset[i,"DEDUC"]==9,
                                         yes=temp$edu[i]<-4,
                                         no=ifelse(dataset[i,"DEDUC"]==12,
                                                   yes=temp$edu[i]<-5,
                                                   no=temp$edu[i]<-NA)))))
    # immig
    ifelse(dataset[i,"RES_COUNTRY"]=="UNITED STATES",
           yes=ifelse(dataset[i,"BPLACE_CNT"]=="UNITED STATES",
                      yes=temp$immig[i]<-0,
                      no=temp$immig[i]<-1),
           no=ifelse(dataset[i,"BPLACE_CNT"]=="UNITED STATES",
                     yes=temp$immig[i]<-3,
                     no=temp$immig[i]<-2))
    # pimmig
    ifelse(!(dataset[i,"FATHER_BCOUNTRY"]=="UNITED STATES"|
           dataset[i,"FATHER_BCOUNTRY"]=="UNKNOWN"),
           yes=ifelse(!(dataset[i,"MOTHER_BCOUNTRY"]=="UNITED STATES"|
                      dataset[i,"MOTHER_BCOUNTRY"]=="UNKNOWN"),
                      yes=temp$pimmig[i]<-2,
                      no=temp$pimmig[i]<-1),
           no=ifelse(!(dataset[i,"MOTHER_BCOUNTRY"]=="UNITED STATES"|
                     dataset[i,"MOTHER_BCOUNTRY"]=="UNKNOWN"),
                     yes=temp$pimmig[i]<-1,
                     no=ifelse(dataset[i,"FATHER_BCOUNTRY"]=="UNITED STATES"&
                               dataset[i,"MOTHER_BCOUNTRY"]=="UNITED STATES",
                               yes=temp$pimmig[i]<-0,
                               no=temp$pimmig[i]<-NA)))
    # marital
    ifelse(dataset[i,"MARITAL"]=="M"|
           dataset[i,"MARITAL"]=="A",
           yes=temp$marital[i]<-1,
           no=ifelse(dataset[i,"MARITAL"]=="W",
                     yes=temp$marital[i]<-3,
                     no=ifelse(dataset[i,"MARITAL"]=="D",
                               yes=temp$marital[i]<-4,
                               no=ifelse(dataset[i,"MARITAL"]=="S",
                                         yes=temp$marital[i]<-5,
                                         no=temp$marital[i]<-NA))))
    # veteran
    ifelse(dataset[i,"ARMED"]=="Y",
           yes=temp$veteran[i]<-1,
```

```
          no=ifelse(dataset[i,"ARMED"]=="N",
                    yes=temp$veteran[i]<-0,
                    no=temp$veteran[i]<-NA))
  # preg
  if(is.na(dataset[i,"PREG"])){temp$preg[i]<-NA}
  else{if(dataset[i,"PREG"]==1){temp$preg[i]<-0}
  else{if(dataset[i,"PREG"]==2){temp$preg[i]<-1}
       else{if(dataset[i,"PREG"]==3|dataset[i,"PREG"]==4){temp$preg[i]<-2}
         else{temp$preg[i]<-NA}}}}
# resadd
temp$resadd[i]<-str_remove_all(paste(dataset[i,"RES_ADDR_NUM"],
                                     dataset[i,"RES_STREET_PREFIX"],
                                     dataset[i,"RES_ADDR1"],
                                     dataset[i,"RES_STREET_DESIG"],
                                     dataset[i,"RES_STREET_SUFFIX"],
                                     dataset[i,"RES_ADDR2"]),
                               " NA")
# rescity
temp$rescity[i]<-unlist(dataset[i,"RES_CITY"])
# resstate
temp$resstate[i]<-unlist(dataset[i,"RES_STATE"])
# reszip
temp$reszip[i]<-unlist(dataset[i,"RES_ZIP"])
# resnat
temp$resnat[i]<-unlist(dataset[i,"RES_COUNTRY"])
# dplace
ifelse(dataset[i,"DPLACE"]==9,
       yes=temp$dplace[i]<-NA,
       no=temp$dplace[i]<-unlist(dataset[i,"DPLACE"]))
# dfacilitynum
temp$dfacilitynum[i]<-unlist(dataset[i,"DFACILITYL"])
# dadd
temp$ddad[i]<-str_remove_all(paste(dataset[i,"DADDR_NUM"],
                                   dataset[i,"DSTREET_PREFIX"],
                                   dataset[i,"DADDR1"],
                                   dataset[i,"DSTREET_DESIG"],
                                   dataset[i,"DSTREET_SUFFIX"],
                                   dataset[i,"DADDR2"]),
                             " NA")
# dcity
temp$dcity[i]<-unlist(dataset[i,"DNAME_CITY"])
# dstate
temp$dstate[i]<-unlist(dataset[i,"DSTATEL"])
# dzip
temp$dzip[i]<-str_extract(dataset[i,"DZIP9"],"^.{5}")
# dnat
temp$dnat[i]<-unlist(dataset[i,"DCOUNTRY"])
# travel
ifelse(!is.na(dataset[i,"DNAME_CITY"])&
       !is.na(dataset[i,"RES_CITY"]),
       yes=ifelse(!dataset[i,"DNAME_CITY"]==dataset[i,"RES_CITY"],
                  yes=temp$travel[i]<-1,
                  no=temp$travel[i]<-0),
```

```
       no=temp$travel[i]<-NA)
# icd1
z<-str_trim(str_split(str_replace_all(dataset[i,"TRX_REC_AXIS_CD"],"  "," "),
                      " ",simplify=T),side="both")
y<-vector(mode="character")
l<-1
for(k in 1:length(z)){
  if(str_length(z[k])>4){
    if(str_length(z[k])<6)
    {y[l]<-str_remove(z[k],".$")
    l<-l+1}
    else
    {y[l]<-str_split(z[k],"0",simplify=T)[1]
    l<-l+1
    y[l]<-str_split(z[k],"0",simplify=T)[2]
    l<-l+1}}
  else
  {y[l]<-z[k]
  l<-l+1}}
if(is.na(y[1])){temp$icd1[i]<-NA}
else{temp$icd1[i]<-y[1]}
# icd2
if(is.na(y[2])){temp$icd2[i]<-NA}
else{temp$icd2[i]<-y[2]}
# icd3
if(is.na(y[3])){temp$icd3[i]<-NA}
else{temp$icd3[i]<-y[3]}
# icd4
if(is.na(y[4])){temp$icd4[i]<-NA}
else{temp$icd4[i]<-y[4]}
# icd5
if(is.na(y[5])){temp$icd5[i]<-NA}
else{temp$icd5[i]<-y[5]}
# icd6
if(is.na(y[6])){temp$icd6[i]<-NA}
else{temp$icd6[i]<-y[6]}
# icd7
if(is.na(y[7])){temp$icd7[i]<-NA}
else{temp$icd7[i]<-y[7]}
# icd8
if(is.na(y[8])){temp$icd8[i]<-NA}
else{temp$icd8[i]<-y[8]}
# icd9
if(is.na(y[9])){temp$icd9[i]<-NA}
else{temp$icd9[i]<-y[9]}
# icd10
if(is.na(y[10])){temp$icd10[i]<-NA}
else{temp$icd10[i]<-y[10]}
# icd11
if(is.na(y[11])){temp$icd11[i]<-NA}
else{temp$icd11[i]<-y[11]}
# icd12
if(is.na(y[12])){temp$icd12[i]<-NA}
```

```
  else{temp$icd12[i]<-y[12]}
  # icd13
  if(is.na(y[13])){temp$icd13[i]<-NA}
  else{temp$icd13[i]<-y[13]}
  #icd14
  if(is.na(y[14])){temp$icd14[i]<-NA}
  else{temp$icd14[i]<-y[14]}
  # icd15
  if(is.na(y[15])){temp$icd15[i]<-NA}
  else{temp$icd15[i]<-y[15]}
  #icd16
  if(is.na(y[16])){temp$icd16[i]<-NA}
  else{temp$icd16[i]<-y[16]}
  }
  return(as_tibble(temp))
}
```

Simulated Data

Individual Data Set

The following data set mimics that created by the above functions.

```
#### Preparing occupation data ####
occup<-occups%>%
  transmute(var=map2_chr(.x=occup,.y=indust,.f=~paste(.x,.y,sep=";")))
occup<-sample(occup[[1]],500,replace=T)
occup<-data.frame(occup)%>%
  separate(occup, into=c("occup","indust"),sep=";")

#### Subsetting icd10 codes ####
# To replicate repetition in actual data set
# Header indicates if it it a HIPAA valid code
icd10<-sample_n(icd10[icd10$Header==1,],20000)

#### Coded data ####
# For reproducability
set.seed(8282019)

ind<-data.frame(
  batch=sample(c(1,2),500,replace=T),
  sfnum=sample(0:999999,500,replace=F),
  ddate=sample(seq(as.Date("2000-01-01"),as.Date("2017-12-31"),by="day"),500,replace=T),
  male=sample(0:1,500,replace=T),
  age=sample(0:100,500,replace=T),
  race=sample(1:7,500,replace=T),
  occup=occup$occup,
  indust=occup$indust,
  edu=sample(1:5,500,replace=T),
  immig=sample(0:4,500,replace=T),
  pimmig=sample(0:2,500,replace=T),
  marital=sample(c(1,3:5),500,replace=T),
  veteran=sample(0:1,500,replace=T),
  preg=sample(0:2,500,replace=T),
  resadd="1234 CIRCLE ST",
  rescity= "ANYTOWN",
  resstate= "MASSACHUSETTS",
  reszip=sample(shp@data$ZCTA5CE10,500,replace=T),
  resnat="UNITED STATES",
  dplace=sample(1:8,500,replace=T),
  dfacilitynum="000000",
  ddad="1234 SQUARE ST",
  dcity="ANYTOWN",
  dstate="MASSACHUSETTS",
  dzip=sample(shp@data$ZCTA5CE10,500,replace=T),
  dnat="UNITED STATES",
  travel=sample(0:1,500,replace=T),
  icd1=sample(icd10$ICD10Code,500,replace=T),
  icd2=sample(icd10$ICD10Code,500,replace=T),
  icd3=sample(icd10$ICD10Code,500,replace=T),
  icd4=sample(icd10$ICD10Code,500,replace=T),
```

```
  icd5=sample(icd10$ICD10Code,500,replace=T),
  icd6=sample(icd10$ICD10Code,500,replace=T),
  icd7=sample(icd10$ICD10Code,500,replace=T),
  icd8=sample(icd10$ICD10Code,500,replace=T),
  icd9=sample(icd10$ICD10Code,500,replace=T),
  icd10=sample(icd10$ICD10Code,500,replace=T),
  icd11=sample(icd10$ICD10Code,500,replace=T),
  icd12=sample(icd10$ICD10Code,500,replace=T),
  icd13=sample(icd10$ICD10Code,500,replace=T),
  icd14=sample(icd10$ICD10Code,500,replace=T),
  icd15=sample(icd10$ICD10Code,500,replace=T),
  icd16=sample(icd10$ICD10Code,500,replace=T))

#### Factored data ####
ind.f<-ind%>%
  # Changing coding to descriptive factors
  mutate(batch=factor(batch,levels=c(1:2)),
         male=factor(male,levels=c(0:1),labels=c("FEMALE","MALE")),
         race=factor(race,levels=c(1:7),
                     labels=c("NON-HISPANIC WHITE",
                              "NON-HISPANIC BLACK",
                              "HISPANIC / LATINO",
                              "ASIAN",
                              "NATIVE AMERICAN / AMERICAN INDIAN / ALASKA NATIVE",
                              "MULTI-RACIAL",
                              "OTHER")),
         edu=factor(edu,levels=c(1:5),
                    labels=c("LESS THAN HIGHSCHOOL",
                             "HIGH SCHOOL / GED / CERTIFICATE / SOME COLLEGE",
                             "BACHELOR'S / ASSOCIATE'S DEGREE",
                             "MASTER'S DEGREE OR HIGHER",
                             "SPECIAL EDUCATION")),
         immig=factor(immig,levels=c(0:4),
                      labels=c("BORN AND LIVE IN US",
                               "BORN ELSEWHERE AND LIVE IN US",
                               "BORN ELSEWHERE AND LIVE ELSEWHERE",
                               "BORN IN US AND LIVE ELSEWHERE",
                               "BORN ELSEWHERE")),
         pimmig=factor(pimmig,levels=c(0:2),
                       labels=c("BOTH PARENTS BORN IN US",
                                "AT LEAST ONE PARENT BORN OUTSIDE US",
                                "BOTH PARENTS BORN OUTSIDE US")),
         marital=factor(marital,levels=c(1,3:5),
                        labels=c("MARRIED OR SEPERATED",
                                 "WIDOWED",
                                 "DIVORCED",
                                 "NEVER MARRIED")),
         veteran=factor(veteran,levels=c(0:1),
                        labels=c("NOT A VETERAN",
                                 "VETERAN")),
         preg=factor(preg,levels=c(0:2),
                     labels=c("NOT PREGNANT IN LAST YEAR",
                              "PREGNANT AT DEATH",
```

```
                              "NOT PREGNANT AT DEATH, PREGNANT IN LAST YEAR")),
         dplace=factor(dplace,levels=c(1:8),
                       labels=c("HOSPITAL, INPATIENT",
                                "HOSPITAL, OUTPATIENT / ER",
                                "HOSPITAL, DOA",
                                "RESIDENCE",
                                "HOSPICE",
                                "NURSING HOME",
                                "ASSISTED LIVING FACILITY / REST HOME",
                                "OTHER")),
         travel=factor(travel,levels=c(0:1),
                       labels=c("DIED AND RESIDE IN SAME CITY",
                                "DIED AND RESIDE IN DIFFERENT CITIES")))

#### Class conversions ####
# Necessitated by simulated data, not included in actual analysis
ind.f$dcity<-as.character(ind.f$dcity)
ind.f$ddad<-as.character(ind.f$ddad)
ind.f$dfacilitynum<-as.character(ind.f$dfacilitynum)
ind.f$dnat<-as.character(ind.f$dnat)
ind.f$dstate<-as.character(ind.f$dstate)
ind.f$dzip<-as.character(ind.f$dzip)
ind.f$icd1<-as.character(ind.f$icd1)
ind.f$icd2<-as.character(ind.f$icd2)
ind.f$icd3<-as.character(ind.f$icd3)
ind.f$icd4<-as.character(ind.f$icd4)
ind.f$icd5<-as.character(ind.f$icd5)
ind.f$icd6<-as.character(ind.f$icd6)
ind.f$icd7<-as.character(ind.f$icd7)
ind.f$icd8<-as.character(ind.f$icd8)
ind.f$icd9<-as.character(ind.f$icd9)
ind.f$icd10<-as.character(ind.f$icd10)
ind.f$icd11<-as.character(ind.f$icd11)
ind.f$icd12<-as.character(ind.f$icd12)
ind.f$icd13<-as.character(ind.f$icd13)
ind.f$icd14<-as.character(ind.f$icd14)
ind.f$icd15<-as.character(ind.f$icd15)
ind.f$icd16<-as.character(ind.f$icd16)
ind.f$indust<-as.character(ind.f$indust)
ind.f$occup<-as.character(ind.f$occup)
ind.f$resadd<-as.character(ind.f$resadd)
ind.f$rescity<-as.character(ind.f$rescity)
ind.f$resnat<-as.character(ind.f$resnat)
ind.f$resstate<-as.character(ind.f$resstate)
ind.f$reszip<-as.character(ind.f$reszip)
ind.f$sfnum<-as.character(ind.f$sfnum)
```

```r
head(ind.f,2)
```

```
##   batch  sfnum      ddate   male age               race        occup
## 1     2 635949 2014-10-01 FEMALE   7 NON-HISPANIC WHITE       Waiter
## 2     2 844076 2006-09-21   MALE   1       MULTI-RACIAL Veterinarian
##          indust               edu                             immig
## 1 Food/Beverage SPECIAL EDUCATION               BORN AND LIVE IN US
## 2       Medical SPECIAL EDUCATION BORN ELSEWHERE AND LIVE ELSEWHERE
##                    pimmig       marital       veteran
## 1 BOTH PARENTS BORN IN US      DIVORCED       VETERAN
## 2 BOTH PARENTS BORN IN US NEVER MARRIED NOT A VETERAN
##                        preg         resadd rescity      resstate reszip
## 1         PREGNANT AT DEATH 1234 CIRCLE ST ANYTOWN MASSACHUSETTS  02367
## 2 NOT PREGNANT IN LAST YEAR 1234 CIRCLE ST ANYTOWN MASSACHUSETTS  01012
##          resnat              dplace dfacilitynum           ddad   dcity
## 1 UNITED STATES HOSPITAL, INPATIENT       000000 1234 SQUARE ST ANYTOWN
## 2 UNITED STATES             HOSPICE       000000 1234 SQUARE ST ANYTOWN
##          dstate  dzip          dnat                              travel
## 1 MASSACHUSETTS 01562 UNITED STATES        DIED AND RESIDE IN SAME CITY
## 2 MASSACHUSETTS 02462 UNITED STATES DIED AND RESIDE IN DIFFERENT CITIES
##      icd1    icd2    icd3    icd4    icd5    icd6    icd7    icd8    icd9
## 1 T22349A S72114R T2210XS  M11042 S27322D V8612XS S72102A S065X3S S66128S
## 2 S72441B S63281D T79A19S Y384X2S T401X4S  Z91012 S40819D V00832A S62304B
##    icd10   icd11   icd12   icd13   icd14   icd15   icd16
## 1 M05742 S271XXA T8325XS  H31012    Z947  M24652   Z8759
## 2  Z0571 T8059XD T34811D T425X6A T84398S S65209S V9522XD
```

Aggregate Data Set

```
# All possible combination of month, year, and zip
# Data set intended to be mapped so 0 is preferable over NA
base<-data.frame(zip=rep(shp@data$ZCTA5CE10,(12*18)),
                 month=rep(c(rep(1,538),rep(2,538),rep(3,538),rep(4,538),
                         rep(5,538),rep(6,538),rep(7,538),rep(8,538),
                         rep(9,538),rep(10,538),rep(11,538),rep(12,538)),18),
                 year=c(rep(2000,538*12),rep(2001,538*12),rep(2002,538*12),
                        rep(2003,538*12),rep(2004,538*12),rep(2005,538*12),
                        rep(2006,538*12),rep(2007,538*12),rep(2008,538*12),
                        rep(2009,538*12),rep(2010,538*12),rep(2011,538*12),
                        rep(2012,538*12),rep(2013,538*12),rep(2014,538*12),
                        rep(2015,538*12),rep(2016,538*12),rep(2017,538*12)))
agMonth<-base%>%
  # join to aggregated counts
  left_join(ind%>%
              # Extract death year and month from date object
              mutate(dyear=year(as.Date(ddate)),
                     dmonth=month(as.Date(ddate)))%>%
              # Determine number of cases in each zip code in each month
              group_by(reszip,dyear,dmonth)%>%
              summarize(cases=n()),
            by=c("zip"="reszip","month"="dmonth","year"="dyear"))%>%
  # Turn NA to 0
  mutate(cases=map_dbl(.x=cases,.f=~if(is.na(.x)){return(0)}else{return(.x)}))

head(agMonth)
```

```
##     zip month year cases
## 1 02536     1 2000     0
## 2 02556     1 2000     0
## 3 02540     1 2000     0
## 4 02646     1 2000     0
## 5 01237     1 2000     0
## 6 01259     1 2000     0
```

Exploratory Data Analysis

Summary Tables

These tables give an overview of the data based on the type of data it is. Helpful during the data cleaning stage when deciding on possible class conversions.

```
#### Numeric Variables ####
ind.f%>%
  select(names(ind[map_lgl(ind.f,is.numeric)]))%>%
  gather(colnames(ind[map_lgl(ind.f,is.numeric)]),key=variable,value=value)%>%
  group_by(variable)%>%
  summarize(Mean=mean(value,na.rm=T),
  SD=sd(value,na.rm=T),
  R1=range(value,na.rm=T)[1],
  R2=range(value,na.rm=T)[2],
  UniqueValues=length(unique(value[!is.na(value)])),
  PropMissingness=sum(is.na(value))/length(value))%>%
  mutate(Range=paste0("[",R1,", ",R2,"]"),
         percent=paste0(round(PropMissingness*100,3),"%"))%>%
  select(Variable=variable,Mean,SD,Range,`Unique Values`=UniqueValues,
         Missingness=percent)%>%
  kable(booktabs=T,digits=3,
        caption="Summary of Quantitative Variables in Individual Data Set",align="c")%>%
  kable_styling(latex_options=c("HOLD_position","striped"),position="center")
```

Table 1: Summary of Quantitative Variables in Individual Data Set

Variable	Mean	SD	Range	Unique Values	Missingness
age	51.112	28.89	[0, 100]	101	0%

```
#### Character variables ####
ind.f%>%
  select(names(ind.f[map_lgl(ind.f,is.character)]))%>%
  gather(variable,value)%>%
  group_by(variable)%>%
  summarize(UniqueValues=length(unique(value[!is.na(value)])),
  PropMissingness=sum(is.na(value))/length(value))%>%
  mutate(percent=paste0(round(PropMissingness*100,3),"%"))%>%
  select(Variable=variable,`Unique Values`=UniqueValues,Missingness=percent)%>%
  kable(booktabs=T,digits=3,
        caption="Summary of Character Variables in Individual Data Set",align="c")%>%
  kable_styling(latex_options=c("HOLD_position","striped"),position="center")
```

Table 2: Summary of Character Variables in Individual Data Set

Variable	Unique Values	Missingness
dcity	1	0%
ddad	1	0%
dfacilitynum	1	0%
dnat	1	0%
dstate	1	0%
dzip	322	0%
icd1	493	0%
icd10	497	0%
icd11	491	0%
icd12	496	0%
icd13	494	0%
icd14	491	0%
icd15	498	0%
icd16	493	0%
icd2	495	0%
icd3	495	0%
icd4	495	0%
icd5	495	0%
icd6	496	0%
icd7	491	0%
icd8	491	0%
icd9	490	0%
indust	15	0%
occup	30	0%
resadd	1	0%
rescity	1	0%
resnat	1	0%
resstate	1	0%
reszip	326	0%
sfnum	500	0%

```
#### Factor variables ####
var<-names(ind.f)[map_lgl(ind.f,is.factor)]
fac<-data.frame()
for(i in 1:11){
  df<-as.data.frame(table(ind.f[,var[i]],useNA="always"))
  df$name<-var[i]
  fac<-bind_rows(fac,df)
}
fac%>%
  select(name,everything())%>%
  mutate(Freq=prettyNum(Freq,big.mark=","))%>%
  kable(booktabs=T,longtable=T,digits=3,
        caption="Summary of Categorical Variables in Individual Data Set",
        align="c",
        col.names=c("Variable","Value","Frequency"))%>%
  kable_styling(latex_options=c("HOLD_position","repeat_header","striped"),
                position="center")%>%
```

```
collapse_rows(columns=1,latex_hline="major",valign="middle")
```

Table 3: Summary of Categorical Variables in Individual Data Set

Variable	Value	Frequency
batch	1	241
	2	259
	NA	0
male	FEMALE	244
	MALE	256
	NA	0
race	NON-HISPANIC WHITE	48
	NON-HISPANIC BLACK	66
	HISPANIC / LATINO	75
	ASIAN	80
	NATIVE AMERICAN / AMERICAN INDIAN / ALASKA NATIVE	66
	MULTI-RACIAL	84
	OTHER	81
	NA	0
edu	LESS THAN HIGHSCHOOL	100
	HIGH SCHOOL / GED / CERTIFICATE / SOME COLLEGE	101
	BACHELOR'S / ASSOCIATE'S DEGREE	109
	MASTER'S DEGREE OR HIGHER	103
	SPECIAL EDUCATION	87
	NA	0
immig	BORN AND LIVE IN US	96
	BORN ELSEWHERE AND LIVE IN US	107
	BORN ELSEWHERE AND LIVE ELSEWHERE	108
	BORN IN US AND LIVE ELSEWHERE	96
	BORN ELSEWHERE	93
	NA	0
pimmig	BOTH PARENTS BORN IN US	173
	AT LEAST ONE PARENT BORN OUTSIDE US	158
	BOTH PARENTS BORN OUTSIDE US	169
	NA	0
marital	MARRIED OR SEPERATED	122
	WIDOWED	127
	DIVORCED	123
	NEVER MARRIED	128
	NA	0
veteran	NOT A VETERAN	247
	VETERAN	253
	NA	0
preg	NOT PREGNANT IN LAST YEAR	177
	PREGNANT AT DEATH	168
	NOT PREGNANT AT DEATH, PREGNANT IN LAST YEAR	155
	NA	0
	HOSPITAL, INPATIENT	68

Table 3: Summary of Categorical Variables in Individual Data Set
(continued)

Variable	Value	Frequency
dplace	HOSPITAL, OUTPATIENT / ER	68
	HOSPITAL, DOA	69
	RESIDENCE	50
	HOSPICE	66
	NURSING HOME	55
	ASSISTED LIVING FACILITY / REST HOME	63
	OTHER	61
	NA	0
travel	DIED AND RESIDE IN SAME CITY	239
	DIED AND RESIDE IN DIFFERENT CITIES	261
	NA	0

```
#### Date Variables ####
ind.f%>%
  select(names(ind.f[map_lgl(ind.f,is.Date)]))%>%
  gather(variable,value)%>%
  group_by(variable)%>%
  summarize(range=paste0("[",
                         format(range(value,na.rm=T)[1],"%b %d, %Y"),"; ",
                         format(range(value,na.rm=T)[2],"%b %d, %Y"),"]"),
            UniqueValues=length(unique(value[!is.na(value)])),
            # In vital data, sometimes death date is unknown
            PropMissingness=paste0(sum(is.na(value))/length(value),"%"))%>%
  kable(booktabs=T,digits=3,
        caption="Summary of Date Variables in Individual Data Set",align="c",
        col.names=c("Variable","Range","Unique Values","Missingness"))%>%
  kable_styling(latex_options=c("HOLD_position","striped"),position="center")
```

Table 4: Summary of Date Variables in Individual Data Set

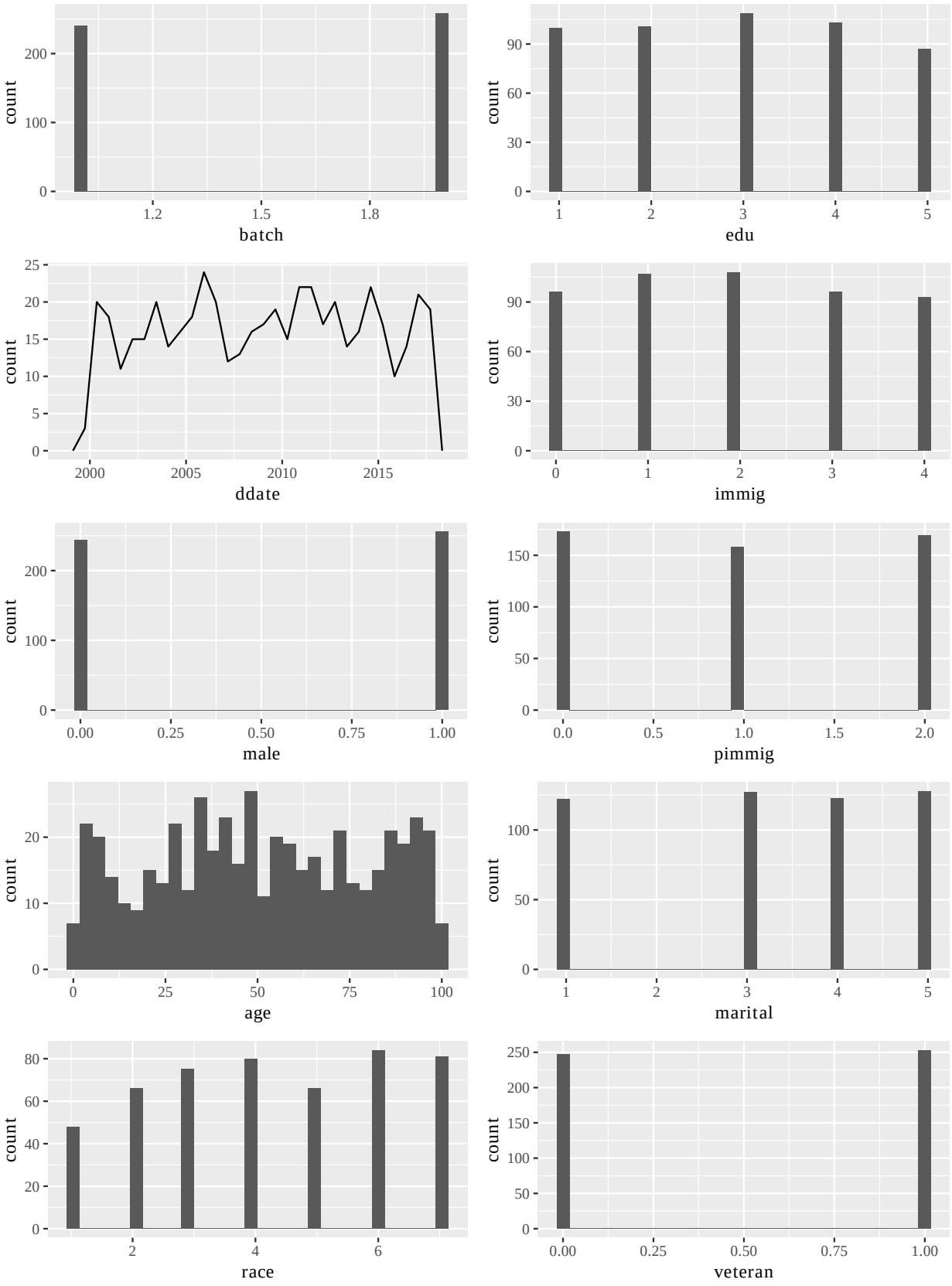
Variable	Range	Unique Values	Missingness
ddate	[Jan 05, 2000; Dec 23, 2017]	479	0%

Univariate Graphics Loop

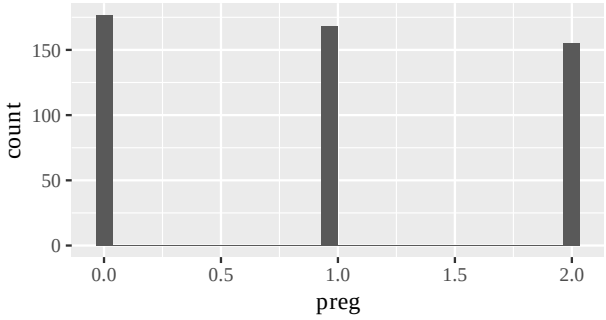
Plots all variables with appropriate graphical summaries. Used to find possible relationships for later investigation. Not all of these graphs are insightful, so an argument (`rmCol`) is available is specify variables for the function to ignore.

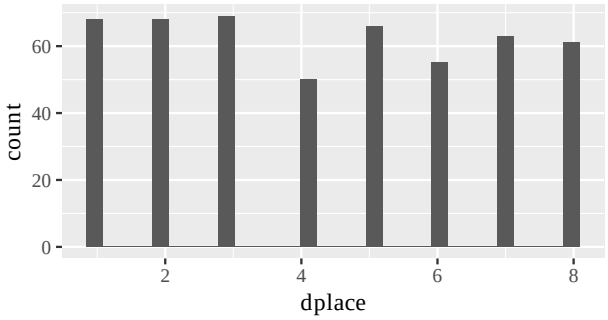
```
uni.plots<-function(dataset,rmCol=NULL){
  if(is.null(rmCol)){data<-dataset}
  else{data<-dataset[,-rmCol]}
  plots<-list()
  for (i in 1:length(data)){
    # Numeric variables
    if (is.numeric(data[[i]])){
      plots[[i]]<- ggplot(data,aes_string(x=colnames(data[i])))+
                   geom_histogram()+
                   xlab(colnames(data[i]))+
                   ylab("count")+
                   theme(text=element_text(family="LM Roman 10"))
    } else {
      # Date variables
      if (is.Date(data[[i]])){
        plots[[i]]<- ggplot(data,aes_string(x=colnames(data[i])))+
                   geom_freqpoly()+
                   xlab(colnames(data[i]))+
                   ylab("count")+
                   theme(text=element_text(family="LM Roman 10"))
      } else {
        # Character / factor variables
        plots[[i]]<- ggplot(data,aes_string(x=colnames(data[i])))+
        geom_bar()+
        xlab(colnames(data[i]))+
        ylab("count")+
        theme(text=element_text(family="LM Roman 10"))
      }
    }
  }
  return(marrangeGrob(plots,nrow=5,ncol=2,
                      top=textGrob("Univariate Plots",
                                   gp=gpar(fontsize=16,
                                           fontfamily="LM Roman 10"))))
}

# Uses coded data for clarity of axis labels
# Coded variables treated as numeric even though usually categorical
# Not an issue since EDA is not used for presentation
uni.plots(ind,c(2,7:8,15:19,21:26,28:length(ind)))
```

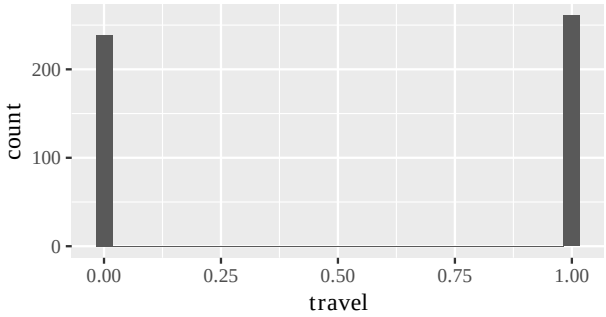
Univariate Plots



Univariate Plots







Cause of Death Tables

```
#### Joining in ICD10 descriptions ####
ind.f<-ind.f%>%
  left_join(icd10[,c(2,5)],by=c("icd1"="ICD10Code"))%>%
  rename(icd1.desc=LongDesc)%>%
  left_join(icd10[,c(2,5)],by=c("icd2"="ICD10Code"))%>%
  rename(icd2.desc=LongDesc)%>%
  left_join(icd10[,c(2,5)],by=c("icd3"="ICD10Code"))%>%
  rename(icd3.desc=LongDesc)%>%
  left_join(icd10[,c(2,5)],by=c("icd4"="ICD10Code"))%>%
  rename(icd4.desc=LongDesc)%>%
  left_join(icd10[,c(2,5)],by=c("icd5"="ICD10Code"))%>%
  rename(icd5.desc=LongDesc)%>%
  left_join(icd10[,c(2,5)],by=c("icd6"="ICD10Code"))%>%
  rename(icd6.desc=LongDesc)%>%
  left_join(icd10[,c(2,5)],by=c("icd7"="ICD10Code"))%>%
  rename(icd7.desc=LongDesc)%>%
  left_join(icd10[,c(2,5)],by=c("icd8"="ICD10Code"))%>%
  rename(icd8.desc=LongDesc)%>%
  left_join(icd10[,c(2,5)],by=c("icd9"="ICD10Code"))%>%
  rename(icd9.desc=LongDesc)%>%
  left_join(icd10[,c(2,5)],by=c("icd10"="ICD10Code"))%>%
  rename(icd10.desc=LongDesc)%>%
  left_join(icd10[,c(2,5)],by=c("icd11"="ICD10Code"))%>%
  rename(icd11.desc=LongDesc)%>%
  left_join(icd10[,c(2,5)],by=c("icd12"="ICD10Code"))%>%
  rename(icd12.desc=LongDesc)%>%
  left_join(icd10[,c(2,5)],by=c("icd13"="ICD10Code"))%>%
  rename(icd13.desc=LongDesc)%>%
  left_join(icd10[,c(2,5)],by=c("icd14"="ICD10Code"))%>%
  rename(icd14.desc=LongDesc)%>%
  left_join(icd10[,c(2,5)],by=c("icd15"="ICD10Code"))%>%
  rename(icd15.desc=LongDesc)%>%
  left_join(icd10[,c(2,5)],by=c("icd16"="ICD10Code"))%>%
  rename(icd16.desc=LongDesc)
```

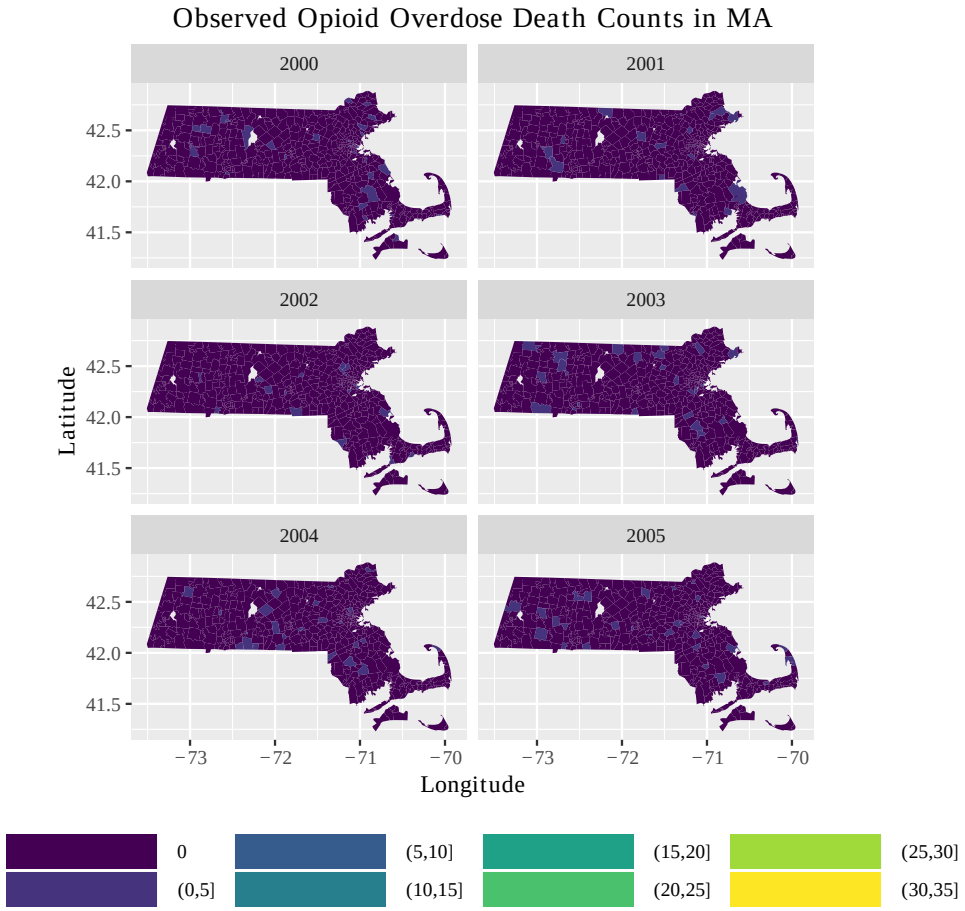
```r
#### Primary ####
# For illustration, project examined secondary and tertiary causes as well
as.data.frame(table(ind.f$icd1.desc))%>%
  top_n(5,Freq)%>%
  kable(booktabs=T,digits=3,longtable=T,caption="Summary of Primary Causes of Death",align="c",
        col.names=c("Cause of Death","Frequency"))%>%
  kable_styling(latex_options=c("HOLD_position","striped","repeat_header"),position="center")%>%
  column_spec(1,width="8in")
```

Table 5: Summary of Primary Causes of Death

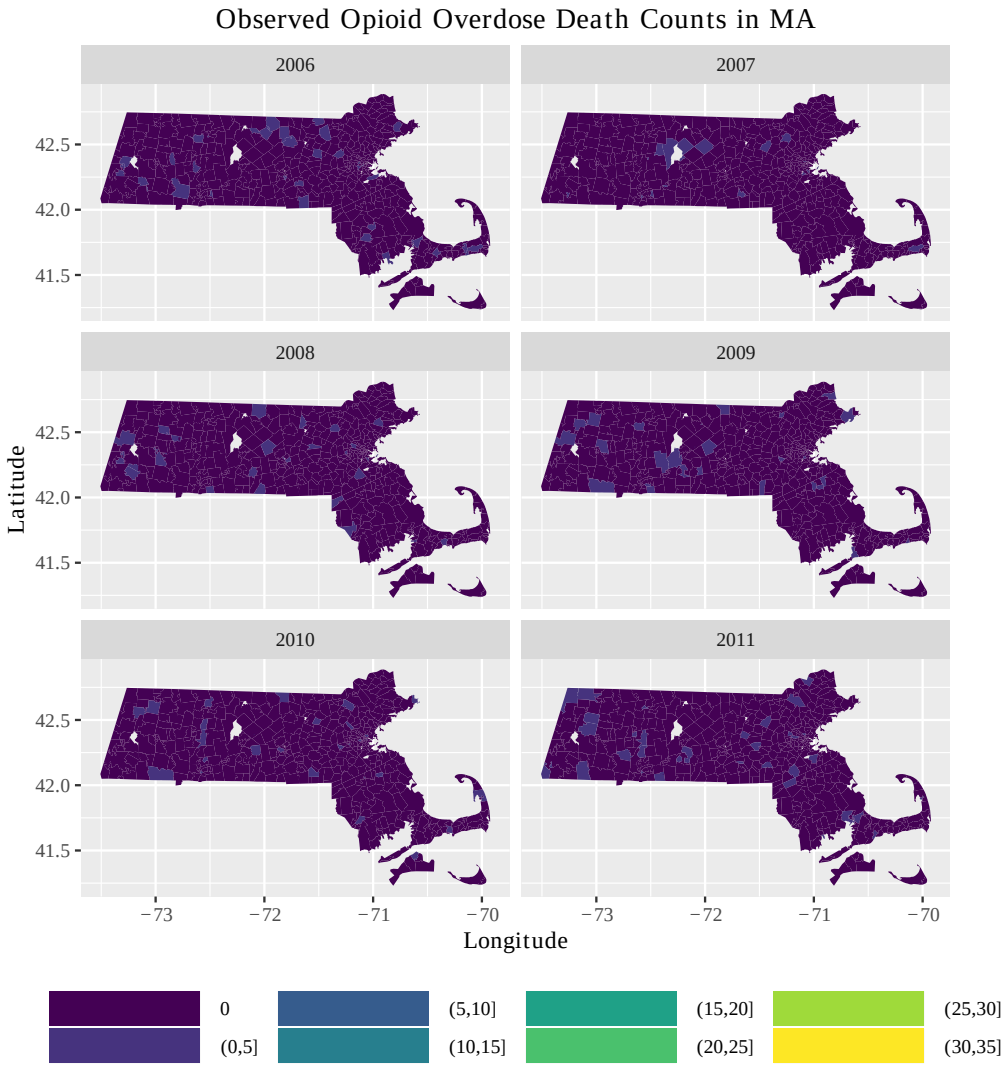
Cause of Death	Frequency
Displaced fracture of distal phalanx of unspecified thumb, initial encounter for open fracture	2
Displaced pilon fracture of left tibia, subsequent encounter for open fracture type I or II with nonunion	2
Maternal care for viable fetus in abdominal pregnancy, third trimester, fetus 2	2
Nondisplaced oblique fracture of shaft of unspecified tibia, initial encounter for open fracture type I or II	2
Retinal neovascularization, unspecified, unspecified eye	2
Thromboembolism in childbirth	2
Unspecified physeal fracture of lower end of humerus, left arm, subsequent encounter for fracture with malunion	2

Mapping

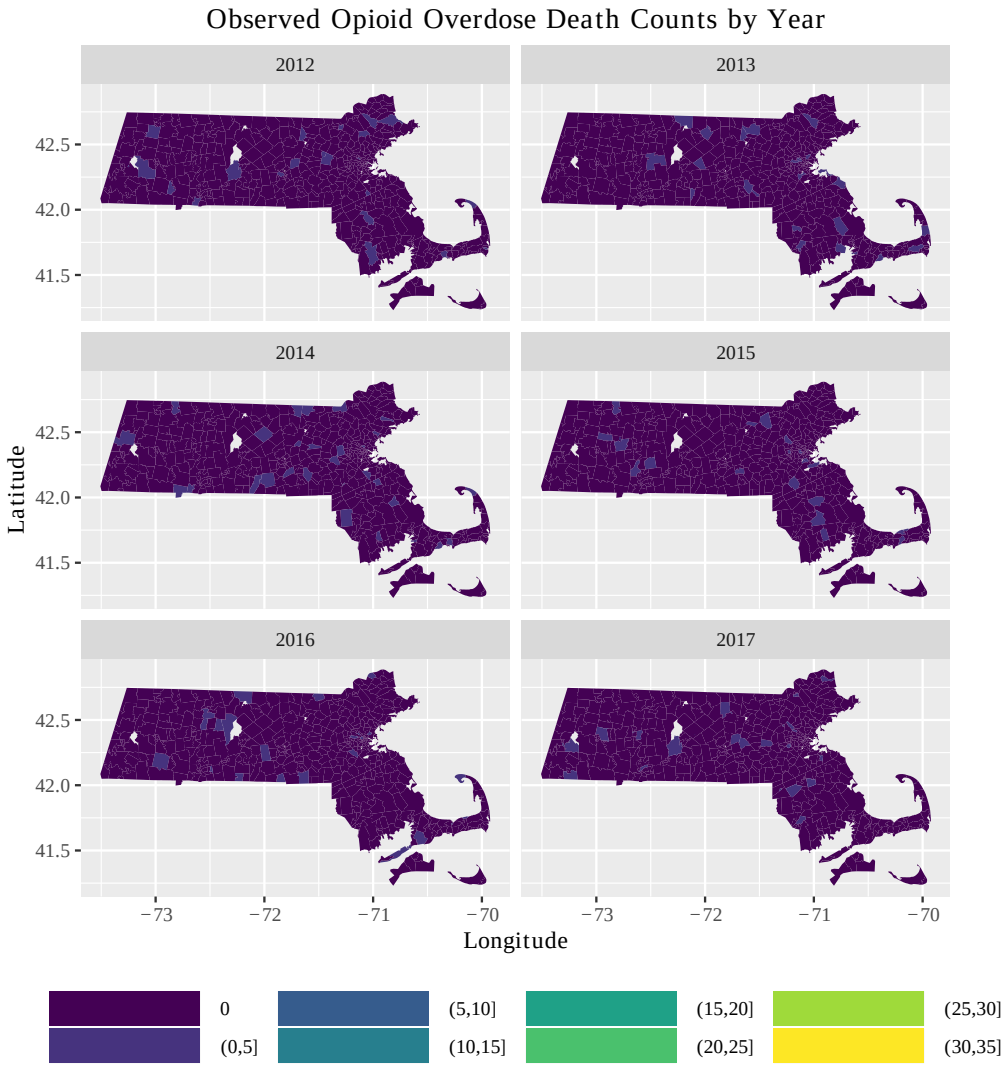
```
map<-fortify(shp,region="ZCTA5CE10")
map<- merge(map, agMonth%>%rename(id=zip)%>%group_by(id,year)%>%
             summarize(cases=sum(cases)), by="id", all.x = TRUE)
map<- map[order(map$order),]
# 2000-2005
ggplot()+
  geom_polygon(data=map[map$year<=2005,],
               aes(long,lat,group=group,
                   fill=cut(cases,c(-1,0,5,10,15,20,25,30,35))),color=NA)+
  coord_fixed(1.2)+
  facet_wrap(~year,ncol=2)+
  scale_fill_viridis(discrete=T,drop=F,na.value="antiquewhite",
                     labels =c("0","(0,5]","(5,10]","(10,15]",
                               "(15,20]","(20,25]","(25,30]",
                               "(30,35]"))+
  labs(x="Longitude",y="Latitude",fill=NULL,
       title ="Observed Opioid Overdose Death Counts in MA")+
  theme(legend.position = "bottom",legend.key.width=unit(2.5,"cm"),
        legend.spacing.x = unit(.3, 'cm'),plot.title = element_text(hjust = 0.5),
        text=element_text(family="LM Roman 10"))
```



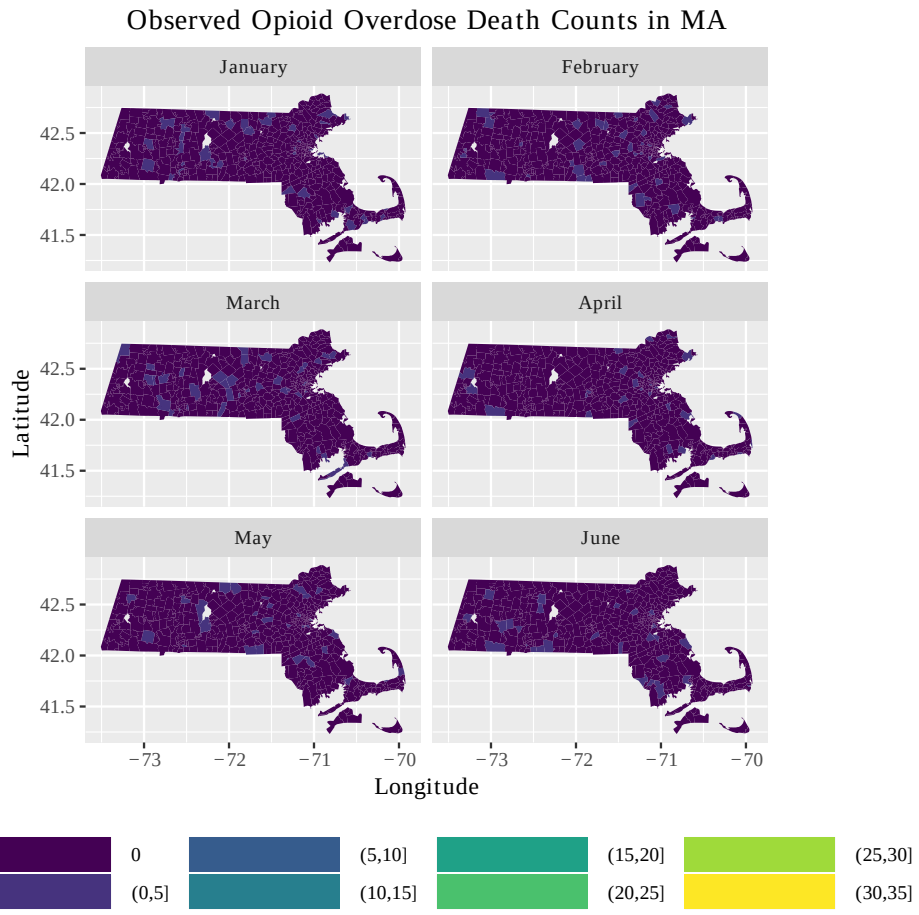
```
#2006-2011
ggplot()+
  geom_polygon(data=map[map$year<=2011&map$year>=2006,],
               aes(long,lat,group=group,
                   fill=cut(cases,c(-1,0,5,10,15,20,25,30,35))),color=NA)+
  coord_fixed(1.2)+
  facet_wrap(~year,ncol=2)+
  scale_fill_viridis(discrete=T,drop=F,na.value="antiquewhite",
                     labels =c("0","(0,5]","(5,10]","(10,15]",
                               "(15,20]","(20,25]","(25,30]",
                               "(30,35]"))+
  labs(x="Longitude",y="Latitude",fill=NULL,
       title ="Observed Opioid Overdose Death Counts in MA")+
  theme(legend.position = "bottom",legend.key.width=unit(2.5,"cm"),
        legend.spacing.x = unit(.3, 'cm'),plot.title = element_text(hjust = 0.5),
        text=element_text(family="LM Roman 10"))
```



```
#2012-2017
ggplot()+
  geom_polygon(data=map[map$year<=2017&map$year>=2012,],
               aes(long,lat,group=group,
                   fill=cut(cases,c(-1,0,5,10,15,20,25,30,35))),color=NA)+
  coord_fixed(1.2)+
  facet_wrap(~year,ncol=2)+
  scale_fill_viridis(discrete=T,drop=F,na.value="antiquewhite",
                     labels =c("0","(0,5]","(5,10]","(10,15]",
                               "(15,20]","(20,25]","(25,30]",
                               "(30,35]"))+
  labs(x="Longitude",y="Latitude",fill=NULL,
       title ="Observed Opioid Overdose Death Counts by Year")+
  theme(legend.position = "bottom",legend.key.width=unit(2.5,"cm"),
        legend.spacing.x = unit(.3, 'cm'),plot.title = element_text(hjust = 0.5),
        text=element_text(family="LM Roman 10"))
```



```
map<-fortify(shp,region="ZCTA5CE10")
map<- merge(map,
            agMonth%>%rename(id=zip)%>%mutate(month=month(month,label=T,abbr=F))%>%
              group_by(id,month)%>%summarize(cases=sum(cases)), by="id", all.x = TRUE)
map<- map[order(map$order),]
# Jan-Jul
ggplot()+
  geom_polygon(data=map[map$month=="January"|map$month=="February"|map$month=="March"|
                        map$month=="April"|map$month=="May"|map$month=="June",],
               aes(long,lat,group=group,
                   fill=cut(cases,c(-1,0,5,10,15,20,25,30,35))),color=NA)+
  coord_fixed(1.2)+
  facet_wrap(~month,ncol=2)+
  scale_fill_viridis(discrete=T,drop=F,na.value="antiquewhite",
                     labels =c("0","(0,5]","(5,10]","(10,15]","(15,20]","(20,25]",
                               "(25,30]","(30,35]"))+
  labs(x="Longitude",y="Latitude",fill=NULL,
       title ="Observed Opioid Overdose Death Counts in MA")+
  theme(legend.position = "bottom",legend.key.width=unit(2.5,"cm"),
        legend.spacing.x = unit(.3, 'cm'),plot.title = element_text(hjust = 0.5),
        text=element_text(family="LM Roman 10"))
```



```
# Aug-Dec
ggplot()+
  geom_polygon(data=map[map$month=="July"|map$month=="August"|
                        map$month=="September"|map$month=="October"|
                        map$month=="November"|map$month=="December",],
               aes(long,lat,group=group,
                   fill=cut(cases,c(-1,0,5,10,15,20,25,30,35))),color=NA)+
  coord_fixed(1.2)+
  facet_wrap(~month,ncol=2)+
  scale_fill_viridis(discrete=T,drop=F,na.value="antiquewhite",
                     labels =c("0","(0,5]","(5,10]","(10,15]","(15,20]",
                               "(20,25]","(25,30]","(30,35]"))+
  labs(x="Longitude",y="Latitude",fill=NULL,
       title ="Observed Opioid Overdose Death Counts in MA")+
  theme(legend.position = "bottom",legend.key.width=unit(2.5,"cm"),
        legend.spacing.x = unit(.3, 'cm'),plot.title = element_text(hjust = 0.5),
        text=element_text(family="LM Roman 10"))
```