Analysis of DIH Prosecutions: 2000 to 2019

Kelly Kung

9/6/2021

Contents

1 Set Up

1.1 R Code

```
#packages we need for this code file
library(ggplot2)
library(mgcv)
```

```
## Loading required package: nlme
```

```
## This is mgcv 1.8-34. For overview type 'help("mgcv-package")'.
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

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

```
library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v tibble  3.0.6     v dplyr   1.0.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
## v purrr   0.3.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x dplyr::collapse()        masks nlme::collapse()
## x lubridate::date()        masks base::date()
## x dplyr::filter()          masks stats::filter()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x lubridate::setdiff()     masks base::setdiff()
## x lubridate::union()       masks base::union()
```

```
library(dplyr)
library(DHARMa)
```

```
## This is DHARMa 0.3.3.0. For overview type '?DHARMa'. For recent changes, type news(package = 'DHARMa
```

```
library(mgcViz)
```

```
## Loading required package: qgam

## Loading required package: rgl

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
## Registered S3 method overwritten by 'mgcViz':
##   method from
##   +.gg   GGally

##
## Attaching package: 'mgcViz'

## The following objects are masked from 'package:stats':
##
##     qqline, qqnorm, qqplot
```

```
library(extrafont)
```

```
## Registering fonts with R
```

```
library(arm)
```

```
## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loading required package: lme4

##
## Attaching package: 'lme4'

## The following object is masked from 'package:nlme':
##
##     lmList

##
## arm (Version 1.11-2, built: 2020-7-27)

## Working directory is /Users/kkung/OneDrive - Boston University/Research-Lok
```

```
loadfonts()
library(stargazer)
```

```
##
## Please cite as:

##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.

##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

```
#define functions we will need for analysis
#expit function
expit<-function(x){
  return(exp(x)/(1 + exp(x)))
}

#logit function
logit<-function(x){
  return(log(x/(1 - x)))
}
```

1.2 Data

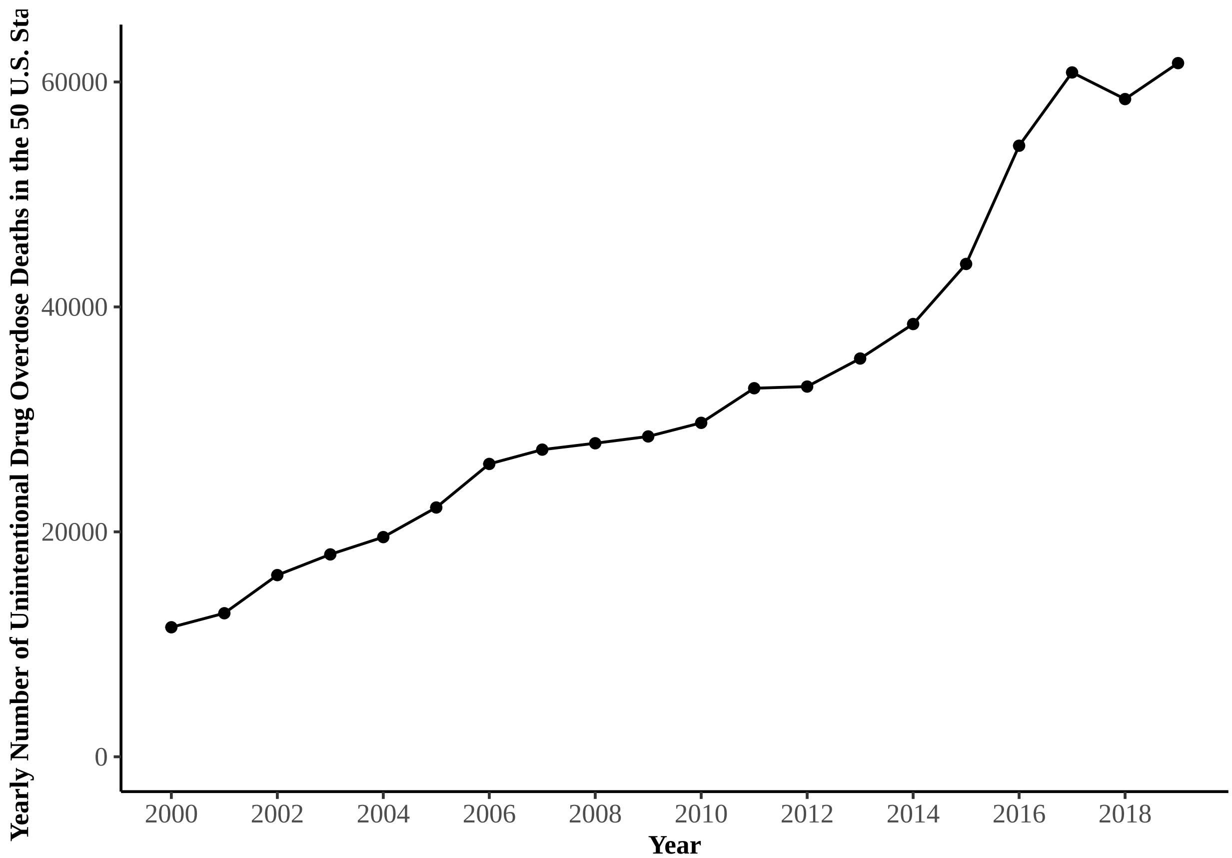
```
#read in data
main_analysis_data<-read.csv("./Data/full_data_set_9_6_21_unintentional.csv")

################################## set up data set ################################
#add the intervention dates and time period data
main_analysis_data$Intervention_First_Date<-as.Date(main_analysis_data$Intervention_First_Date)
main_analysis_data$Time_Period_Start<-as.Date(main_analysis_data$Time_Period_Start)
names(main_analysis_data)[which(colnames(main_analysis_data) == "sum_deaths")] <- "imputed_deaths"

################################## set up the Regions ##############################
#set up the regions according to Census: https://www.census.gov/geographies/reference-maps/2010/geo/201
NE.name <- c("Connecticut","Maine","Massachusetts","New Hampshire",
             "Rhode Island","Vermont","New Jersey","New York",
             "Pennsylvania")

MW.name <- c("Indiana","Illinois","Michigan","Ohio","Wisconsin",
             "Iowa","Kansas","Minnesota","Missouri","Nebraska",
             "North Dakota","South Dakota")

S.name <- c("Delaware","District of Columbia","Florida","Georgia",
            "Maryland","North Carolina","South Carolina","Virginia",
            "West Virginia","Alabama","Kentucky","Mississippi",
            "Tennessee","Arkansas","Louisiana","Oklahoma","Texas")

W.name <- c("Arizona","Colorado","Idaho","New Mexico","Montana",
            "Utah","Nevada","Wyoming","Alaska","California",
            "Hawaii","Oregon","Washington")
```

```
region.list <- list(
  Northeast=NE.name,
  Midwest=MW.name,
  South=S.name,
  West=W.name)

#initialize vector with "West" and then impute the other regions for the states
main_analysis_data$Region<-rep("West", nrow(main_analysis_data))
for(state in unique(main_analysis_data$State)){
  if(state %in% region.list$Northeast){
    main_analysis_data$Region[main_analysis_data$State == state]<-"Northeast"
  }else if(state %in% region.list$Midwest){
    main_analysis_data$Region[main_analysis_data$State == state]<-"Midwest"
  }else if(state %in% region.list$South){
    main_analysis_data$Region[main_analysis_data$State == state]<-"South"
  }
}
```

2 Exploratory Data Analysis

2.1 Overdose Deaths

```
############################## EDA: Plot the Outcome and Intervention Trends ###########################
#plot the time series of the number of deaths and probability of overdose death
od_data_recent <- read.csv("./Data/unintentional_od_yearly_1999_2019_17_up.txt", sep = "\t", stringsAsF
od_data_recent$Deaths <- as.numeric(od_data_recent$Deaths)
od_data_recent<-od_data_recent[!is.na(od_data_recent$Year),] #delete the rows that just contains data s
od_data_recent<- od_data_recent %>% filter(Year > 1999 & Year < 2020) %>% group_by(Year) %>%
  summarise(sum_deaths = sum(Deaths, na.rm = TRUE))

# pdf("./Figures/total_od_deaths_all_paper_9_6_21_2000_2019.pdf")
ggplot(data = od_data_recent, mapping = aes(x = Year, y = sum_deaths)) +
  geom_line() + geom_point() +
  labs(x = "Year", y = "Yearly Number of Unintentional Drug Overdose Deaths in the 50 U.S. States") +
  theme(panel.background = element_rect("white"), panel.border = element_blank(), panel.grid.major = ele
        panel.grid.minor = element_blank(), axis.line = element_line(colour = "black"),
        axis.title=element_text(family="Times", size=10, face="bold"),
        axis.text=element_text(family="Times",size=10)) +
  scale_x_continuous(breaks = seq(2000, 2020, by = 2)) +
  ylim(c(0, 62000))
```




```
# dev.off()

main_analysis_data_sum <- main_analysis_data %>% group_by(year = year(Time_Period_Start)) %>%
  summarise(total_deaths = sum(imputed_deaths),
            sum_pop = sum(population)/2,
            total_prop = sum(imputed_deaths)/(sum(population)/2),
            total_prop_by_100000 = 100000*sum(imputed_deaths)/(sum(population)/2))
# %>%mutate(date = as.Date(as.yearmon(year)))

#compute the percentage difference between 2000 and 2019
death_2000 <- main_analysis_data_sum$total_deaths[main_analysis_data_sum$year == 2000]
death_2019 <- main_analysis_data_sum$total_deaths[main_analysis_data_sum$year == 2019]

((death_2019 - death_2000)/death_2000)*100
```

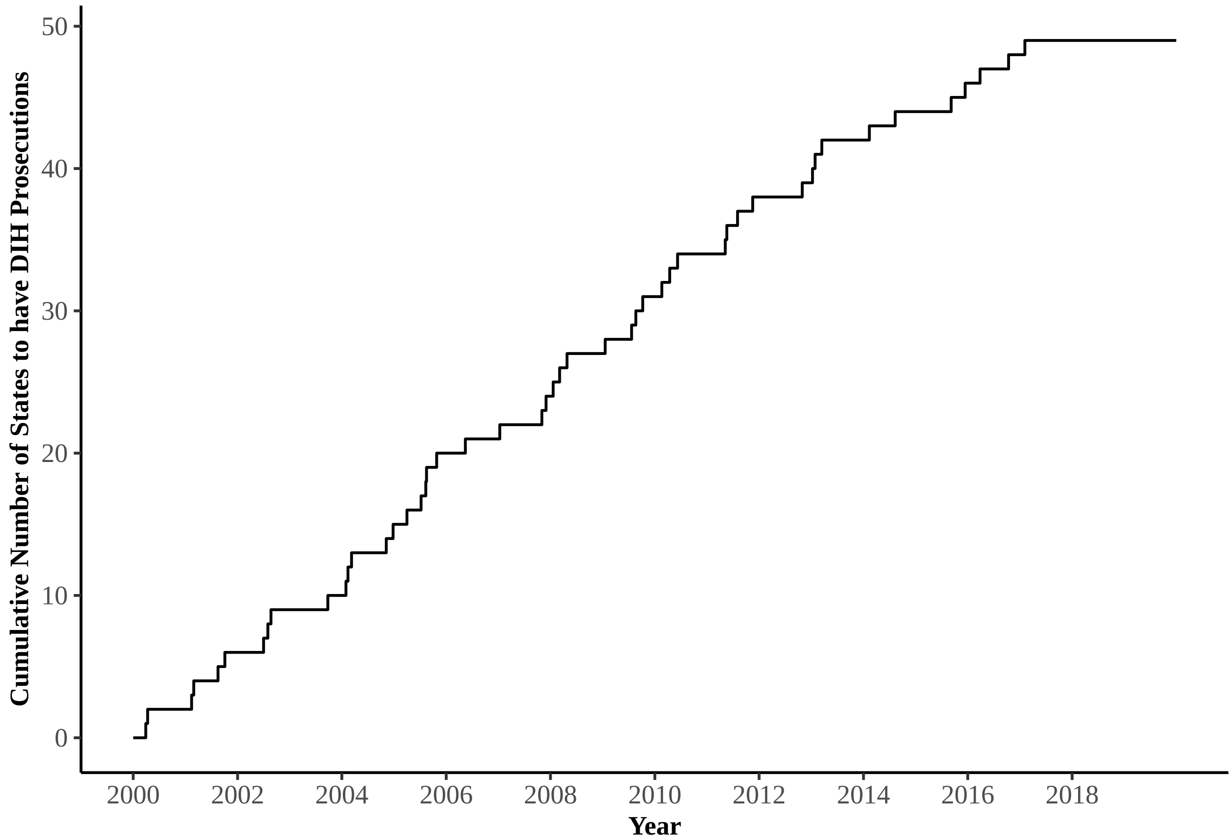
```
## [1] 435.5701
```

2.2 Intervention: DIH Prosecutions

```
#plot the number of states with an intervention for each time point
#first, create a data set to find the number of states with an intervention at each time point
#initialize the data set with the start date of the time period
num_states_with_intervention<-data.frame("Start_Date" = unique((main_analysis_data$Intervention_First_Da
```

```r
numStates<-c()

#for each time period i, we first find the states where the first intervention date occurred before i
#then, we append it to numStates
for(i in unique((num_states_with_intervention$Start_Date))){
  states_w_int<-unique(main_analysis_data$State[(main_analysis_data$Intervention_First_Date)<=i])
  numStates<-append(numStates, length(states_w_int[!is.na(states_w_int)]))
}
num_states_with_intervention$numStates<-numStates
num_states_with_intervention$Start_Date <- as.Date(num_states_with_intervention$Start_Date)
num_states_with_intervention <- rbind(data.frame("Start_Date" = c(as.Date("2000-01-01"),
                                                                  as.Date("2019-12-31")),
                                                 "numStates" = c(0, max(num_states_with_intervention$nu
                                      num_states_with_intervention)
num_states_with_intervention <- num_states_with_intervention %>% arrange(Start_Date) %>%
  mutate(lag_numStates = lag(numStates))

num_states_with_intervention <- num_states_with_intervention %>%
  pivot_longer( c("lag_numStates", "numStates"), "numStates")

# pdf("Figures/num_states_with_intervention_9_6_21.pdf")
ggplot(num_states_with_intervention, aes(x = Start_Date, y = value, group = 1)) +
  geom_line() +
  # geom_point(num_states_with_intervention[num_states_with_intervention$numStates == "numStates",],
  #            mapping = aes(x = Start_Date, y = value, group = 1), size = 1) +
  labs(x = "Year", y = "Cumulative Number of States to have DIH Prosecutions") +
  theme(axis.text=element_text(family="Times",size=10),
        axis.title=element_text(family="Times", size=10, face="bold"),
        panel.border = element_blank(), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(), axis.line = element_line(colour = "black"),
        axis.text.x = element_text(family="Times", size=10),
        panel.background = element_rect("white")) +
  scale_x_date(date_labels="%Y", breaks = seq(as.Date("2000-01-01"), as.Date("2018-01-01"), by = "2 yea
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```




```
# dev.off()
```

2.3 Policy Dates

```
#add the intervention variable as a measure of number of states with DIH prosecution
main_analysis_data <- main_analysis_data %>%
  group_by(Time_Period_Start) %>%
  mutate(num_states_w_intervention = sum(Intervention_Redefined))

############################## Look at the policy dates ######################
policy_dates <- main_analysis_data %>% group_by(State) %>%
  summarise(unique(format(Intervention_First_Date, "%Y-%m")),
            unique(format(as.Date(Naloxone_Pharmacy_Yes_First_Date), "%Y-%m")),
            unique(format(as.Date(Naloxone_Pharmacy_No_First_Date), "%Y-%m")),
            unique(format(as.Date(Medical_Marijuana_First_Date), "%Y-%m")),
            unique(format(as.Date(Recreational_Marijuana_First_Date), "%Y-%m")),
            unique(format(as.Date(PDMP_First_Date), "%Y-%m")),
            unique(format(as.Date(GSL_First_Date), "%Y-%m")),
            unique(format(as.Date(Medicaid_Expansion_First_Date), "%Y-%m")))
names(policy_dates) <- c("State", "DIH Prosecutions", "NAL: Pharmacists Yes",
                         "NAL: Pharmacists No", "MML", "RML", "PDMP", "GSL",
                         "Medicaid")
# write.csv(policy_dates, "./Data/policy_dates_9_6_21.csv")
```

3 Main Analysis: At Least One DIH Prosecution Report in Media

3.1 Analysis

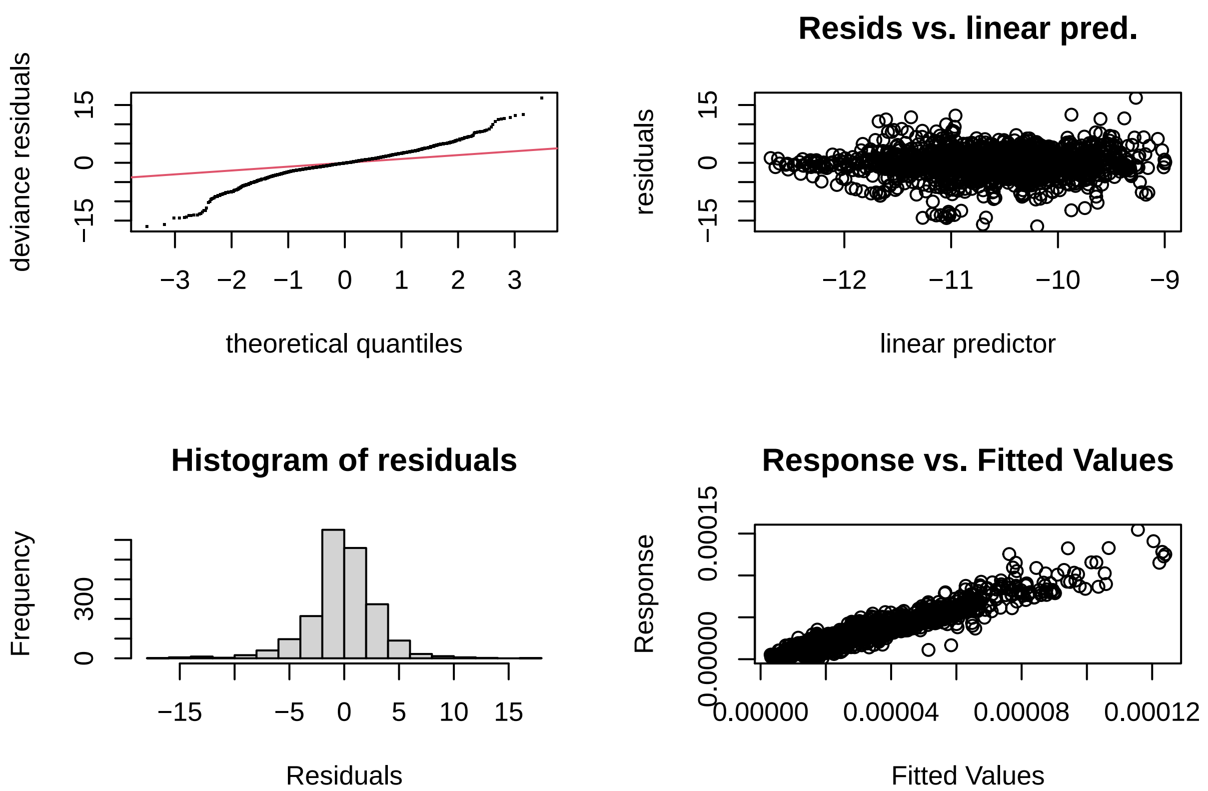
```
############################## Run Model with Spline Time Effects by Region ##############################
#model that we will be using for the main analysis
#cr is used for cubic regression spline -- we are smoothing time effects by region
#run the analysis for all the states
main_analysis_model<-gam(cbind(round(imputed_deaths), round(num_alive))~ State +
                           s(Time_Period_ID, bs = "cr", by = as.factor(Region)) +
                           Naloxone_Pharmacy_Yes_Redefined +
                           Naloxone_Pharmacy_No_Redefined +
                           Medical_Marijuana_Redefined +
                           Recreational_Marijuana_Redefined +
                           GSL_Redefined +
                           PDMP_Redefined +
                           Medicaid_Expansion_Redefined +
                           Intervention_Redefined +
                           num_states_w_intervention,
                         data = main_analysis_data, family = "binomial")

#summary output of the model
stargazer(main_analysis_model)
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Sun, Sep 12, 2021 - 17:50:36

```
gam.check(main_analysis_model)
```

Table 1:

	Dependent variable:
	cbind(round(imputed_deaths), round(num_alive))
StateAlaska	0.250***
	(0.028)
StateArizona	0.305***
	(0.014)
StateArkansas	−0.396***
	(0.020)
StateCalifornia	−0.170***
	(0.013)
StateColorado	0.092***
	(0.016)
StateConnecticut	0.192***
	(0.016)
StateDelaware	0.438***
	(0.022)
StateFlorida	0.268***
	(0.012)
StateGeorgia	−0.087***
	(0.013)
StateHawaii	−0.217***
	(0.026)
StateIdaho	−0.159***
	(0.024)
StateIllinois	−0.022*
	(0.013)
StateIndiana	0.079***
	(0.014)
StateIowa	−0.745***
	(0.021)
StateKansas	−0.343***
	(0.019)
StateKentucky	0.641***
	(0.014)
StateLouisiana	0.283***
	(0.014)
StateMaine	0.167***
	(0.022)
StateMaryland	−1.069***



Method: UBRE Optimizer: outer newton full convergence after 6 iterations. Gradient range [-1.448796e-06,6.372085e-05] (score 8.805847 & scale 1). Hessian positive definite, eigenvalue range [0.0001512652,0.0003395804]. Model rank = 95 / 95

Basis dimension (k) checking results. Low p-value (k-index<1) may indicate that k is too low, especially if edf is close to k'.

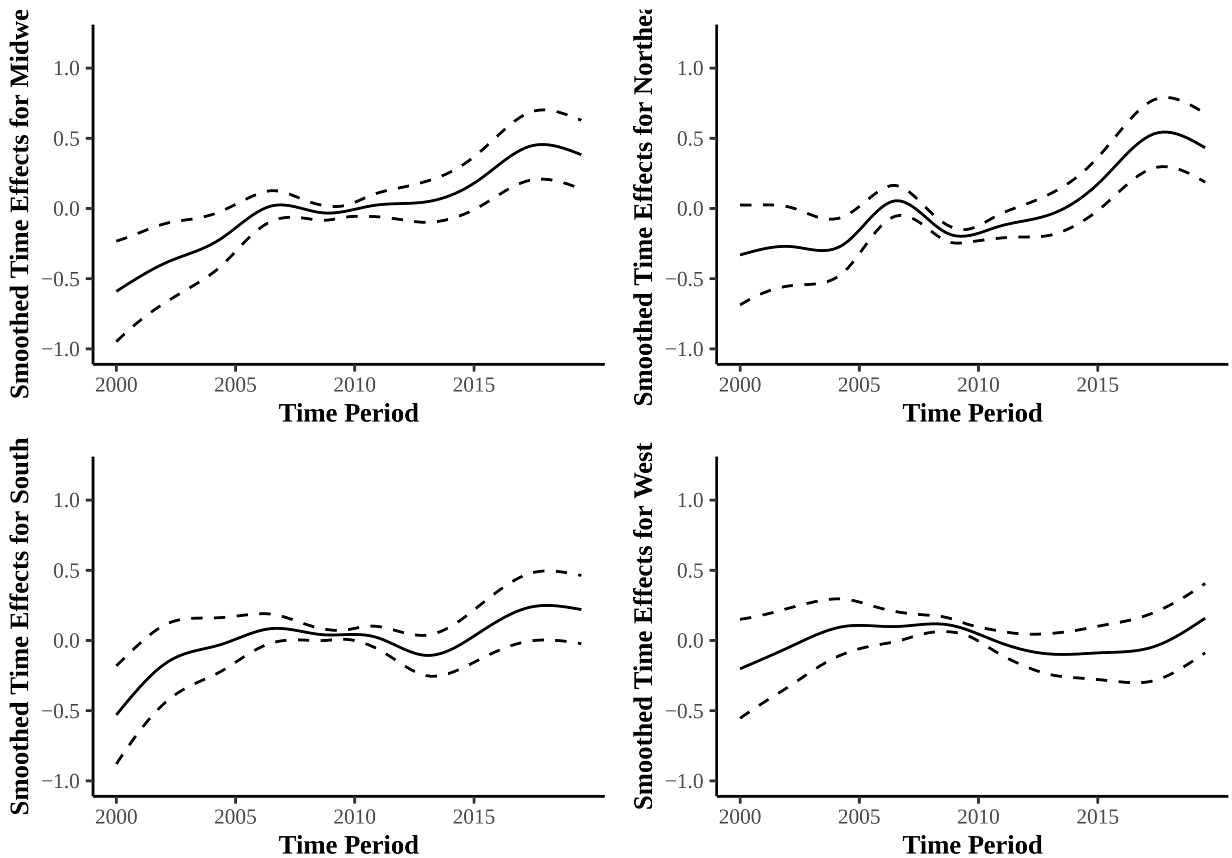
```
k'   edf k-index p-value
```

s(Time_Period_ID):as.factor(Region)Midwest 9.00 8.86 1.05 0.99 s(Time_Period_ID):as.factor(Region)Northeast 9.00 8.93 1.05 0.98 s(Time_Period_ID):as.factor(Region)South 9.00 8.90 1.05 0.99 s(Time_Period_ID):as.factor(Region)West 9.00 8.57 1.05 0.96

3.2 Plots

```
main_analysis_model_object <- getViz(main_analysis_model)

midwest_plot <- plot(sm(main_analysis_model_object, 1)) +
  l_fitLine() +
  l_ciLine(mul = 5, linetype = 2)  + theme_classic() +
  labs(x = "Time Period", y = "Smoothed Time Effects for Midwest") +
  scale_x_continuous(breaks=c(1,11,21,31), labels=c("2000", "2005",
                                                    "2010", "2015"))  +
  theme(text=element_text(family="Times",size=10),
        title = element_text(family="Times", size=10, face = "bold"),
```

```r
        panel.background = element_rect("white")) +
  ylim(c(-1,1.2))

northeast_plot <- plot(sm(main_analysis_model_object,2)) +
  l_fitLine() +
  l_ciLine(mul = 5, linetype = 2)  + theme_classic() +
  labs(x = "Time Period", y = "Smoothed Time Effects for Northeast") +
  scale_x_continuous(breaks=c(1,11,21,31), labels=c("2000", "2005",
                                                    "2010", "2015"))+
  theme(text=element_text(family="Times",size=10),
        title = element_text(family="Times", size=10, face = "bold"),
        panel.background = element_rect("white")) +
  ylim(c(-1,1.2))

south_plot <- plot(sm(main_analysis_model_object, 3)) +
  l_fitLine() +
  l_ciLine(mul = 5, linetype = 2)  + theme_classic() +
  labs(x = "Time Period", y = "Smoothed Time Effects for South") +
  scale_x_continuous(breaks=c(1,11,21,31), labels=c("2000", "2005",
                                                    "2010", "2015"))+
  theme(text=element_text(family="Times",size=10),
        title = element_text(family="Times", size=10, face = "bold"),
        panel.background = element_rect("white")) +
  ylim(c(-1,1.2))

west_plot <- plot(sm(main_analysis_model_object, 4)) +
  l_fitLine() +
  l_ciLine(mul = 5, linetype = 2)  + theme_classic() +
  labs(x = "Time Period", y = "Smoothed Time Effects for West") +
  scale_x_continuous(breaks=c(1,11,21,31), labels=c("2000", "2005",
                                                    "2010", "2015"))+
  theme(text=element_text(family="Times",size=10),
        title = element_text(family="Times", size=10, face = "bold"),
        panel.background = element_rect("white")) +
  ylim(c(-1,1.2))

# pdf("./Figures/time_smoothed_effects_9_6_21.pdf")
gridPrint(midwest_plot, northeast_plot, south_plot, west_plot, ncol = 2)
```




```r
# dev.off()

total_pop <- main_analysis_data %>%
  group_by(year = year(Time_Period_Start), State) %>%
  summarise(pop = unique(population)) %>%
  group_by(year) %>%
  summarise(sum(pop))
```

```
## `summarise()` has grouped output by 'year'. You can override using the `.groups` argument.
```

```r
main_analysis_data %>%
  group_by(year(Time_Period_Start)) %>%
  summarise(sum_deaths = sum(imputed_deaths)*100000) %>%
  mutate(sum_deaths/total_pop$`sum(pop)`)
```

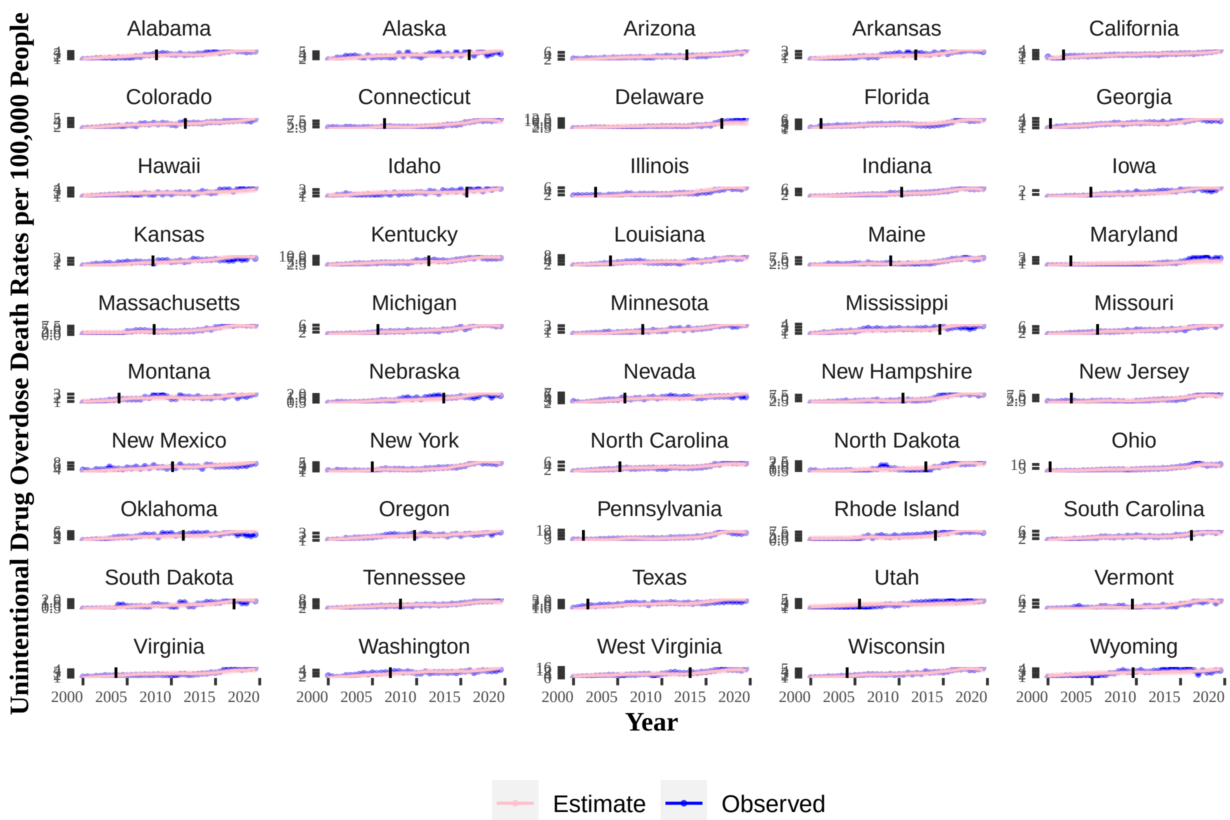
```
## # A tibble: 20 x 3
##    `year(Time_Period_Start)`  sum_deaths `sum_deaths/total_pop$\`sum(pop)\``
##  *                     <dbl>       <dbl>                               <dbl>
##  1                      2000  1151390000                                2.35
##  2                      2001  1276465000                                2.57
##  3                      2002  1614890000                                3.22
##  4                      2003  1799140000.                               3.55
##  5                      2004  1953250000                                3.82
##  6                      2005  2216225000                                4.29
```

```
## 7                     2006 2603525000                                    4.99
## 8                     2007 2730800000                                    5.18
## 9                     2008 2787500000                                    5.23
## 10                    2009 2848100000                                    5.29
## 11                    2010 2969200000                                    5.48
## 12                    2011 3276800000                                    6.01
## 13                    2012 3291600000                                    5.98
## 14                    2013 3541000000                                    6.38
## 15                    2014 3847600000                                    6.88
## 16                    2015 4381900000                                    7.77
## 17                    2016 5433100000                                    9.56
## 18                    2017 6084700000                                   10.6
## 19                    2018 5847900000                                   10.1
## 20                    2019 6166500000                                   10.6
```

```
# main_analysis_data %>%
#   group_by(State) %>%
#   summarise(min_death_rate = min(imputed_deaths/population*100000),
#             max_death_rate = max(imputed_deaths/population*100000)) %>%
#   mutate(range_death_rate = max_death_rate - min_death_rate) %>% View()
#
# #summarize the DIH dates
# main_analysis_data %>%
#   group_by(Time_Period_Start) %>%
#   summarise(prop_w_intervention = mean(Intervention_Redefined > 0)) %>%
#   View()

#create a data frame to store the results and compute the confidence intervals
#initialize the columns
main_analysis_plot_table<-data.frame(State = main_analysis_data$State)
main_analysis_plot_table$Fitted<-rep(NA, nrow(main_analysis_plot_table))
main_analysis_plot_table$Observed<-rep(NA, nrow(main_analysis_plot_table))
main_analysis_plot_table$Time<-main_analysis_data$Time_Period_ID
main_analysis_plot_table$Time_Date<-main_analysis_data$Time_Period_Start
main_analysis_plot_table$Intervention_Date<-main_analysis_data$Intervention_First_Date

#we want to compare the fitted probability of overdose death and the observed values to see how the mod
for(i in unique(main_analysis_plot_table$State)){
  #for each state, we first subset the main analysis data to only look at the data for that state
  index_of_state<-which(main_analysis_plot_table$State == i)
  #impute the fitted and observed probability of overdose death for the state
  main_analysis_plot_table$Fitted[index_of_state]<-fitted(main_analysis_model)[index_of_state]
  main_analysis_plot_table$Observed[index_of_state] <- (main_analysis_data$imputed_deaths[main_analysis_
}

#plot to compare the fitted values vs observed deaths
# pdf("./Figures/GAM_fitted_vs_actual_by_Region_9_6_21_with_int_date_full_data.pdf")
ggplot(data = main_analysis_plot_table, aes(x = Time_Date, y = Observed*100000, group = 1,
                                            color = "Observed")) +
  geom_line(aes(color = "Observed"))+ geom_point(aes(color = "Observed"), size = .5, alpha = .5) +
  geom_line(data = main_analysis_plot_table, aes(x = Time_Date, y = Fitted*100000, group = 1,
                                                 color = "Estimate")) +
  geom_point(data = main_analysis_plot_table, aes(x = Time_Date, y = Fitted*100000,
                                                  color = "Estimate"),
```

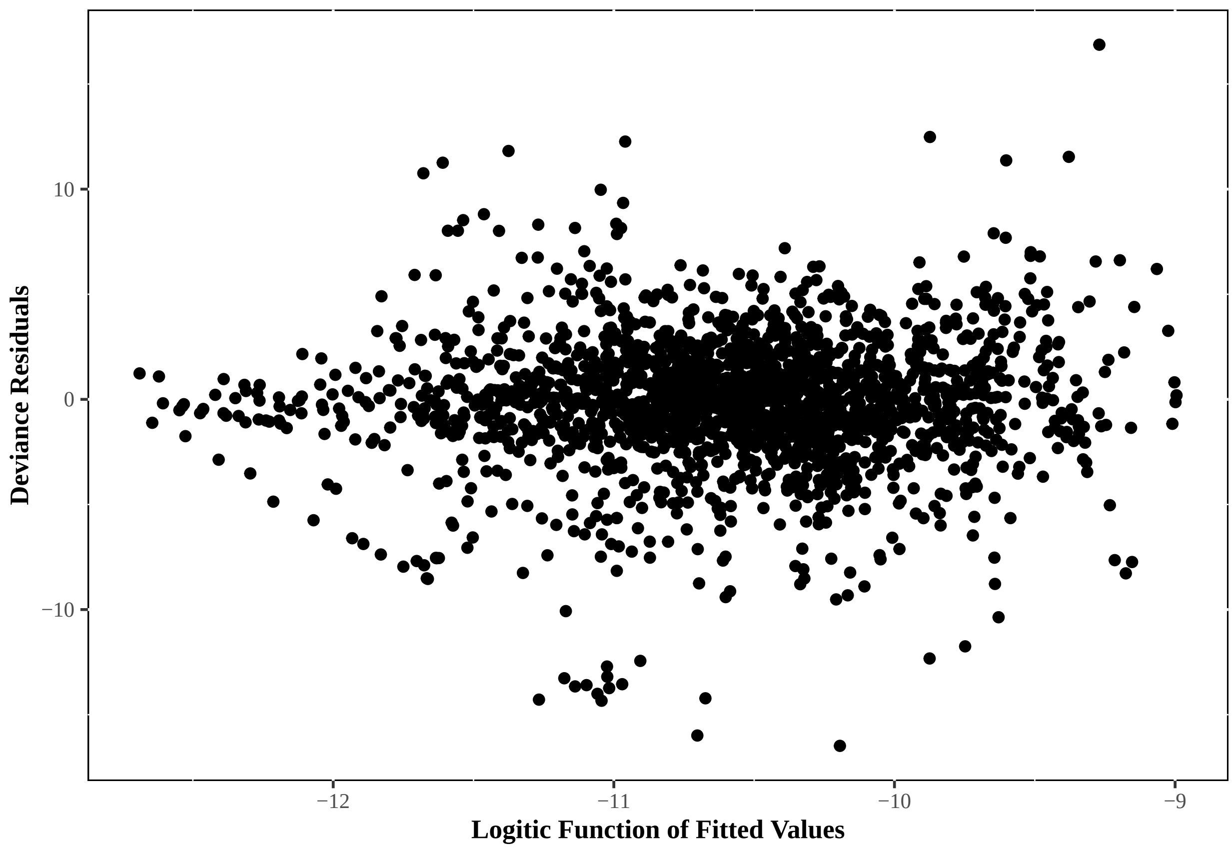
```r
            size = .5, alpha = .5) +
  scale_color_manual(values = c("pink", "blue")) +
  geom_vline(main_analysis_plot_table, mapping = aes(xintercept = Intervention_Date)) +
  facet_wrap(facets = vars(State), scales = "free_y", ncol = 5) +
  theme(axis.text.x = element_text(hjust = 1, size = 6, family = "Times"),
        axis.text.y = element_text(size = 6, family = "Times"),
        axis.title=element_text(size = 10,face = "bold", family = "Times"),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        strip.background = element_blank(),
        strip.text = element_text(size=8),
        panel.background = element_rect("white"),
        legend.position = "bottom") +
  labs(x = "Year", y = "Unintentional Drug Overdose Death Rates per 100,000 People",
       color = "")
```

```
## Warning: Removed 40 rows containing missing values (geom_vline).
```

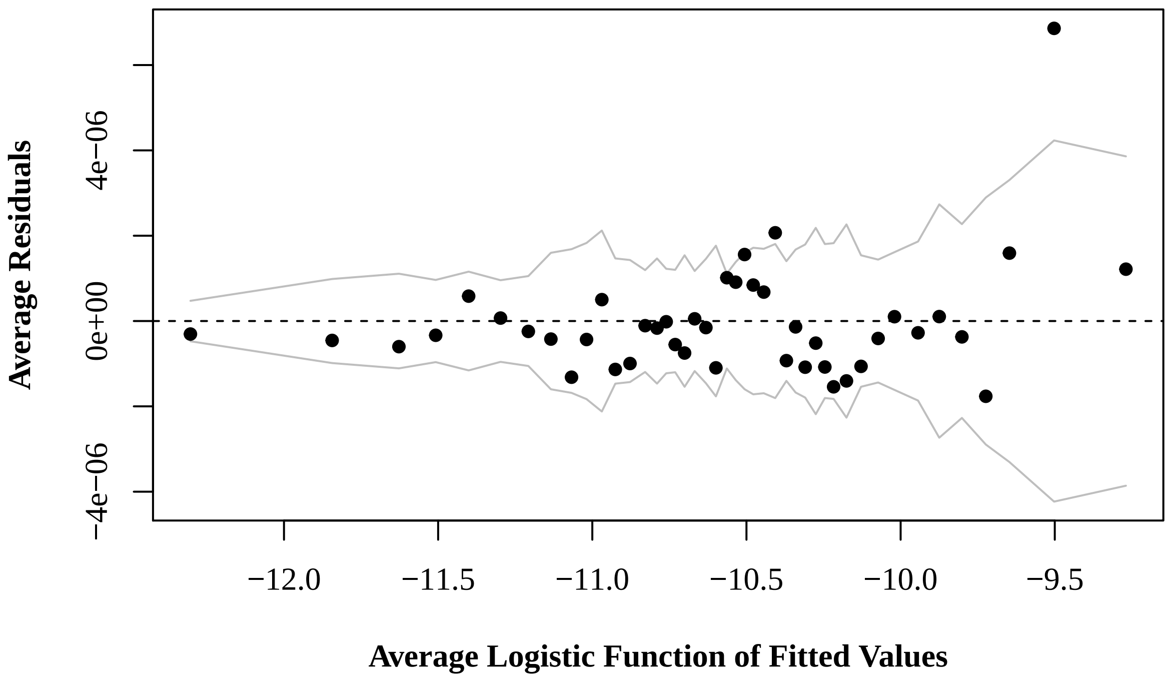



```r
# dev.off()

#diagnostic plots to check model
residTab <- data.frame(logit_fitted_vals = logit(fitted(main_analysis_model)),
                       resids = residuals(main_analysis_model))
# pdf("./Figures/deviance_resids_9_6_21.pdf")
```

```r
ggplot(residTab, aes(x = logit_fitted_vals, y = resids)) +
  geom_point() +
  theme(text = element_text(size = 10, family = "Times"),
        title = element_text(size = 10, family = "Times", face = "bold"),
        panel.background = element_rect(fill = "white", color = "black")) +
  # theme_classic() +
  labs(x = "Logitic Function of Fitted Values", y = "Deviance Residuals")
```




```r
# dev.off()

pred_vals <- predict(main_analysis_model)
resids <- resid(main_analysis_model, type = "response")
# pdf("./Figures/binned_resids_plot_9_6_21.pdf")
par(font.lab = 2)
par(family = "Times")
binnedplot(pred_vals, resids, main = "", xlab = "Average Logistic Function of Fitted Values",
           ylab = "Average Residuals")
```




```
# dev.off()
```

3.3 Compile Results

```
############################## Main Analysis: Make Data Frame of Results and 95% CI ###################
#store the coefficients into the table
main_analysis_full_table<-data.frame(coef(main_analysis_model))
#check to see how the table looks
head(main_analysis_full_table)
```

```
##                   coef.main_analysis_model.
## (Intercept)                    -11.05928955
## StateAlaska                      0.25032740
## StateArizona                     0.30457546
## StateArkansas                   -0.39568637
## StateCalifornia                 -0.17047311
## StateColorado                    0.09249597
```

```
#rename the column to "Coefficient_Estimate"
colnames(main_analysis_full_table)<-c("Coefficient_Estimate")

#vector of covariates
covariates<-c("Naloxone_Pharmacy_Yes_Redefined", "Naloxone_Pharmacy_No_Redefined",
```

```r
              "Medical_Marijuana_Redefined",
              "Recreational_Marijuana_Redefined",
              "GSL_Redefined", "PDMP_Redefined",
              "Medicaid_Expansion_Redefined", "Intervention_Redefined",
              "num_states_w_intervention")

#rename the variable names of the regression output so that they look nicer:
#currently there are 3 types of coefficients: state effects, the covariates, and smoothed time effects
#for each row in the main analysis table
for(i in 1:length(rownames(main_analysis_full_table))){

  #if the coefficient is not in the covariates vector
  if(!(rownames(main_analysis_full_table)[i] %in% covariates)){

    #we see if it's a state effect
    if(substr(rownames(main_analysis_full_table)[i], start = 1, stop = 5) == "State"){

      #if so, here, the names look like: StateMassachusetts or StateGeorgia, so take out the "State" pa
      #and just rename these rows to just the state name
      rownames(main_analysis_full_table)[i]<-substr(rownames(main_analysis_full_table)[i], start = 6,
                                             stop = nchar(rownames(main_analysis_full_table)[i]))

    }else if(rownames(main_analysis_full_table)[i] == "(Intercept)"){

      #otherwise, if the current name is Intercept, we rename it so that we know that Alabama is the ba
      rownames(main_analysis_full_table)[i]<-"Intercept/Alabama"

    }else if(substr(rownames(main_analysis_full_table)[i], start = 1, stop = 35) == "s(Time_Period_ID):a

      #otherwise, it's the smoothed time effects which look like: s(Time_Period_ID):as.factor(Region)We
      #or s(Time_Period_ID):as.factor(Region)South, so we want to get rid of "s(Time_Period_ID):as.fact
      #and change it to "Smoothed Time for Region"
      rownames(main_analysis_full_table)[i]<-paste("Smoothed Time for Region ",
                                                   substr(rownames(main_analysis_full_table)[i], start =
                                                          stop = nchar(rownames(main_analysis_full_tabl
                                                   sep = "")

    }
  }
}

#confidence intervals for the coefficients
main_analysis_full_table$Coefficient_Lower_Bound<-main_analysis_full_table$Coefficient_Estimate - 1.96*s
main_analysis_full_table$Coefficient_Upper_Bound<-main_analysis_full_table$Coefficient_Estimate + 1.96*s

#impute the estimates and confidence intervals in the odds ratio scale
main_analysis_full_table$Odds_Ratio<-exp(main_analysis_full_table$Coefficient_Estimate)
main_analysis_full_table$Odds_Ratio_LB<-exp(main_analysis_full_table$Coefficient_Lower_Bound)
main_analysis_full_table$Odds_Ratio_UB<-exp(main_analysis_full_table$Coefficient_Upper_Bound)

#store the standard error and p-value
main_analysis_full_table$Standard_Error<-summary(main_analysis_model)$se
#note that there is no p-value for the smoothed time effects, so we put a NA for those rows
```

```r
main_analysis_full_table$p_value<-c(summary(main_analysis_model)$p.pv, rep(NA, length(coef(main_analysis

head(main_analysis_full_table)
```

```
##                    Coefficient_Estimate Coefficient_Lower_Bound
## Intercept/Alabama          -11.05928955            -11.19842493
## Alaska                       0.25032740              0.19627234
## Arizona                      0.30457546              0.27724686
## Arkansas                    -0.39568637             -0.43420110
## California                  -0.17047311             -0.19647490
## Colorado                     0.09249597              0.06106175
##                    Coefficient_Upper_Bound   Odds_Ratio Odds_Ratio_LB
## Intercept/Alabama              -10.9201542 1.574025e-05  1.369575e-05
## Alaska                           0.3043825 1.284446e+00  1.216858e+00
## Arizona                          0.3319041 1.356049e+00  1.319492e+00
## Arkansas                        -0.3571716 6.732178e-01  6.477820e-01
## California                      -0.1444713 8.432658e-01  8.216220e-01
## Colorado                         0.1239302 1.096909e+00  1.062965e+00
##                    Odds_Ratio_UB Standard_Error       p_value
## Intercept/Alabama   1.808995e-05     0.07098744  0.000000e+00
## Alaska              1.355787e+00     0.02757911  1.119133e-19
## Arizona             1.393619e+00     0.01394317 8.851746e-106
## Arkansas            6.996524e-01     0.01965037  3.546730e-90
## California          8.654797e-01     0.01326622  8.583035e-38
## Colorado            1.131937e+00     0.01603787  8.052817e-09
```

```r
tail(main_analysis_full_table)
```

```
##                                  Coefficient_Estimate Coefficient_Lower_Bound
## Smoothed Time for Region West.4            0.17071662              0.13483168
## Smoothed Time for Region West.5            0.04607653             -0.02157082
## Smoothed Time for Region West.6           -0.03846207             -0.13394493
## Smoothed Time for Region West.7           -0.03151909             -0.14422243
## Smoothed Time for Region West.8            0.01989377             -0.11800391
## Smoothed Time for Region West.9            0.18842096              0.07222067
##                                  Coefficient_Upper_Bound Odds_Ratio
## Smoothed Time for Region West.4               0.20660157  1.1861546
## Smoothed Time for Region West.5               0.11372387  1.0471545
## Smoothed Time for Region West.6               0.05702078  0.9622682
## Smoothed Time for Region West.7               0.08118425  0.9689725
## Smoothed Time for Region West.8               0.15779145  1.0200930
## Smoothed Time for Region West.9               0.30462125  1.2073417
##                                  Odds_Ratio_LB Odds_Ratio_UB Standard_Error
## Smoothed Time for Region West.4      1.1443441      1.229493     0.01830865
## Smoothed Time for Region West.5      0.9786602      1.120443     0.03451395
## Smoothed Time for Region West.6      0.8746382      1.058678     0.04871574
## Smoothed Time for Region West.7      0.8656952      1.084571     0.05750171
## Smoothed Time for Region West.8      0.8886926      1.170922     0.07035596
## Smoothed Time for Region West.9      1.0748925      1.356111     0.05928586
##                                  p_value
## Smoothed Time for Region West.4       NA
## Smoothed Time for Region West.5       NA
```

```
## Smoothed Time for Region West.6      NA
## Smoothed Time for Region West.7      NA
## Smoothed Time for Region West.8      NA
## Smoothed Time for Region West.9      NA
```

```
#save the table into a CSV
# write.csv(round(main_analysis_full_table,5), "./Data/coefficients_GAM_9_6_21_full_data_uninentional_o

#export a table with just the covariates
#first, find the rows that contains the covariates
covariate_Index<-which(rownames(main_analysis_full_table) %in% covariates)
main_analysis_covariate_table<-(round(main_analysis_full_table[covariate_Index,], 5))

#rename the variables so that it looks cleaner
rownames(main_analysis_covariate_table)<-c("Naloxone_Pharmacy_Yes", "Naloxone_Pharmacy_No",
                                           "Medical_Marijuana",
                                           "Recreational_Marijuana",
                                           "GSL", "PDMP", "Medicaid_Expansion",
                                           "Intervention", "Total Number with DIH Prosecutions")

#now, reorganize the data so that the covariates are on top and the rest of the variable sare below
main_analysis_covariate_table<-rbind(main_analysis_covariate_table, main_analysis_full_table[-covariate_
#remove the columns that aren't in odds ratio scale
main_analysis_covariate_table<-main_analysis_covariate_table[,-which(colnames(main_analysis_covariate_ta
                                                                      c("Coefficient_Estimate", "Coeffi

colnames(main_analysis_covariate_table)<-c("Risk_Ratio_Estimates", "RR_95_CI_LB", "RR_95_CI_UB", "p-valu
head(main_analysis_covariate_table, 10)
```

```
##                                    Risk_Ratio_Estimates  RR_95_CI_LB
## Naloxone_Pharmacy_Yes                      9.749800e-01 9.602700e-01
## Naloxone_Pharmacy_No                       1.006720e+00 9.939600e-01
## Medical_Marijuana                          1.065370e+00 1.053630e+00
## Recreational_Marijuana                     9.637100e-01 9.475300e-01
## GSL                                        1.036210e+00 1.024840e+00
## PDMP                                       9.793700e-01 9.682900e-01
## Medicaid_Expansion                         1.105550e+00 1.092500e+00
## Intervention                               1.070800e+00 1.059460e+00
## Total Number with DIH Prosecutions         1.014340e+00 1.009550e+00
## Intercept/Alabama                          1.574025e-05 1.369575e-05
##                                     RR_95_CI_UB p-value
## Naloxone_Pharmacy_Yes              9.899100e-01 0.00108
## Naloxone_Pharmacy_No               1.019660e+00 0.30344
## Medical_Marijuana                  1.077240e+00 0.00000
## Recreational_Marijuana             9.801500e-01 0.00002
## GSL                                1.047710e+00 0.00000
## PDMP                               9.905900e-01 0.00033
## Medicaid_Expansion                 1.118750e+00 0.00000
## Intervention                       1.082260e+00 0.00000
## Total Number with DIH Prosecutions 1.019150e+00 0.00000
## Intercept/Alabama                  1.808995e-05 0.00000
```

```r
#save the table into a CSV
# write.csv(round(main_analysis_covariate_table, 3), "./Data/coefficients_covariates_9_6_21_full_data_u
```

3.4 Attributable Deaths

```r
############################## Main Analysis: Number of Overdose Deaths Attributed to Intervention ####
#find the number of deaths attributable to the intervention
#first, we subset the data so that we only focus on the time points for which at least one state had th
attr_deaths_anlys_main_analysis<-main_analysis_data[which(main_analysis_data$Intervention_Redefined>0),]

#compute the probability of overdose had intervention not occurred
prob_od_no_int_main_analysis<-expit(-coef(main_analysis_model)["Intervention_Redefined"]*attr_deaths_an
                                    + logit(attr_deaths_anlys_main_analysis$imputed_deaths/attr_deaths_

#compute the lower and upper bounds of 95% CI of probability of overdose had intervention not occurred
#here, we compute the lower and upper bounds of the 95% CI of all the coefficients using the standard e
coef_lb<-coef(main_analysis_model) - 1.96*summary(main_analysis_model)$se
coef_ub<-coef(main_analysis_model) + 1.96*summary(main_analysis_model)$se

#we then calculate the upper and lower bounds of the probability of overdose death had intervention not
#the lower and upper bounds of the coefficient of the intervention variable
prob_od_no_int_LB_main_analysis<-expit(-coef_lb[names(coef_lb) == "Intervention_Redefined"]*attr_deaths
                                       + logit(attr_deaths_anlys_main_analysis$imputed_deaths/attr_deat

prob_od_no_int_UB_main_analysis<-expit(-coef_ub[names(coef_ub) == "Intervention_Redefined"]*attr_deaths
                                       + logit(attr_deaths_anlys_main_analysis$imputed_deaths/attr_deat


#estimate the number of deaths attributable to the intervention
#first, initialize the vectors to store the numbers
num_attr_od_UB<-num_attr_od_LB<-num_attr_od<-rep(NA, length(unique(attr_deaths_anlys_main_analysis$Time


#for each time period, we first find the indices of rows containing data from that time point
#then, we find the total number of deaths that attributable to the intervention

index<-1 #keep track of where to store the values in the vector

for(time in sort(unique(attr_deaths_anlys_main_analysis$Time_Period_ID))){
  #find the indices of rows where the time point = time
  time_point_index<-which(attr_deaths_anlys_main_analysis$Time_Period_ID == time)

  #find the number of deaths attributable to intervention = observed number of deaths with intervention
  num_attr_od[index]<-sum(attr_deaths_anlys_main_analysis$imputed_deaths[time_point_index]
                          - prob_od_no_int_main_analysis[time_point_index]*attr_deaths_anlys_main_analy
  #find the lower and upper bounds of the estimated number of deaths attributable to the intervention
  num_attr_od_LB[index]<-sum(attr_deaths_anlys_main_analysis$imputed_deaths[time_point_index]
                             - prob_od_no_int_LB_main_analysis[time_point_index]*attr_deaths_anlys_main
  num_attr_od_UB[index]<-sum(attr_deaths_anlys_main_analysis$imputed_deaths[time_point_index]
                             - prob_od_no_int_UB_main_analysis[time_point_index]*attr_deaths_anlys_main
  index<-index + 1
```

```
}

num_attr_od_main_analysis<-data.frame("Time_Period_ID" = sort(unique(attr_deaths_anlys_main_analysis$Tim
                                      "Time_Start" = sort(unique(attr_deaths_anlys_main_analysis$Time_Pe
                                      "Num_Attr_Deaths" = num_attr_od,
                                      "Num_Attr_Deaths_LB" = num_attr_od_LB,
                                      "Num_Attr_Deaths_UB" = num_attr_od_UB)

#sum up the total number of excess deaths attributable to the intervention
sum(num_attr_od_main_analysis$Num_Attr_Deaths)
```

[1] 36888.23

```
stargazer(num_attr_od_main_analysis$Num_Attr_Deaths)
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Sun, Sep 12, 2021 - 17:50:41

Table 2:

11.014	21.884	77.067	146.218	231.200	273.289	306.074	292.898	356.371	355.004	430.985	49

```
num_attr_od_main_analysis$Time_Start<-as.Date(num_attr_od_main_analysis$Time_Start)

#compute the 95% CI for the total
sum(num_attr_od_main_analysis$Num_Attr_Deaths_LB)
```

[1] 31311.05

```
sum(num_attr_od_main_analysis$Num_Attr_Deaths_UB)
```

[1] 42406.57

```
#sum up the number of excess deaths per year
yearly_num_Attr_Deaths_main_analysis<-num_attr_od_main_analysis %>%
  group_by("year" = year(Time_Start)) %>%
  summarise("deaths" = sum(Num_Attr_Deaths), death_lb = sum(Num_Attr_Deaths_LB),
            death_ub = sum(Num_Attr_Deaths_UB))

stargazer(yearly_num_Attr_Deaths_main_analysis$deaths)
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Sun, Sep 12, 2021 - 17:50:41

Table 3:

32.898	223.285	504.489	598.972	711.375	925.207	1,178.500	1,288.211	1,439.185	1,491.513	1,(

```
stargazer(yearly_num_Attr_Deaths_main_analysis$death_lb)
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Sun, Sep 12, 2021 - 17:50:41

Table 4:

27.900	189.401	428.202	508.407	603.805	785.265	1,000.331	1,093.449	1,221.565	1,266.016	1,3

```
stargazer(yearly_num_Attr_Deaths_main_analysis$death_ub)
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Sun, Sep 12, 2021 - 17:50:41

Table 5:

37.853	256.857	579.975	688.584	817.816	1,063.694	1,354.785	1,480.917	1,654.519	1,714.629	1