

# 120c Data Report 2

```
#Created a global environment file from part 1, then ran questions 1-3
```

```
#Then loaded global environment file from part 2 to run question 4
```

```
#libraries and source
```

```
library(dplyr) ##please update it to the latest version
```

```
library(stringr)
```

```
library(zoo)
```

```
library(ggplot2)
```

```
library(urbnmapr)
```

```
#libraries from part2
```

```
library(deSolve)
```

```
library(mFilter)
```

```
library(RobustGaSP)
```

```
library(matrixStats)
```

```
source(file = "C:/Users/David/Desktop/Pstat 120c/120c Report 2/covid-19-20201128/data_and_functions/functions_cov  
id19.R")
```

```
set.seed(1)
```

```
#Load Part 1 Environment (created by running "120c Report 2 Part 1.R")
```

```
load(file = "C:/Users/David/Desktop/Pstat 120c/120c Report 2/Part1Enviro.RData")
```

1a)

```

# 1a Dates national -----

##get dates you want to analyze
start_date = as.Date("2020-3-25")

end_date = as.Date("2020-10-25")

# Ntnl death/confirm -----

##the deaths and confirmed cases for the state on the selected dates
nation_death_selected = nation_death_sum[1 + which(all_dates %in% seq.Date(start_date, end_date, by=1))]
nation_confirmed_selected = nation_confirmed_sum[which(all_dates %in% seq.Date(start_date, end_date, by=1))]

nation_death_selected=as.numeric(nation_death_selected)
nation_confirmed_selected=as.numeric(nation_confirmed_selected)

##plot cumulative confirmed cases and death
date_selected=seq.Date(start_date, end_date, by=1)
par(mfrow=c(1,2))

##daily increase between each date
daily_date_selected=date_selected[2:length(date_selected)]

##let's get the daily confirmed cases
nation_confirmed_selected_daily=nation_confirmed_selected[2:length(nation_confirmed_selected)]-nation_confirmed_selected[1:(length(nation_confirmed_selected)-1)]
##create a data frame
daily_confirmed_nation_df = data.frame(date = daily_date_selected, value = nation_confirmed_selected_daily)

##let's get the daily death cases
nation_death_selected_daily=nation_death_selected[2:length(nation_death_selected)]-nation_death_selected[1:(length(nation_death_selected)-1)]
##create a data frame
daily_death_nation_df = data.frame(date = daily_date_selected, value = nation_death_selected_daily)

# Ntnl confirmed graph -----

##let's obtain a seven-day average of the smoothed version of the confirmed cases and deaths
nation_confirmed_selected_daily_avg = data_seven_day_smoothing(nation_confirmed_selected_daily)

daily_confirmed_nation_smoothed_df = data.frame(date = daily_date_selected, value = nation_confirmed_selected_daily_avg)

nation_death_selected_daily_avg = data_seven_day_smoothing(nation_death_selected_daily)

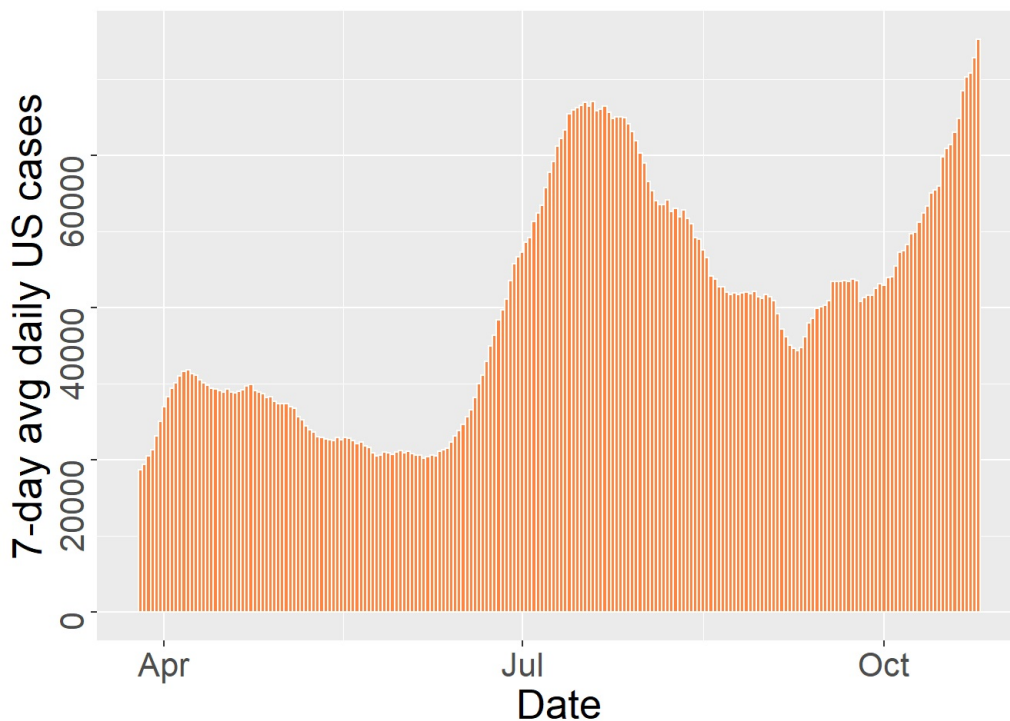
daily_death_nation_smoothed_df = data.frame(date = daily_date_selected, value = nation_death_selected_daily_avg)

# 1.a.1 7-day avg national -----

##plot the smoothed version

###daily confirmed cases in US
daily_confirmed_nation_smoothed_df %>%
  ggplot(aes(x=date, y=value)) +
  geom_bar(stat = 'identity', color="white", fill="#ff8540", width = 1) +
  ylab("7-day avg daily US cases")+
  xlab("Date")+
  theme(text = element_text(size = 20),
        legend.title = element_text(size = 15),
        legend.text = element_text(size = 15),
        legend.key.width=unit(1,"cm"),
        axis.text.y = element_text(angle=90, hjust=1))

```



```
#####this is state-level positive rate
###deal with positive rate

# test+ national graph -----

nation_test = covid_19_project %>%
  dplyr::select(date, state, totalTestResultsIncrease, positiveIncrease)

nation_test_aggregated = nation_test %>%
  group_by(date) %>%
  summarise_each(funs(sum), positiveIncrease, totalTestResultsIncrease)

nation_test_aggregated$positiveIncrease_7_day_avg = data_seven_day_smoothing(nation_test_aggregated$positiveIncrease)

nation_test_aggregated$totalTestResultsIncrease_7_day_avg = data_seven_day_smoothing(nation_test_aggregated$totalTestResultsIncrease)

nation_test_aggregated$positive_rate = nation_test_aggregated$positiveIncrease_7_day_avg / nation_test_aggregated$totalTestResultsIncrease_7_day_avg

# ##reverse the sequence because it start from the current date
# nation_daily_test_selected=rev(us_test_PositiveRateus_test_PositiveRate[nation_test_aggregated$date>=(start_date) & nation_test_aggregated$date<=end_date])

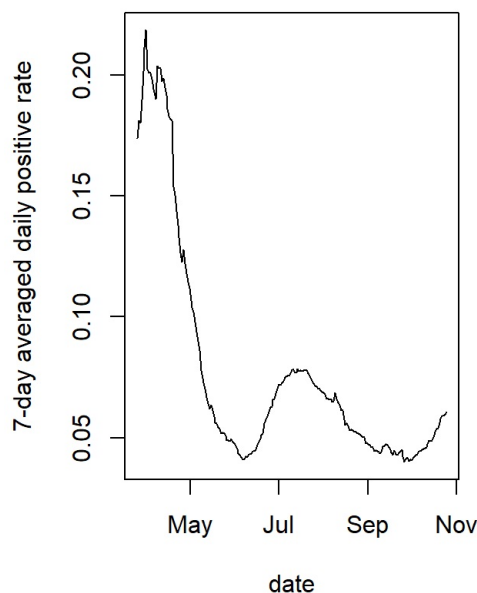
###let's smooth it and get the seven day average
us_test_daily_test_smoothed = data_seven_day_smoothing(nation_test_aggregated$totalTestResultsIncrease)
us_test_daily_positive_smoothed = data_seven_day_smoothing(nation_test_aggregated$positiveIncrease)

##note that the following sequences start from the latest day
us_test_PositiveRate_smoothed = us_test_daily_positive_smoothed / us_test_daily_test_smoothed

us_test_PositiveRate_smoothed_selected=us_test_PositiveRate_smoothed[nation_test_aggregated$date>=(start_date) & nation_test_aggregated$date<=end_date]

# 1.a.2 7-day test+ national-----

###plot the smoothed positive rates
plot(date_selected, us_test_PositiveRate_smoothed_selected, type='l', xlab='date', ylab='7-day averaged daily positive rate')
```



On April 20, the daily positive test rate is around 20% while the daily confirmed cases are hovering around 38,000 new cases nationally.

On July 20, the daily positive test rate is around 8% while the daily confirmed cases exceed 60,000 new cases nationally.

The reason that the positive testing rate decreases while the daily cases increase is likely because of the increase in testing availability as the year went on. More available tests means that tests become more available to asymptomatic/non-infected individuals, which brings down the average test positivity ratio. It does not, however, mean that the virus is being curbed.

1b)

```
# 1.b.1 Apr 20-----

date_for_map = as.Date("2020-4-20")

# extract the averaged daily confirmed cases for all US counties on the selected date
nation_daily_confirmed_avg_selected = us_daily_confirm_clean_avg[, c(1:11, 11+which(all_dates==date_for_map))]

# Ratio = 7-day averaged daily confirmed cases / population
nation_daily_confirmed_rate_avg_selected = nation_daily_confirmed_avg_selected
nation_daily_confirmed_rate_avg_selected[,12] = nation_daily_confirmed_rate_avg_selected[,12] / nation_population
nation_daily_confirmed_rate_avg_selected[,12][nation_daily_confirmed_rate_avg_selected[,12] == 0] = NA
colnames(nation_daily_confirmed_rate_avg_selected)[12] = "Ratio"

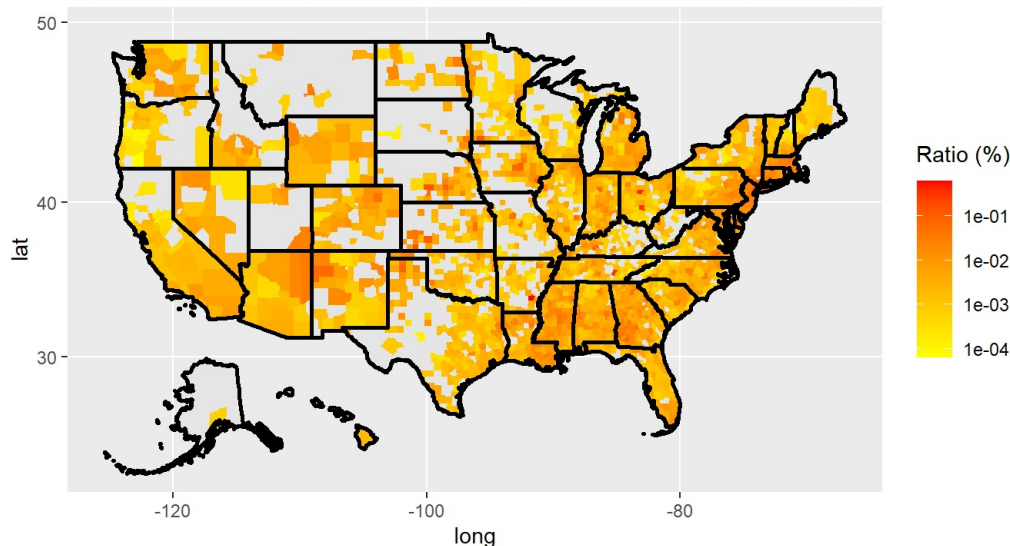
# joint the averaged daily confirmed cases with the shape file for all US counties
nation_daily_confirmed_rate_avg_selected_joint <- left_join(nation_daily_confirmed_rate_avg_selected, counties, b
y = "county_fips")

range(nation_daily_confirmed_rate_avg_selected$Ratio*100, na.rm=T)
```

```
## [1] -0.006166939 0.624122499
```

```
nation_daily_confirmed_rate_avg_selected_joint %>%
ggplot(aes(long, lat, group = group)) +
geom_polygon(aes(fill = (Ratio*100)), show.legend = T) +
geom_polygon(
data = urbnmapr::states, mapping = aes(x = long, y = lat, group = group),
fill = NA, color = 'black', size = 1
) +
scale_fill_gradient(low = "yellow", high = "red", na.value = "grey90", trans = "log10"
###, limits = c(0.00038, 1.158)
)+
coord_map() +
labs(fill = expression("Ratio (%)")) +
ggtitle(paste0("7-day averaged daily confirmed cases / population in the U.S.", ", ", date_for_map))
```

# 7-day averaged daily confirmed cases / population in the U.S., 2020-04-20



```
# 1.b.2 July 20-----

date_for_map = as.Date("2020-7-20")

# extract the averaged daily confirmed cases for all US counties on the selected date
nation_daily_confirmed_avg_selected = us_daily_confirm_clean_avg[, c(1:11, 11+which(all_dates==date_for_map))]

# Ratio = 7-day averaged daily confirmed cases / population
nation_daily_confirmed_rate_avg_selected = nation_daily_confirmed_avg_selected
nation_daily_confirmed_rate_avg_selected[,12] = nation_daily_confirmed_rate_avg_selected[,12] / nation_population
nation_daily_confirmed_rate_avg_selected[,12][nation_daily_confirmed_rate_avg_selected[,12] == 0] = NA
colnames(nation_daily_confirmed_rate_avg_selected)[12] = "Ratio"

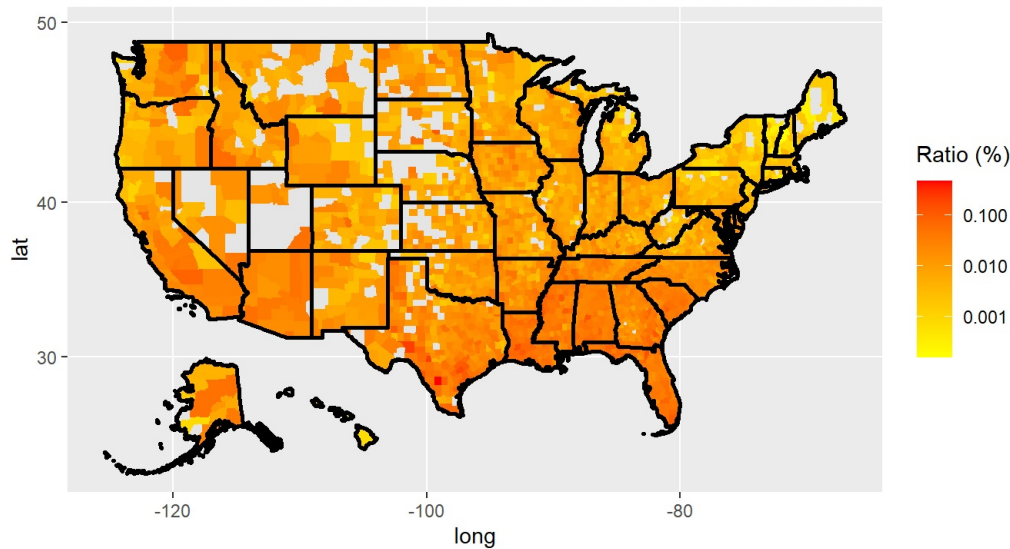
# joint the averaged daily confirmed cases with the shape file for all US counties
nation_daily_confirmed_rate_avg_selected_joint <- left_join(nation_daily_confirmed_rate_avg_selected, counties, by = "county_fips")

range(nation_daily_confirmed_rate_avg_selected$Ratio*100, na.rm=T)
```

```
## [1] -0.0122519 0.4863222
```

```
nation_daily_confirmed_rate_avg_selected_joint %>%
  ggplot(aes(long, lat, group = group)) +
  geom_polygon(aes(fill = (Ratio*100)), show.legend = T) +
  geom_polygon(
    data = urbnmapr::states, mapping = aes(x = long, y = lat, group = group),
    fill = NA, color = 'black', size = 1
  ) +
  scale_fill_gradient(low = "yellow", high = "red", na.value = "grey90", trans = "log10"
    ###, limits = c(0.00038, 1.158)
  ) +
  coord_map() +
  labs(fill = expression("Ratio (%)")) +
  ggtitle(paste0("7-day averaged daily confirmed cases / population in the U.S.", ", ", date_for_map))
```

# 7-day averaged daily confirmed cases / population in the U.S., 2020-07-20



```
# 1.b.3 Oct 20-----

date_for_map = as.Date("2020-10-20")

# extract the averaged daily confirmed cases for all US counties on the selected date
nation_daily_confirmed_avg_selected = us_daily_confirm_clean_avg[, c(1:11, 11+which(all_dates==date_for_map))]

# Ratio = 7-day averaged daily confirmed cases / population
nation_daily_confirmed_rate_avg_selected = nation_daily_confirmed_avg_selected
nation_daily_confirmed_rate_avg_selected[,12] = nation_daily_confirmed_rate_avg_selected[,12] / nation_population
nation_daily_confirmed_rate_avg_selected[,12][nation_daily_confirmed_rate_avg_selected[,12] == 0] = NA
colnames(nation_daily_confirmed_rate_avg_selected)[12] = "Ratio"

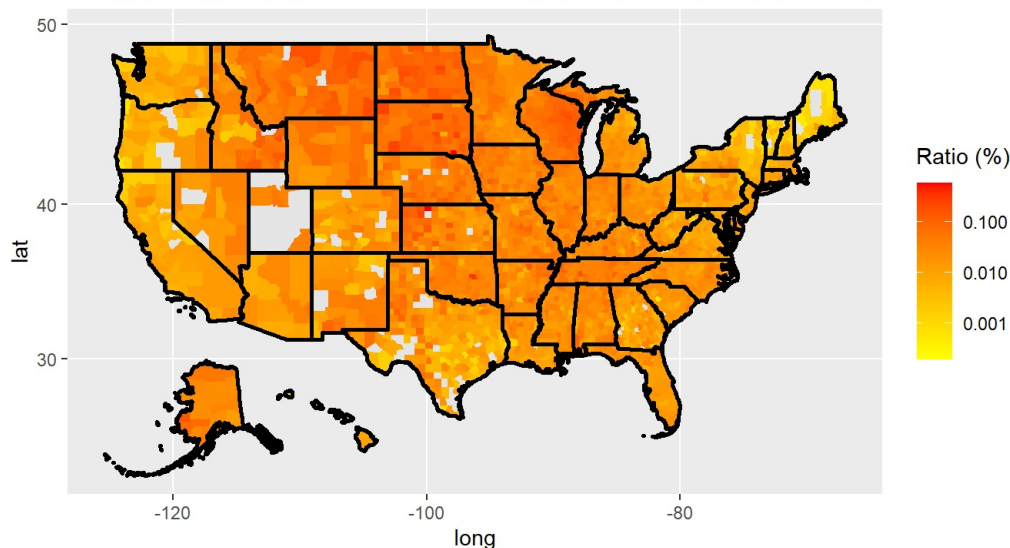
# joint the averaged daily confirmed cases with the shape file for all US counties
nation_daily_confirmed_rate_avg_selected_joint <- left_join(nation_daily_confirmed_rate_avg_selected, counties, by = "county_fips")

range(nation_daily_confirmed_rate_avg_selected$Ratio*100, na.rm=T)
```

```
## [1] -0.05078323 0.59956831
```

```
nation_daily_confirmed_rate_avg_selected_joint %>%
  ggplot(aes(long, lat, group = group)) +
  geom_polygon(aes(fill = (Ratio*100)), show.legend = T) +
  geom_polygon(
    data = urbnmapr::states, mapping = aes(x = long, y = lat, group = group),
    fill = NA, color = 'black', size = 1
  ) +
  scale_fill_gradient(low = "yellow", high = "red", na.value = "grey90", trans = "log10"
    ###, limits = c(0.00038, 1.158)
  ) +
  coord_map() +
  labs(fill = expression("Ratio (%)")) +
  ggtitle(paste0("7-day averaged daily confirmed cases / population in the U.S.", ", ", date_for_map))
```

7-day averaged daily confirmed cases / population in the U.S., 2020-10-20



1c) The general trend of 1b's plots is that of movement from the coastal New England area towards states in the southern belt and then movement into the country's Midwestern interior. As more outbreaks reach the interior, we see testing data become more comprehensive and positive in those areas.

1d) On April 20, I would restrict travel out of New York, New Jersey, and Massachusetts metropolitan areas to curb the spread away from the most affected areas. I would advise that governors ramp up pro-mask messaging and give promises of federal assistance to states that follow strict schedules of business closures and re-openings based on testing data analysis by the NIH.

On June 20, I would mobilize the National Guard to create ad-hoc testing sites and hospitals to support afflicted states in the South. I would invoke strong rhetoric encouraging business closures and public gathering bans in all states. If possible, I would try to support federal legislation mandating mask wearing and social distancing, enforceable by fines. Also, to support business closures, I would give citizens stipends to stay home and avoid virus spreading.

On October 20, the situation is very bad in the Midwest, so I would remobilize the National Guard to assist under-staffed and under-equipped hospitals in these areas. At this point, it's clear that winter weather is exacerbating the spread of the virus. So, in advance of the holiday season, I would create financial incentives for citizens to opt into a national contact tracing system that provides tax breaks and additional stipends for staying at home.

2a)

```

# 2a Dates Michigan -----

##get dates you want to analyze
start_date = as.Date("2020-7-20")

end_date = as.Date("2020-10-26")

##the deaths and confirmed cases for the state on the selected dates
MI_death_selected = MI_death_sum[1 + which(all_dates %in% seq.Date(start_date, end_date, by=1))]
MI_confirmed_selected = MI_confirmed_sum[which(all_dates %in% seq.Date(start_date, end_date, by=1))]

MI_death_selected=as.numeric(MI_death_selected)
MI_confirmed_selected=as.numeric(MI_confirmed_selected)

##plot cumulative confirmed cases and death
date_selected=seq.Date(start_date, end_date, by=1)
par(mfrow=c(1,2))

##daily increase between each date
daily_date_selected=date_selected[2:length(date_selected)]

##let's get the daily confirmed cases
MI_confirmed_selected_daily=MI_confirmed_selected[2:length(MI_confirmed_selected)]-MI_confirmed_selected[1:(length(MI_confirmed_selected)-1)]
##create a data frame
daily_confirmed_MI_df = data.frame(date = daily_date_selected, value = MI_confirmed_selected_daily)

##let's get the daily death cases
MI_death_selected_daily=MI_death_selected[2:length(MI_death_selected)]-MI_death_selected[1:(length(MI_death_selected)-1)]
##create a data frame
daily_death_MI_df = data.frame(date = daily_date_selected, value = MI_death_selected_daily)

##let's obtain a seven-day average of the smoothed version of the confirmed cases and deaths

MI_confirmed_selected_daily_avg = data_seven_day_smoothing(MI_confirmed_selected_daily)

daily_confirmed_MI_smoothed_df = data.frame(date = daily_date_selected, value = MI_confirmed_selected_daily_avg)

MI_death_selected_daily_avg = data_seven_day_smoothing(MI_death_selected_daily)

daily_death_MI_smoothed_df = data.frame(date = daily_date_selected, value = MI_death_selected_daily_avg)

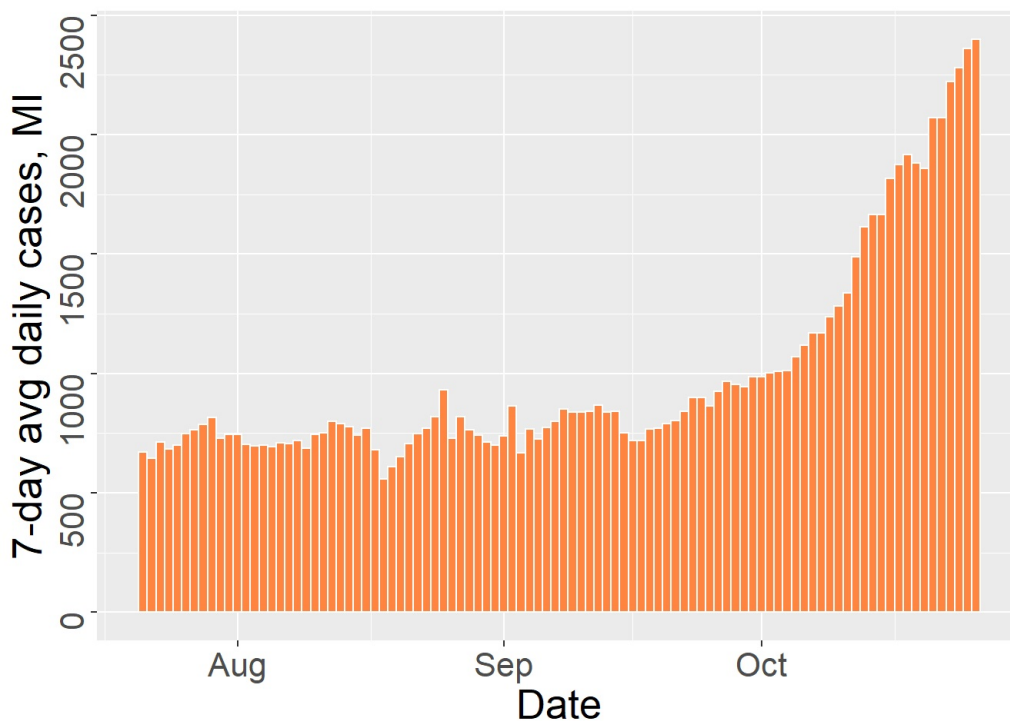
# 2.a.1 7-day avg MI -----

##plot the smoothed version

###daily confirmed cases in MI
daily_confirmed_MI_smoothed_df %>%
  ggplot(aes(x=date, y=value)) +
  geom_bar(stat = 'identity', color="white", fill="#ff8540", width = 1) +
  ylab("7-day avg daily cases, MI")+
  xlab("Date")+
  theme(text = element_text(size = 20),
        legend.title = element_text(size = 15),
        legend.text = element_text(size = 15),
        legend.key.width=unit(1,"cm"),
        axis.text.y = element_text(angle=90, hjust=1))

```





```
##get the test positive rates

#####this is state-level positive rate
###deal with positive rate

MI_test =
  dplyr::filter(nation_test, state == "WI")

MI_test_aggregated = MI_test %>%
  group_by(date) %>%
  summarise_each(funs(sum), positiveIncrease, totalTestResultsIncrease)

MI_test_aggregated$positiveIncrease_7_day_avg = data_seven_day_smoothing(MI_test_aggregated$positiveIncrease)

MI_test_aggregated$totalTestResultsIncrease_7_day_avg = data_seven_day_smoothing(MI_test_aggregated$totalTestResultsIncrease)

MI_test_aggregated$positive_rate = MI_test_aggregated$positiveIncrease_7_day_avg / MI_test_aggregated$totalTestResultsIncrease_7_day_avg

# ##reverse the sequence because it start from the current date
# nation_daily_test_selected=rev(us_test_PositiveRateus_test_PositiveRate[nation_test_aggregated$date>=(start_date) & nation_test_aggregated$date<=end_date])

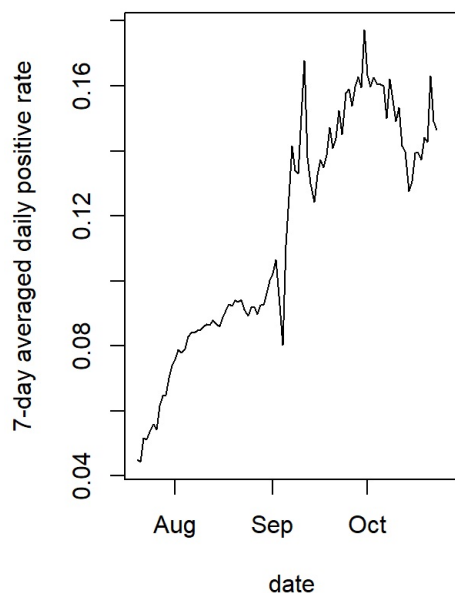
###let's smooth it and get the seven day average
MI_test_daily_test_smoothed = data_seven_day_smoothing(MI_test_aggregated$totalTestResultsIncrease)
MI_test_daily_positive_smoothed = data_seven_day_smoothing(MI_test_aggregated$positiveIncrease)

##note that the following sequences start from the latest day
MI_test_PositiveRate_smoothed = MI_test_daily_positive_smoothed / MI_test_daily_test_smoothed

MI_test_PositiveRate_smoothed_selected=MI_test_PositiveRate_smoothed[nation_test_aggregated$date>=(start_date) & nation_test_aggregated$date<=end_date]

# 2.a.2 7-day test+ MI-----

###plot the smoothed positive rates
plot(date_selected, MI_test_PositiveRate_smoothed_selected,type='l',xlab='date',ylab='7-day averaged daily positive rate')
```



Due to the upward trend of the chart, it looks like more people were infected on Oct 20 than July 20.

The 7-day average of the test positive rate also trends upward, indicating a correlation between the charts.

2b)

```
# 2.b.1 MI 7-day cty map July 20 -----

###Let's see whether we can make a map for MI about the confirmed cases over county population
### at a particular date

# set the date for map the end date you selected before
date_for_map = as.Date("2020-7-20")

##Start of state-specific code

##you can make the daily death cases to be zero if it is smaller than zero
#us_daily_death_clean[us_daily_death_clean<0] = 0

# select the daily confirm cases for CA counties
MI_daily_confirm = us_daily_confirm_clean%>%
  filter(Province_State == "Michigan")

# This shape file contains the coordinates for county boundaries
MI_shape_data = counties %>%
  filter(state_name == "Michigan")

# get the population for CA counties
MI_population = us_daily_death_clean %>%
  filter(Province_State == "Michigan")%>%
  dplyr::select(Population)
MI_population = as.numeric(MI_population$Population)

# extract the daily confirmed cases on the selected date date_for_map
MI_daily_confirm_selected = MI_daily_confirm[, c(1:11, 11+ which(all_dates == date_for_map))]

# the Ratio = daily confirmed cases / population
MI_daily_confirm_rate_selected = MI_daily_confirm_selected
MI_daily_confirm_rate_selected[,12] = MI_daily_confirm_selected[,12]/MI_population
MI_daily_confirm_rate_selected[,12][MI_daily_confirm_rate_selected[,12]==0]=NA
colnames(MI_daily_confirm_rate_selected)[12] = "Ratio"

##This is the county and rate
#MI_daily_confirm_rate_selected[,c(11,12)]

# joint the daily confirmed cases with the shape file
MI_daily_confirm_rate_selected_joint <- left_join(MI_daily_confirm_rate_selected, MI_shape_data, by = "county_fips")

# find the lower and upper limits of the ratio
range(MI_daily_confirm_rate_selected$Ratio*100, na.rm = T)
```

```
## [1] -0.002301337 0.024018666
```

```
# calculate the 7-day averaged daily confirm cases
MI_daily_confirm_avg = MI_daily_confirm
MI_daily_confirm_avg[, 13:dim(MI_daily_confirm)[2]] = t(apply(MI_daily_confirm[,13:dim(MI_daily_confirm)[2]], 1,
data_seven_day_smoothing))

# extract the averaged daily confirmed cases on the selected date
MI_daily_confirm_avg_selected = MI_daily_confirm_avg[, c(1:11, 11+ which(all_dates == date_for_map))]

# the Ratio = averaged daily confirmed cases / population
MI_daily_confirm_rate_avg_selected = MI_daily_confirm_avg_selected
MI_daily_confirm_rate_avg_selected[, 12] = MI_daily_confirm_avg_selected[, 12]/MI_population
MI_daily_confirm_rate_avg_selected[,12][MI_daily_confirm_rate_avg_selected[,12]==0]=NA
colnames(MI_daily_confirm_rate_avg_selected)[12] = "Ratio"

## county and rate
#MI_daily_confirm_rate_avg_selected[,c(11,12)]

index_largest=which(MI_daily_confirm_rate_avg_selected[,c(12)]==max(MI_daily_confirm_rate_avg_selected[,c(12)]))
#MI_daily_confirm_rate_avg_selected[index_largest,c(11,12)]

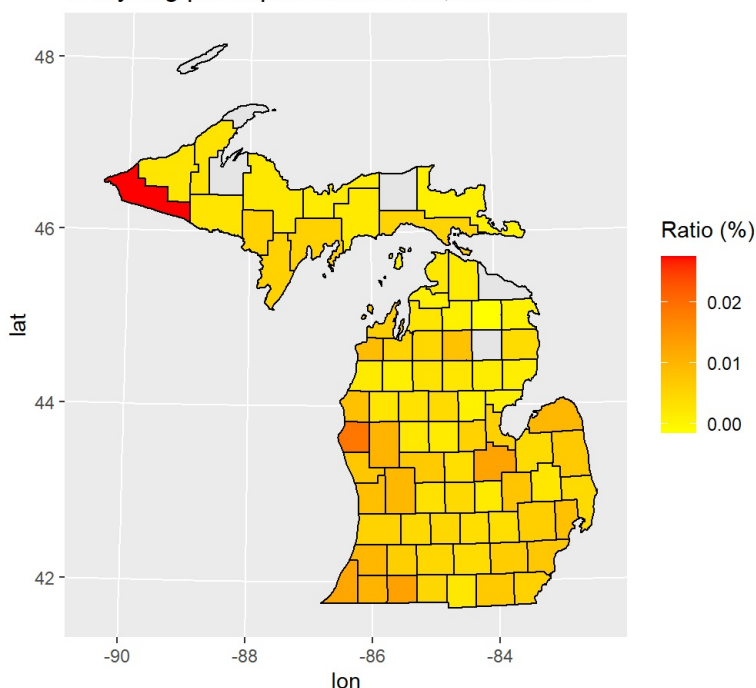
# joint the averaged daily confirmed cases with the shape file
MI_daily_confirm_rate_avg_selected_joint <- left_join(MI_daily_confirm_rate_avg_selected, MI_shape_data, by = "co
unty_fips")

# find the lower and upper limits of the ratio
range(MI_daily_confirm_rate_avg_selected$Ratio*100, na.rm = T)
```

```
## [1] -0.001531487 0.027600307
```

```
MI_daily_confirm_rate_avg_selected_joint %>%
ggplot(aes(long, lat, group = group, fill = Ratio*100)) +
scale_fill_gradient(low = "yellow", high = "red", na.value = "grey90")+
geom_polygon(col = "black") +
coord_map(projection = "albers", lat0 = 10, lat1 = 45) +
labs(fill = expression("Ratio (%)")) +
ggtitle(paste0("7-day avg per capita cases in MI", " ", " ", date_for_map))+
xlab("lon") +ylab("lat" )
```

7-day avg per capita cases in MI, 2020-07-20



```
# 2.b.2 MI cty map Oct 20 -----

# set the date for map the end date you selected before
date_for_map = as.Date("2020-10-20")

##Start of state-specific code

##you can make the daily death cases to be zero if it is smaller than zero
#us_daily_death_clean[us_daily_death_clean<0] = 0

# select the daily confirm cases for CA counties
MI_daily_confirm = us_daily_confirm_clean%>%
  filter(Province_State == "Michigan")

# This shape file contains the coordinates for county boundaries
MI_shape_data = counties %>%
  filter(state_name == "Michigan")

# get the population for CA counties
MI_population = us_daily_death_clean %>%
  filter(Province_State == "Michigan")%>%
  dplyr::select(Population)
MI_population = as.numeric(MI_population$Population)

# extract the daily confirmed cases on the selected date date_for_map
MI_daily_confirm_selected = MI_daily_confirm[, c(1:11, 11+ which(all_dates == date_for_map))]

# the Ratio = daily confirmed cases / population
MI_daily_confirm_rate_selected = MI_daily_confirm_selected
MI_daily_confirm_rate_selected[,12] = MI_daily_confirm_selected[,12]/MI_population
MI_daily_confirm_rate_selected[,12][MI_daily_confirm_rate_selected[,12]==0]=NA
colnames(MI_daily_confirm_rate_selected)[12] = "Ratio"

##This is the county and rate
#MI_daily_confirm_rate_selected[,c(11,12)]

# joint the daily confirmed cases with the shape file
MI_daily_confirm_rate_selected_joint <- left_join(MI_daily_confirm_rate_selected, MI_shape_data, by = "county_fips")

# find the lower and upper limits of the ratio
range(MI_daily_confirm_rate_selected$Ratio*100, na.rm = T)
```

```
## [1] -0.003956322 0.178890877
```

```
# calculate the 7-day averaged daily confirm cases
MI_daily_confirm_avg = MI_daily_confirm
MI_daily_confirm_avg[, 13:dim(MI_daily_confirm)[2]] = t(apply(MI_daily_confirm[,13:dim(MI_daily_confirm)[2]], 1,
data_seven_day_smoothing))

# extract the averaged daily confirmed cases on the selected date
MI_daily_confirm_avg_selected = MI_daily_confirm_avg[, c(1:11, 11+ which(all_dates == date_for_map))]

# the Ratio = averaged daily confirmed cases / population
MI_daily_confirm_rate_avg_selected = MI_daily_confirm_avg_selected
MI_daily_confirm_rate_avg_selected[, 12] = MI_daily_confirm_avg_selected[, 12]/MI_population
MI_daily_confirm_rate_avg_selected[,12][MI_daily_confirm_rate_avg_selected[,12]==0]=NA
colnames(MI_daily_confirm_rate_avg_selected)[12] = "Ratio"

## county and rate
#MI_daily_confirm_rate_avg_selected[,c(11,12)]

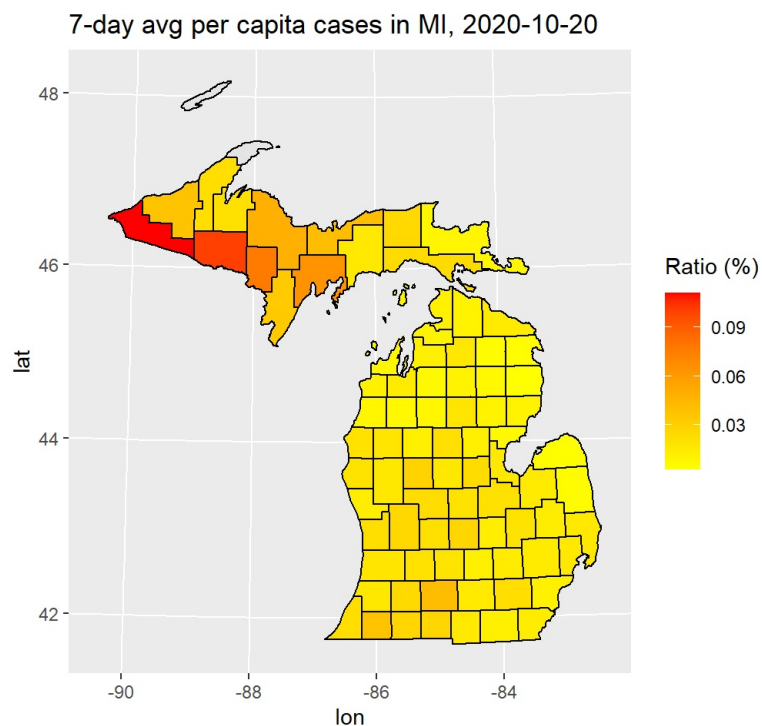
index_largest=which(MI_daily_confirm_rate_avg_selected[,c(12)]==max(MI_daily_confirm_rate_avg_selected[,c(12)]))
#MI_daily_confirm_rate_avg_selected[index_largest,c(11,12)]

# joint the averaged daily confirmed cases with the shape file
MI_daily_confirm_rate_avg_selected_joint <- left_join(MI_daily_confirm_rate_avg_selected, MI_shape_data, by = "county_fips")

# find the lower and upper limits of the ratio
range(MI_daily_confirm_rate_avg_selected$Ratio*100, na.rm = T)
```

```
## [1] 0.002745933 0.111423460
```

```
MI_daily_confirm_rate_avg_selected_joint %>%
  ggplot(aes(long, lat, group = group, fill = Ratio*100)) +
  scale_fill_gradient(low = "yellow", high = "red", na.value = "grey90")+
  geom_polygon(col = "black") +
  coord_map(projection = "albers", lat0 = 10, lat1 = 45) +
  labs(fill = expression("Ratio (%)")) +
  ggtitle(paste0("7-day avg per capita cases in MI", " ", " ", date_for_map))+
  xlab("lon") +ylab("lat" )
```



The plots suggest a general trend of increased per capita confirmed cases over time, especially heightened in the northern counties.

2c)

```
# 2.c.1 MI cty map Oct 22 -----
```

```
# set the date for map the end date you selected before
date_for_map = as.Date("2020-10-22")
```

```
##Start of state-specific code
```

```
##you can make the daily death cases to be zero if it is smaller than zero
#us_daily_death_clean[us_daily_death_clean<0] = 0
```

```
# select the daily confirm cases for CA counties
```

```
MI_daily_confirm = us_daily_confirm_clean%>%
  filter(Province_State == "Michigan")
```

```
# This shape file contains the coordinates for county boundaries
```

```
MI_shape_data = counties %>%
  filter(state_name == "Michigan")
```

```
# get the population for CA counties
```

```
MI_population = us_daily_death_clean %>%
  filter(Province_State == "Michigan")%>%
  dplyr::select(Population)
MI_population = as.numeric(MI_population$Population)
```

```
# extract the daily confirmed cases on the selected date date_for_map
```

```
MI_daily_confirm_selected = MI_daily_confirm[, c(1:11, 11+ which(all_dates == date_for_map))]
```

```
# the Ratio = daily confirmed cases / population
```

```
MI_daily_confirm_rate_selected = MI_daily_confirm_selected
MI_daily_confirm_rate_selected[,12] = MI_daily_confirm_selected[,12]/MI_population
MI_daily_confirm_rate_selected[,12][MI_daily_confirm_rate_selected[,12]==0]=NA
colnames(MI_daily_confirm_rate_selected)[12] = "Ratio"
```

```
##This is the county and rate
```

```
#MI_daily_confirm_rate_selected[,c(11,12)]
```

```
# joint the daily confirmed cases with the shape file
```

```
MI_daily_confirm_rate_selected_joint <- left_join(MI_daily_confirm_rate_selected, MI_shape_data, by = "county_fips")
```

```
# find the lower and upper limits of the ratio
```

```
range(MI_daily_confirm_rate_selected$Ratio*100, na.rm = T)
```

```
## [1] -0.006614632 0.150268336
```

```
# calculate the 7-day averaged daily confirm cases
```

```
MI_daily_confirm_avg = MI_daily_confirm
MI_daily_confirm_avg[, 13:dim(MI_daily_confirm)[2]] = t(apply(MI_daily_confirm[,13:dim(MI_daily_confirm)[2]], 1,
data_seven_day_smoothing))
```

```
# extract the averaged daily confirmed cases on the selected date
```

```
MI_daily_confirm_avg_selected = MI_daily_confirm_avg[, c(1:11, 11+ which(all_dates == date_for_map))]
```

```
# the Ratio = averaged daily confirmed cases / population
```

```
MI_daily_confirm_rate_avg_selected = MI_daily_confirm_avg_selected
MI_daily_confirm_rate_avg_selected[, 12] = MI_daily_confirm_avg_selected[, 12]/MI_population
MI_daily_confirm_rate_avg_selected[,12][MI_daily_confirm_rate_avg_selected[,12]==0]=NA
colnames(MI_daily_confirm_rate_avg_selected)[12] = "Ratio"
```

```
## county and rate
```

```
#MI_daily_confirm_rate_avg_selected[,c(11,12)]
```

```
index_largest=which(MI_daily_confirm_rate_avg_selected[,c(12)]==max(MI_daily_confirm_rate_avg_selected[,c(12)]))
```

```
#MI_daily_confirm_rate_avg_selected[index_largest,c(11,12)]
```

```
# joint the averaged daily confirmed cases with the shape file
```

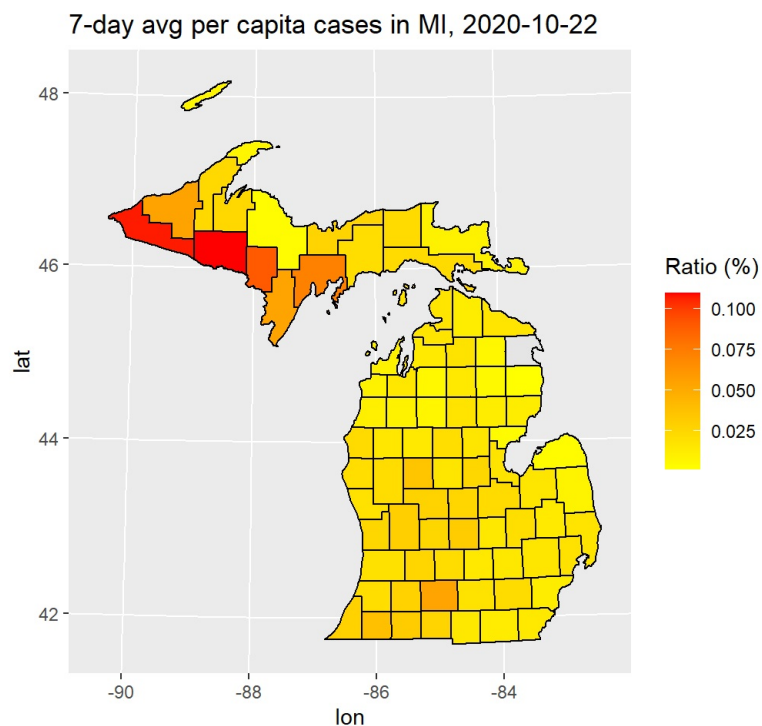
```
MI_daily_confirm_rate_avg_selected_joint <- left_join(MI_daily_confirm_rate_avg_selected, MI_shape_data, by = "county_fips")
```

```
# find the lower and upper limits of the ratio
```

```
range(MI_daily_confirm_rate_avg_selected$Ratio*100, na.rm = T)
```

```
## [1] 0.001372966 0.109731223
```

```
MI_daily_confirm_rate_avg_selected_joint %>%
  ggplot(aes(long, lat, group = group, fill = Ratio*100)) +
  scale_fill_gradient(low = "yellow", high = "red", na.value = "grey90")+
  geom_polygon(col = "black") +
  coord_map(projection = "albers", lat0 = 10, lat1 = 45) +
  labs(fill = expression("Ratio (%)")) +
  ggtitle(paste0("7-day avg per capita cases in MI", " ", " ", date_for_map))+
  xlab("lon") +ylab("lat" )
```



```
# 2.c.2 MI cty map Oct 24 -----
```

```
# set the date for map the end date you selected before
date_for_map = as.Date("2020-10-24")

##Start of state-specific code

##you can make the daily death cases to be zero if it is smaller than zero
#us_daily_death_clean[us_daily_death_clean<0] = 0

# select the daily confirm cases for CA counties
MI_daily_confirm = us_daily_confirm_clean%>%
  filter(Province_State == "Michigan")

# This shape file contains the coordinates for county boundaries
MI_shape_data = counties %>%
  filter(state_name == "Michigan")

# get the population for CA counties
MI_population = us_daily_death_clean %>%
  filter(Province_State == "Michigan")%>%
  dplyr::select(Population)
MI_population = as.numeric(MI_population$Population)

# extract the daily confirmed cases on the selected date date_for_map
MI_daily_confirm_selected = MI_daily_confirm[, c(1:11, 11+ which(all_dates == date_for_map)))]

# the Ratio = daily confirmed cases / population
MI_daily_confirm_rate_selected = MI_daily_confirm_selected
MI_daily_confirm_rate_selected[,12] = MI_daily_confirm_selected[,12]/MI_population
MI_daily_confirm_rate_selected[,12][MI_daily_confirm_rate_selected[,12]==0]=NA
colnames(MI_daily_confirm_rate_selected)[12] = "Ratio"

##This is the county and rate
#MI_daily_confirm_rate_selected[,c(11,12)]

# joint the daily confirmed cases with the shape file
MI_daily_confirm_rate_selected_joint <- left_join(MI_daily_confirm_rate_selected, MI_shape_data, by = "county_fips")

# find the lower and upper limits of the ratio
range(MI_daily_confirm_rate_selected$Ratio*100, na.rm = T)
```

```
## [1] -0.01605394 0.20999247
```

```
# calculate the 7-day averaged daily confirm cases
MI_daily_confirm_avg = MI_daily_confirm
MI_daily_confirm_avg[, 13:dim(MI_daily_confirm)[2]] = t(apply(MI_daily_confirm[,13:dim(MI_daily_confirm)[2]], 1,
data_seven_day_smoothing))

# extract the averaged daily confirmed cases on the selected date
MI_daily_confirm_avg_selected = MI_daily_confirm_avg[, c(1:11, 11+ which(all_dates == date_for_map)))]

# the Ratio = averaged daily confirmed cases / population
MI_daily_confirm_rate_avg_selected = MI_daily_confirm_avg_selected
MI_daily_confirm_rate_avg_selected[, 12] = MI_daily_confirm_avg_selected[, 12]/MI_population
MI_daily_confirm_rate_avg_selected[,12][MI_daily_confirm_rate_avg_selected[,12]==0]=NA
colnames(MI_daily_confirm_rate_avg_selected)[12] = "Ratio"

## county and rate
#MI_daily_confirm_rate_avg_selected[,c(11,12)]

index_largest=which(MI_daily_confirm_rate_avg_selected[,c(12)]==max(MI_daily_confirm_rate_avg_selected[,c(12)]))
#MI_daily_confirm_rate_avg_selected[index_largest,c(11,12)]

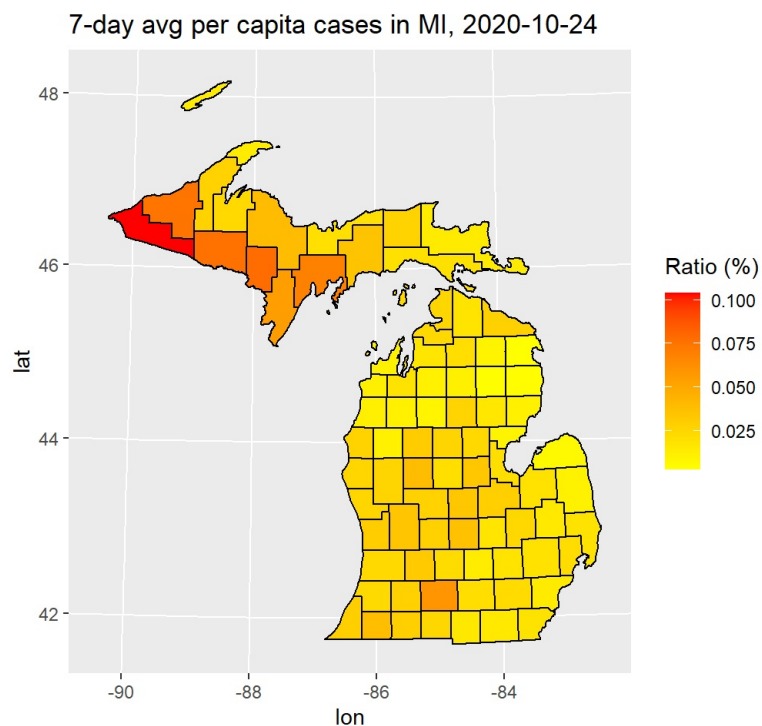
# joint the averaged daily confirmed cases with the shape file
MI_daily_confirm_rate_avg_selected_joint <- left_join(MI_daily_confirm_rate_avg_selected, MI_shape_data, by = "county_fips")

# find the lower and upper limits of the ratio
range(MI_daily_confirm_rate_avg_selected$Ratio*100, na.rm = T)
```

```
## [1] 0.003017577 0.104267825
```



```
MI_daily_confirm_rate_avg_selected_joint %>%
  ggplot(aes(long, lat, group = group, fill = Ratio*100)) +
  scale_fill_gradient(low = "yellow", high = "red", na.value = "grey90")+
  geom_polygon(col = "black") +
  coord_map(projection = "albers", lat0 = 10, lat1 = 45) +
  labs(fill = expression("Ratio (%)")) +
  ggtitle(paste0("7-day avg per capita cases in MI", " ", " ", date_for_map))+
  xlab("lon") +ylab("lat" )
```



```
# 2.c.3 MI cty map Oct 26 -----
```

```
# set the date for map the end date you selected before
date_for_map = as.Date("2020-10-26")
```

```
##Start of state-specific code
```

```
##you can make the daily death cases to be zero if it is smaller than zero
#us_daily_death_clean[us_daily_death_clean<0] = 0
```

```
# select the daily confirm cases for CA counties
```

```
MI_daily_confirm = us_daily_confirm_clean%>%
  filter(Province_State == "Michigan")
```

```
# This shape file contains the coordinates for county boundaries
```

```
MI_shape_data = counties %>%
  filter(state_name == "Michigan")
```

```
# get the population for CA counties
```

```
MI_population = us_daily_death_clean %>%
  filter(Province_State == "Michigan")%>%
  dplyr::select(Population)
MI_population = as.numeric(MI_population$Population)
```

```
# extract the daily confirmed cases on the selected date date_for_map
```

```
MI_daily_confirm_selected = MI_daily_confirm[, c(1:11, 11+ which(all_dates == date_for_map))]
```

```
# the Ratio = daily confirmed cases / population
```

```
MI_daily_confirm_rate_selected = MI_daily_confirm_selected
MI_daily_confirm_rate_selected[,12] = MI_daily_confirm_selected[,12]/MI_population
MI_daily_confirm_rate_selected[,12][MI_daily_confirm_rate_selected[,12]==0]=NA
colnames(MI_daily_confirm_rate_selected)[12] = "Ratio"
```

```
##This is the county and rate
```

```
#MI_daily_confirm_rate_selected[,c(11,12)]
```

```
# joint the daily confirmed cases with the shape file
```

```
MI_daily_confirm_rate_selected_joint <- left_join(MI_daily_confirm_rate_selected, MI_shape_data, by = "county_fips")
```

```
# find the lower and upper limits of the ratio
```

```
range(MI_daily_confirm_rate_selected$Ratio*100, na.rm = T)
```

```
## [1] -0.007128092 0.178890877
```

```
# calculate the 7-day averaged daily confirm cases
```

```
MI_daily_confirm_avg = MI_daily_confirm
MI_daily_confirm_avg[, 13:dim(MI_daily_confirm)[2]] = t(apply(MI_daily_confirm[,13:dim(MI_daily_confirm)[2]], 1,
data_seven_day_smoothing))
```

```
# extract the averaged daily confirmed cases on the selected date
```

```
MI_daily_confirm_avg_selected = MI_daily_confirm_avg[, c(1:11, 11+ which(all_dates == date_for_map))]
```

```
# the Ratio = averaged daily confirmed cases / population
```

```
MI_daily_confirm_rate_avg_selected = MI_daily_confirm_avg_selected
MI_daily_confirm_rate_avg_selected[, 12] = MI_daily_confirm_avg_selected[, 12]/MI_population
MI_daily_confirm_rate_avg_selected[,12][MI_daily_confirm_rate_avg_selected[,12]==0]=NA
colnames(MI_daily_confirm_rate_avg_selected)[12] = "Ratio"
```

```
## county and rate
```

```
#MI_daily_confirm_rate_avg_selected[,c(11,12)]
```

```
index_largest=which(MI_daily_confirm_rate_avg_selected[,c(12)]==max(MI_daily_confirm_rate_avg_selected[,c(12)]))
```

```
#MI_daily_confirm_rate_avg_selected[index_largest,c(11,12)]
```

```
# joint the averaged daily confirmed cases with the shape file
```

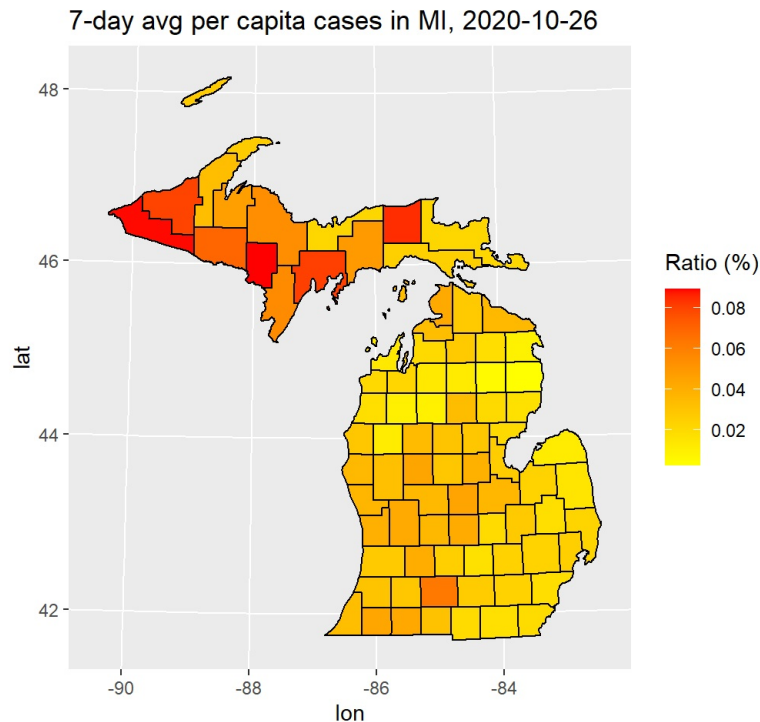
```
MI_daily_confirm_rate_avg_selected_joint <- left_join(MI_daily_confirm_rate_avg_selected, MI_shape_data, by = "county_fips")
```

```
# find the lower and upper limits of the ratio
```

```
range(MI_daily_confirm_rate_avg_selected$Ratio*100, na.rm = T)
```

```
## [1] 0.002745933 0.089430756
```

```
MI_daily_confirm_rate_avg_selected_joint %>%
  ggplot(aes(long, lat, group = group, fill = Ratio*100)) +
  scale_fill_gradient(low = "yellow", high = "red", na.value = "grey90")+
  geom_polygon(col = "black") +
  coord_map(projection = "albers", lat0 = 10, lat1 = 45) +
  labs(fill = expression("Ratio (%)")) +
  ggtitle(paste0("7-day avg per capita cases in MI", " ", " ", date_for_map))+
  xlab("lon") +ylab("lat" )
```



Michigan's upper peninsula is experiencing the fastest increase of confirmed cases, along with Calhoun County in the southern part of the state.

2d)

With nearly a .1% rate confirmed cases per capita, things are not good. I would strongly advise against travel between Michigan proper and the upper peninsula. All non-essential businesses in the the upper peninsula must be closed and social distancing measures enforced by police. A specific travel advisory warning about travel to the hottest regions (including Calhoun County) should be sent out. To the degree allowed by the budget, I will make testing more widely available and make a state address asking citizens to stay at home.

3a)

```
# 3.a.1 cumulative FL -----

##get dates you want to analyze
start_date = as.Date("2020-7-20")

end_date = as.Date("2020-10-26")

##the deaths and confirmed cases for the state on the selected dates
FL_death_selected = FL_death_sum[1 + which(all_dates %in% seq.Date(start_date, end_date, by=1))]
FL_confirmed_selected = FL_confirmed_sum[which(all_dates %in% seq.Date(start_date, end_date, by=1))]

FL_death_selected=as.numeric(FL_death_selected)
FL_confirmed_selected=as.numeric(FL_confirmed_selected)

##plot cumulative confirmed cases and death
date_selected=seq.Date(start_date, end_date, by=1)
par(mfrow=c(1,2))
plot(date_selected,FL_confirmed_selected,xlab='date',ylab='cumulative observed confirmed cases',type='l')

# 3.a2 redoing dates -----

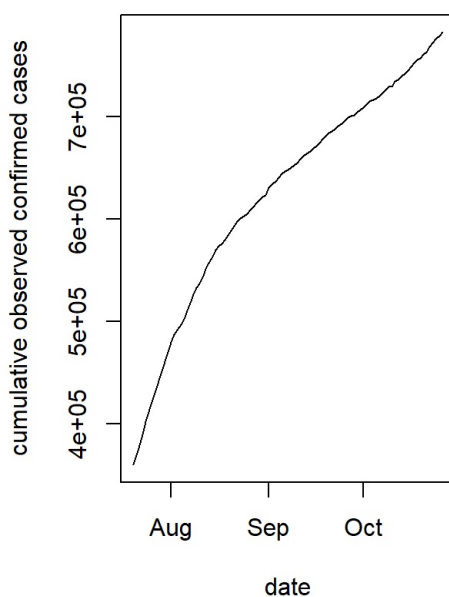
##get dates you want to analyze
start_date = as.Date("2020-7-20")

end_date = as.Date("2020-10-22")

##the deaths and confirmed cases for the state on the selected dates
FL_death_selected = FL_death_sum[1 + which(all_dates %in% seq.Date(start_date, end_date, by=1))]
FL_confirmed_selected = FL_confirmed_sum[which(all_dates %in% seq.Date(start_date, end_date, by=1))]

FL_death_selected=as.numeric(FL_death_selected)
FL_confirmed_selected=as.numeric(FL_confirmed_selected)

##plot cumulative confirmed cases and death
date_selected=seq.Date(start_date, end_date, by=1)
par(mfrow=c(1,2))
```



```

##daily increase between each date
daily_date_selected=date_selected[2:length(date_selected)]

##let's get the daily confirmed cases
FL_confirmed_selected_daily=FL_confirmed_selected[2:length(FL_confirmed_selected)]-FL_confirmed_selected[1:(length(FL_confirmed_selected)-1)]
##create a data frame
daily_confirmed_FL_df = data.frame(date = daily_date_selected, value = FL_confirmed_selected_daily)

##let's get the daily death cases
FL_death_selected_daily=FL_death_selected[2:length(FL_death_selected)]-FL_death_selected[1:(length(FL_death_selected)-1)]
##create a data frame
daily_death_FL_df = data.frame(date = daily_date_selected, value = FL_death_selected_daily)

##let's obtain a seven-day average of the smoothed version of the confirmed cases and deaths

FL_confirmed_selected_daily_avg = data_seven_day_smoothing(FL_confirmed_selected_daily)

daily_confirmed_FL_smoothed_df = data.frame(date = daily_date_selected, value = FL_confirmed_selected_daily_avg)

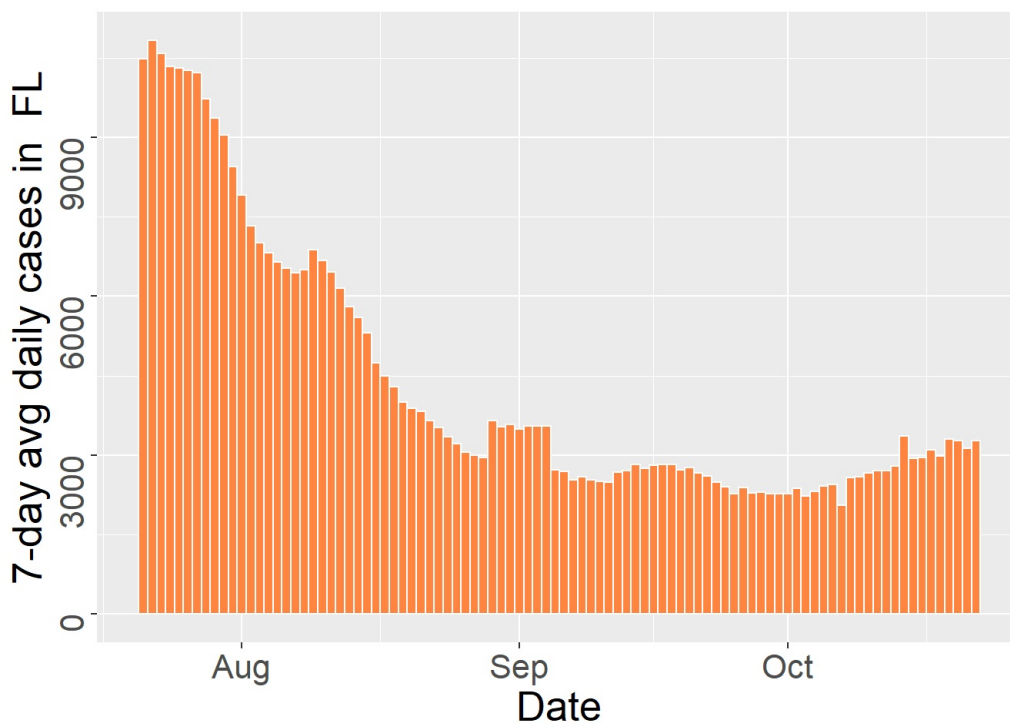
FL_death_selected_daily_avg = data_seven_day_smoothing(FL_death_selected_daily)

daily_death_FL_smoothed_df = data.frame(date = daily_date_selected, value = FL_death_selected_daily_avg)

##plot the smoothed version

###daily confirmed cases in FL
daily_confirmed_FL_smoothed_df %>%
  ggplot(aes(x=date, y=value)) +
  geom_bar(stat = 'identity', color="white", fill="#ff8540", width = 1) +
  ylab("7-day avg daily cases in FL")+
  xlab("Date")+
  theme(text = element_text(size = 20),
        legend.title = element_text(size = 15),
        legend.text = element_text(size = 15),
        legend.key.width=unit(1,"cm"),
        axis.text.y = element_text(angle=90, hjust=1))

```



```
##get the test positive rates

#####this is state-level positive rate
###deal with positive rate

FL_test =
  dplyr::filter(nation_test, state == "FL")

FL_test_aggregated = FL_test %>%
  group_by(date) %>%
  summarise_each(funs(sum), positiveIncrease, totalTestResultsIncrease)

FL_test_aggregated$positiveIncrease_7_day_avg = data_seven_day_smoothing(FL_test_aggregated$positiveIncrease)

FL_test_aggregated$totalTestResultsIncrease_7_day_avg = data_seven_day_smoothing(FL_test_aggregated$totalTestResultsIncrease)

FL_test_aggregated$positive_rate = FL_test_aggregated$positiveIncrease_7_day_avg / FL_test_aggregated$totalTestResultsIncrease_7_day_avg

# ##reverse the sequence because it start from the current date
# nation_daily_test_selected=rev(us_test_PositiveRateus_test_PositiveRate[nation_test_aggregated$date>=(start_date) & nation_test_aggregated$date<=end_date])

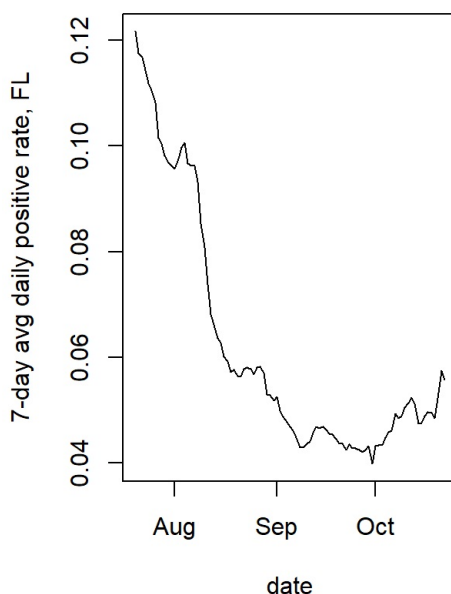
###let's smooth it and get the seven day average
FL_test_daily_test_smoothed = data_seven_day_smoothing(FL_test_aggregated$totalTestResultsIncrease)
FL_test_daily_positive_smoothed = data_seven_day_smoothing(FL_test_aggregated$positiveIncrease)

##note that the following sequences start from the latest day
FL_test_PositiveRate_smoothed = FL_test_daily_positive_smoothed / FL_test_daily_test_smoothed

FL_test_PositiveRate_smoothed_selected=FL_test_PositiveRate_smoothed[nation_test_aggregated$date>=(start_date) & nation_test_aggregated$date<=end_date]

# 3.a.2 7-day test+ FL-----

###plot the smoothed positive rates
plot(date_selected, FL_test_PositiveRate_smoothed_selected,type='l',xlab='date',ylab='7-day avg daily positive rate, FL')
```



It looks like Covid-19 transmission peaked in the summer months. I would say that more were infected on July 20 than October 20 in Florida.

The daily test positivity rate would agree with my assessment, as it drops by about 6 percentage points from July to October.

3b)

```
# 3.b.1 FL confirmed cty map Jul 20 -----  
  
###Let's see whether we can make a map for MI about the confirmed cases over county popultation  
### at a particular date  
  
# set the date for map the end date you selected before  
date_for_map = as.Date("2020-7-20")  
  
##Start of state-specific code  
  
##you can make the daily death cases to be zero if it is smaller than zero  
#us_daily_death_clean[us_daily_death_clean<0] = 0  
  
# select the daily confirm cases for CA counties  
FL_daily_confirm = us_daily_confirm_clean%>%  
  filter(Province_State == "Florida")  
  
# This shape file contains the coordinates for county boundaries  
FL_shape_data = counties %>%  
  filter(state_name == "Florida")  
  
# get the population for CA counties  
FL_population = us_daily_death_clean %>%  
  filter(Province_State == "Florida")%>%  
  dplyr::select(Population)  
FL_population = as.numeric(FL_population$Population)  
  
# extract the daily confirmed cases on the selected date date_for_map  
FL_daily_confirm_selected = FL_daily_confirm[, c(1:11, 11+ which(all_dates == date_for_map))]  
  
# the Ratio = daily confirmed cases / population  
FL_daily_confirm_rate_selected = FL_daily_confirm_selected  
FL_daily_confirm_rate_selected[,12] = FL_daily_confirm_selected[,12]/FL_population  
FL_daily_confirm_rate_selected[,12][FL_daily_confirm_rate_selected[,12]==0]=NA  
colnames(FL_daily_confirm_rate_selected)[12] = "Ratio"  
  
##This is the county and rate  
#FL_daily_confirm_rate_selected[,c(11,12)]  
  
# joint the daily confirmed cases with the shape file  
FL_daily_confirm_rate_selected_joint <- left_join(FL_daily_confirm_rate_selected, FL_shape_data, by = "county_fips")  
  
# find the lower and upper limits of the ratio  
range(FL_daily_confirm_rate_selected$Ratio*100, na.rm = T)  
  
## [1] 0.007820579 0.271280886
```

```
# calculate the 7-day averaged daily confirm cases
FL_daily_confirm_avg = FL_daily_confirm
FL_daily_confirm_avg[, 13:dim(FL_daily_confirm)[2]] = t(apply(FL_daily_confirm[,13:dim(FL_daily_confirm)[2]], 1,
data_seven_day_smoothing))

# extract the averaged daily confirmed cases on the selected date
FL_daily_confirm_avg_selected = FL_daily_confirm_avg[, c(1:11, 11+ which(all_dates == date_for_map))]

# the Ratio = averaged daily confirmed cases / population
FL_daily_confirm_rate_avg_selected = FL_daily_confirm_avg_selected
FL_daily_confirm_rate_avg_selected[, 12] = FL_daily_confirm_avg_selected[, 12]/FL_population
FL_daily_confirm_rate_avg_selected[,12][FL_daily_confirm_rate_avg_selected[,12]==0]=NA
colnames(FL_daily_confirm_rate_avg_selected)[12] = "Ratio"

## county and rate
#FL_daily_confirm_rate_avg_selected[,c(11,12)]

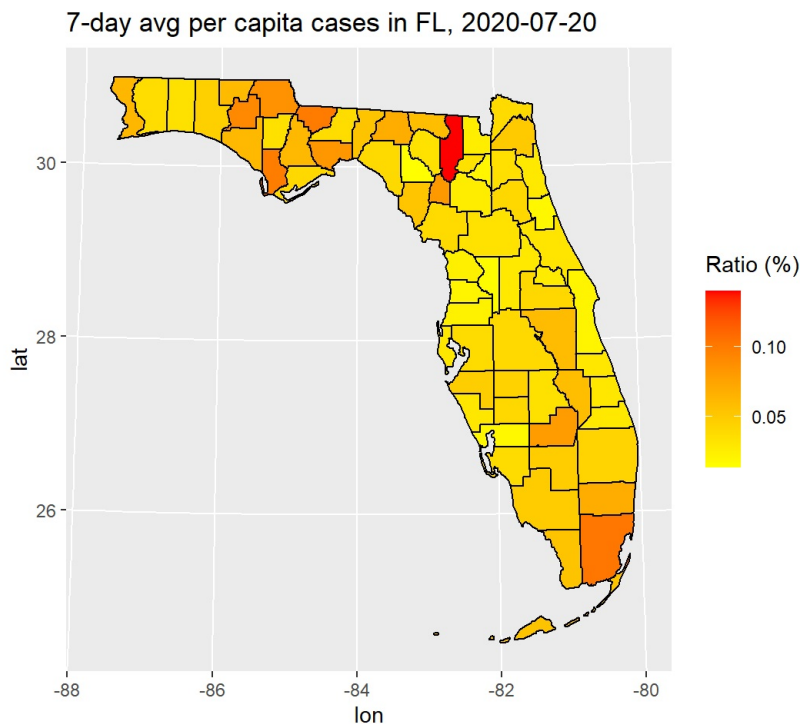
index_largest=which(FL_daily_confirm_rate_avg_selected[,c(12)]==max(FL_daily_confirm_rate_avg_selected[,c(12)]))
#FL_daily_confirm_rate_avg_selected[index_largest,c(11,12)]

# joint the averaged daily confirmed cases with the shape file
FL_daily_confirm_rate_avg_selected_joint <- left_join(FL_daily_confirm_rate_avg_selected, FL_shape_data, by = "county_fips")

# find the lower and upper limits of the ratio
range(FL_daily_confirm_rate_avg_selected$Ratio*100, na.rm = T)
```

```
## [1] 0.0135699 0.1404937
```

```
FL_daily_confirm_rate_avg_selected_joint %>%
ggplot(aes(long, lat, group = group, fill = Ratio*100)) +
scale_fill_gradient(low = "yellow", high = "red", na.value = "grey90")+
geom_polygon(col = "black") +
coord_map(projection = "albers", lat0 = 10, lat1 = 45) +
labs(fill = expression("Ratio (%)")) +
ggtitle(paste0("7-day avg per capita cases in FL", " ", date_for_map))+
xlab("lon") +ylab("lat" )
```





```
# 3.b.2 FL cty map Oct 22 -----
```

```
###Let's see whether we can make a map for MI about the confirmed cases over county population  
### at a particular date
```

```
# set the date for map the end date you selected before  
date_for_map = as.Date("2020-10-22")
```

```
##Start of state-specific code
```

```
##you can make the daily death cases to be zero if it is smaller than zero  
#us_daily_death_clean[us_daily_death_clean<0] = 0
```

```
# select the daily confirm cases for CA counties  
FL_daily_confirm = us_daily_confirm_clean%>%  
  filter(Province_State == "Florida")
```

```
# This shape file contains the coordinates for county boundaries  
FL_shape_data = counties %>%  
  filter(state_name == "Florida")
```

```
# get the population for CA counties  
FL_population = us_daily_death_clean %>%  
  filter(Province_State == "Florida")%>%  
  dplyr::select(Population)  
FL_population = as.numeric(FL_population$Population)
```

```
# extract the daily confirmed cases on the selected date date_for_map  
FL_daily_confirm_selected = FL_daily_confirm[, c(1:11, 11+ which(all_dates == date_for_map))]
```

```
# the Ratio = daily confirmed cases / population  
FL_daily_confirm_rate_selected = FL_daily_confirm_selected  
FL_daily_confirm_rate_selected[,12] = FL_daily_confirm_selected[,12]/FL_population  
FL_daily_confirm_rate_selected[,12][FL_daily_confirm_rate_selected[,12]==0]=NA  
colnames(FL_daily_confirm_rate_selected)[12] = "Ratio"
```

```
##This is the county and rate  
#FL_daily_confirm_rate_selected[,c(11,12)]
```

```
# joint the daily confirmed cases with the shape file  
FL_daily_confirm_rate_selected_joint <- left_join(FL_daily_confirm_rate_selected, FL_shape_data, by = "county_fips")
```

```
# find the lower and upper limits of the ratio  
range(FL_daily_confirm_rate_selected$Ratio*100, na.rm = T)
```

```
## [1] 0.005407451 0.123711340
```

```
# calculate the 7-day averaged daily confirm cases  
FL_daily_confirm_avg = FL_daily_confirm  
FL_daily_confirm_avg[, 13:dim(FL_daily_confirm)[2]] = t(apply(FL_daily_confirm[,13:dim(FL_daily_confirm)[2]], 1,  
data_seven_day_smoothing))
```

```
# extract the averaged daily confirmed cases on the selected date  
FL_daily_confirm_avg_selected = FL_daily_confirm_avg[, c(1:11, 11+ which(all_dates == date_for_map))]
```

```
# the Ratio = averaged daily confirmed cases / population  
FL_daily_confirm_rate_avg_selected = FL_daily_confirm_avg_selected  
FL_daily_confirm_rate_avg_selected[, 12] = FL_daily_confirm_avg_selected[, 12]/FL_population  
FL_daily_confirm_rate_avg_selected[,12][FL_daily_confirm_rate_avg_selected[,12]==0]=NA  
colnames(FL_daily_confirm_rate_avg_selected)[12] = "Ratio"
```

```
## county and rate  
#FL_daily_confirm_rate_avg_selected[,c(11,12)]
```

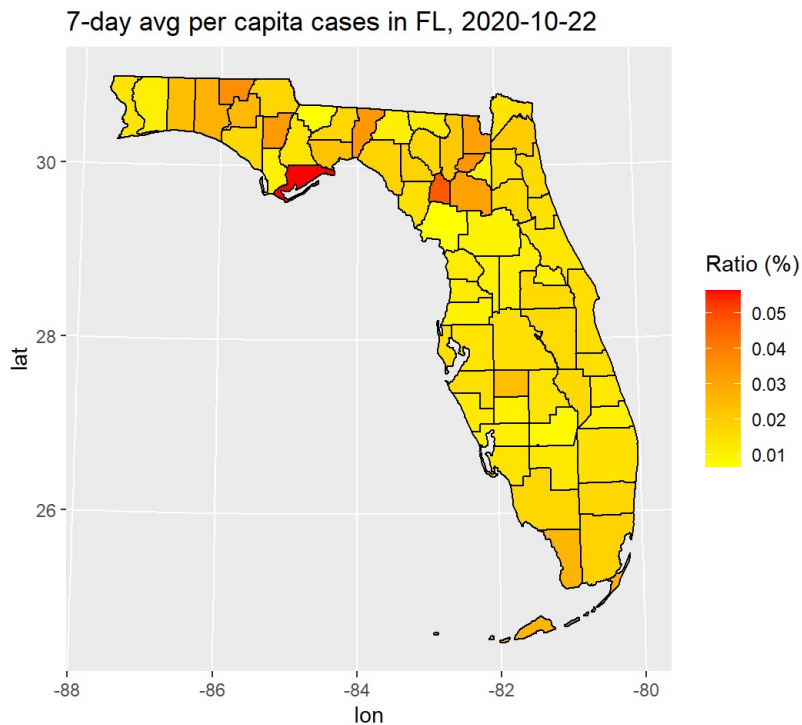
```
index_largest=which(FL_daily_confirm_rate_avg_selected[,c(12)]==max(FL_daily_confirm_rate_avg_selected[,c(12)]))  
#FL_daily_confirm_rate_avg_selected[index_largest,c(11,12)]
```

```
# joint the averaged daily confirmed cases with the shape file  
FL_daily_confirm_rate_avg_selected_joint <- left_join(FL_daily_confirm_rate_avg_selected, FL_shape_data, by = "county_fips")
```

```
# find the lower and upper limits of the ratio  
range(FL_daily_confirm_rate_avg_selected$Ratio*100, na.rm = T)
```

```
## [1] 0.006539975 0.056553756
```

```
FL_daily_confirm_rate_avg_selected_joint %>%
  ggplot(aes(long, lat, group = group, fill = Ratio*100)) +
  scale_fill_gradient(low = "yellow", high = "red", na.value = "grey90")+
  geom_polygon(col = "black") +
  coord_map(projection = "albers", lat0 = 10, lat1 = 45) +
  labs(fill = expression("Ratio (%)")) +
  ggtitle(paste0("7-day avg per capita cases in FL", " ", " ", date_for_map))+
  xlab("lon") +ylab("lat" )
```



On July 20, Miami-Dade and Columbia counties are the visual outliers. Many counties in the Northwestern area are also experiencing increased virus spreading rates.

On October 22, the northwestern area is still experiencing the outlying rates of virus transmission, namely Franklin and Gilchrist counties. Overall, virus transmission rates have plummeted since July 20.

3c)

```
# 2.b.1 MI 7-day cty map July 20 -----
```

```
###Let's see whether we can make a map for MI about the confirmed cases over county population  
### at a particular date
```

```
# set the date for map the end date you selected before  
date_for_map = as.Date("2020-7-20")
```

```
##Start of state-specific code
```

```
##you can make the daily death cases to be zero if it is smaller than zero  
#us_daily_death_clean[us_daily_death_clean<0] = 0
```

```
# select the daily confirm cases for CA counties  
MI_daily_confirm = us_daily_confirm_clean%>%  
  filter(Province_State == "Michigan")
```

```
# This shape file contains the coordinates for county boundaries  
MI_shape_data = counties %>%  
  filter(state_name == "Michigan")
```

```
# get the population for CA counties  
MI_population = us_daily_death_clean %>%  
  filter(Province_State == "Michigan")%>%  
  dplyr::select(Population)  
MI_population = as.numeric(MI_population$Population)
```

```
# extract the daily confirmed cases on the selected date date_for_map  
MI_daily_confirm_selected = MI_daily_confirm[, c(1:11, 11+ which(all_dates == date_for_map))]
```

```
# the Ratio = daily confirmed cases / population  
MI_daily_confirm_rate_selected = MI_daily_confirm_selected  
MI_daily_confirm_rate_selected[,12] = MI_daily_confirm_selected[,12]/MI_population  
MI_daily_confirm_rate_selected[,12][MI_daily_confirm_rate_selected[,12]==0]=NA  
colnames(MI_daily_confirm_rate_selected)[12] = "Ratio"
```

```
##This is the county and rate  
#MI_daily_confirm_rate_selected[,c(11,12)]
```

```
# joint the daily confirmed cases with the shape file  
MI_daily_confirm_rate_selected_joint <- left_join(MI_daily_confirm_rate_selected, MI_shape_data, by = "county_fips")
```

```
# find the lower and upper limits of the ratio  
range(MI_daily_confirm_rate_selected$Ratio*100, na.rm = T)
```

```
## [1] -0.002301337 0.024018666
```

```
# calculate the 7-day averaged daily confirm cases  
MI_daily_confirm_avg = MI_daily_confirm  
MI_daily_confirm_avg[, 13:dim(MI_daily_confirm)[2]] = t(apply(MI_daily_confirm[,13:dim(MI_daily_confirm)[2]], 1,  
data_seven_day_smoothing))
```

```
# extract the averaged daily confirmed cases on the selected date  
MI_daily_confirm_avg_selected = MI_daily_confirm_avg[, c(1:11, 11+ which(all_dates == date_for_map))]
```

```
# the Ratio = averaged daily confirmed cases / population  
MI_daily_confirm_rate_avg_selected = MI_daily_confirm_avg_selected  
MI_daily_confirm_rate_avg_selected[, 12] = MI_daily_confirm_avg_selected[, 12]/MI_population  
MI_daily_confirm_rate_avg_selected[,12][MI_daily_confirm_rate_avg_selected[,12]==0]=NA  
colnames(MI_daily_confirm_rate_avg_selected)[12] = "Ratio"
```

```
## county and rate  
#MI_daily_confirm_rate_avg_selected[,c(11,12)]
```

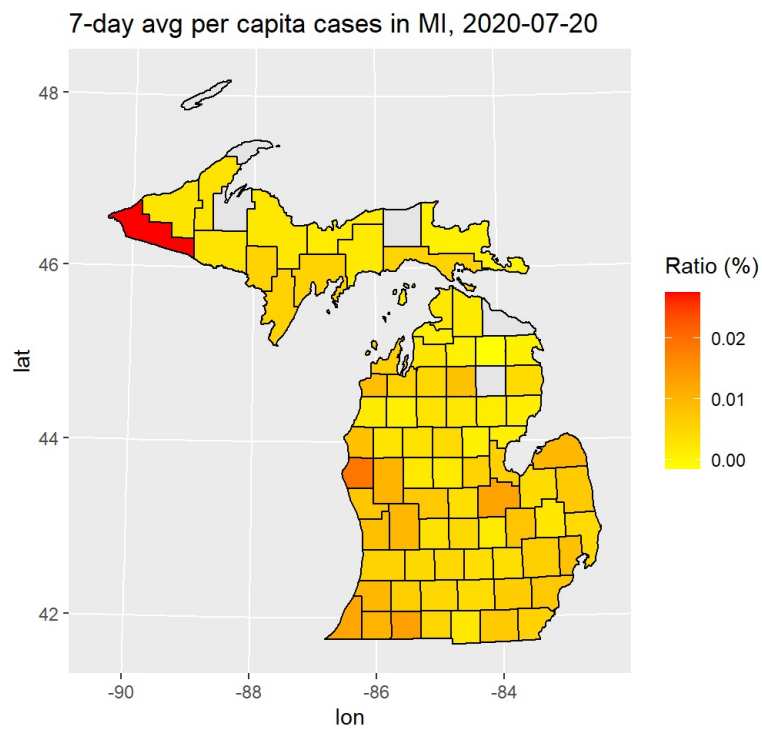
```
index_largest=which(MI_daily_confirm_rate_avg_selected[,c(12)]==max(MI_daily_confirm_rate_avg_selected[,c(12)]))  
#MI_daily_confirm_rate_avg_selected[index_largest,c(11,12)]
```

```
# joint the averaged daily confirmed cases with the shape file  
MI_daily_confirm_rate_avg_selected_joint <- left_join(MI_daily_confirm_rate_avg_selected, MI_shape_data, by = "county_fips")
```

```
# find the lower and upper limits of the ratio  
range(MI_daily_confirm_rate_avg_selected$Ratio*100, na.rm = T)
```

```
## [1] -0.001531487 0.027600307
```

```
MI_daily_confirm_rate_avg_selected_joint %>%  
  ggplot(aes(long, lat, group = group, fill = Ratio*100)) +  
  scale_fill_gradient(low = "yellow", high = "red", na.value = "grey90")+  
  geom_polygon(col = "black") +  
  coord_map(projection = "albers", lat0 = 10, lat1 = 45) +  
  labs(fill = expression("Ratio (%)")) +  
  ggtitle(paste0("7-day avg per capita cases in MI", " ", " ", date_for_map))+  
  xlab("lon") +ylab("lat" )
```



```
# 2.c.1 MI cty map Oct 22 -----
```

```
# set the date for map the end date you selected before
date_for_map = as.Date("2020-10-22")
```

```
##Start of state-specific code
```

```
##you can make the daily death cases to be zero if it is smaller than zero
#us_daily_death_clean[us_daily_death_clean<0] = 0
```

```
# select the daily confirm cases for CA counties
```

```
MI_daily_confirm = us_daily_confirm_clean%>%
  filter(Province_State == "Michigan")
```

```
# This shape file contains the coordinates for county boundaries
```

```
MI_shape_data = counties %>%
  filter(state_name == "Michigan")
```

```
# get the population for CA counties
```

```
MI_population = us_daily_death_clean %>%
  filter(Province_State == "Michigan")%>%
  dplyr::select(Population)
MI_population = as.numeric(MI_population$Population)
```

```
# extract the daily confirmed cases on the selected date date_for_map
```

```
MI_daily_confirm_selected = MI_daily_confirm[, c(1:11, 11+ which(all_dates == date_for_map))]
```

```
# the Ratio = daily confirmed cases / population
```

```
MI_daily_confirm_rate_selected = MI_daily_confirm_selected
MI_daily_confirm_rate_selected[,12] = MI_daily_confirm_selected[,12]/MI_population
MI_daily_confirm_rate_selected[,12][MI_daily_confirm_rate_selected[,12]==0]=NA
colnames(MI_daily_confirm_rate_selected)[12] = "Ratio"
```

```
##This is the county and rate
```

```
#MI_daily_confirm_rate_selected[,c(11,12)]
```

```
# joint the daily confirmed cases with the shape file
```

```
MI_daily_confirm_rate_selected_joint <- left_join(MI_daily_confirm_rate_selected, MI_shape_data, by = "county_fips")
```

```
# find the lower and upper limits of the ratio
```

```
range(MI_daily_confirm_rate_selected$Ratio*100, na.rm = T)
```

```
## [1] -0.006614632 0.150268336
```

```
# calculate the 7-day averaged daily confirm cases
```

```
MI_daily_confirm_avg = MI_daily_confirm
MI_daily_confirm_avg[, 13:dim(MI_daily_confirm)[2]] = t(apply(MI_daily_confirm[,13:dim(MI_daily_confirm)[2]], 1,
data_seven_day_smoothing))
```

```
# extract the averaged daily confirmed cases on the selected date
```

```
MI_daily_confirm_avg_selected = MI_daily_confirm_avg[, c(1:11, 11+ which(all_dates == date_for_map))]
```

```
# the Ratio = averaged daily confirmed cases / population
```

```
MI_daily_confirm_rate_avg_selected = MI_daily_confirm_avg_selected
MI_daily_confirm_rate_avg_selected[, 12] = MI_daily_confirm_avg_selected[, 12]/MI_population
MI_daily_confirm_rate_avg_selected[,12][MI_daily_confirm_rate_avg_selected[,12]==0]=NA
colnames(MI_daily_confirm_rate_avg_selected)[12] = "Ratio"
```

```
## county and rate
```

```
#MI_daily_confirm_rate_avg_selected[,c(11,12)]
```

```
index_largest=which(MI_daily_confirm_rate_avg_selected[,c(12)]==max(MI_daily_confirm_rate_avg_selected[,c(12)]))
```

```
#MI_daily_confirm_rate_avg_selected[index_largest,c(11,12)]
```

```
# joint the averaged daily confirmed cases with the shape file
```

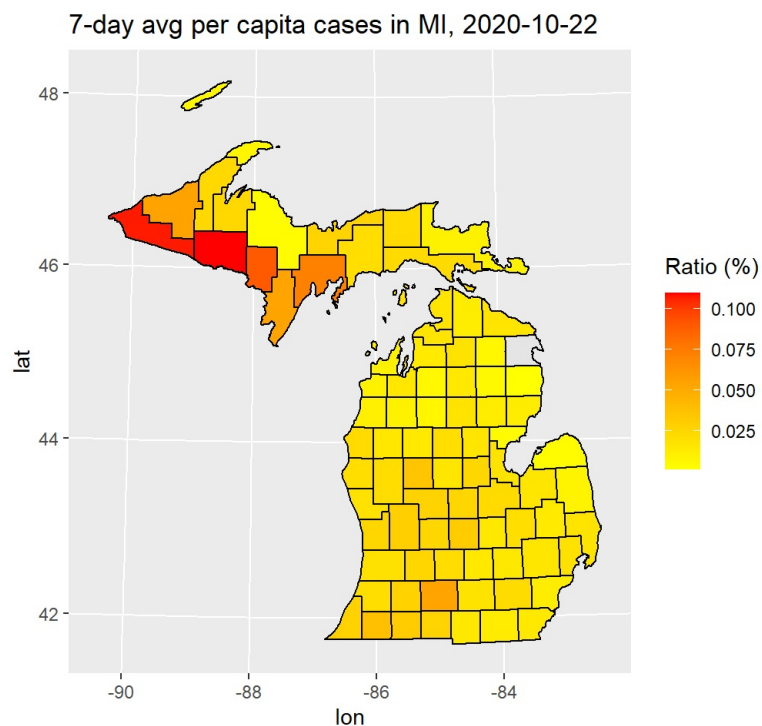
```
MI_daily_confirm_rate_avg_selected_joint <- left_join(MI_daily_confirm_rate_avg_selected, MI_shape_data, by = "county_fips")
```

```
# find the lower and upper limits of the ratio
```

```
range(MI_daily_confirm_rate_avg_selected$Ratio*100, na.rm = T)
```

```
## [1] 0.001372966 0.109731223
```

```
MI_daily_confirm_rate_avg_selected_joint %>%
  ggplot(aes(long, lat, group = group, fill = Ratio*100)) +
  scale_fill_gradient(low = "yellow", high = "red", na.value = "grey90")+
  geom_polygon(col = "black") +
  coord_map(projection = "albers", lat0 = 10, lat1 = 45) +
  labs(fill = expression("Ratio (%)")) +
  ggtitle(paste0("7-day avg per capita cases in MI", " ", " ", date_for_map))+
  xlab("lon") +ylab("lat" )
```



# 3.b.1 FL confirmed cty map Jul 20 -----

###Let's see whether we can make a map for MI about the confirmed cases over county population  
### at a particular date

# set the date for map the end date you selected before  
date\_for\_map = as.Date("2020-7-20")

##Start of state-specific code

##you can make the daily death cases to be zero if it is smaller than zero  
#us\_daily\_death\_clean[us\_daily\_death\_clean<0] = 0

# select the daily confirm cases for CA counties  
FL\_daily\_confirm = us\_daily\_confirm\_clean%>%  
 filter(Province\_State == "Florida")

# This shape file contains the coordinates for county boundaries  
FL\_shape\_data = counties %>%  
 filter(state\_name == "Florida")

# get the population for CA counties  
FL\_population = us\_daily\_death\_clean %>%  
 filter(Province\_State == "Florida")%>%  
 dplyr::select(Population)  
FL\_population = as.numeric(FL\_population\$Population)

# extract the daily confirmed cases on the selected date date\_for\_map  
FL\_daily\_confirm\_selected = FL\_daily\_confirm[, c(1:11, 11+ which(all\_dates == date\_for\_map))]

# the Ratio = daily confirmed cases / population  
FL\_daily\_confirm\_rate\_selected = FL\_daily\_confirm\_selected  
FL\_daily\_confirm\_rate\_selected[,12] = FL\_daily\_confirm\_selected[,12]/FL\_population  
FL\_daily\_confirm\_rate\_selected[,12][FL\_daily\_confirm\_rate\_selected[,12]==0]=NA  
colnames(FL\_daily\_confirm\_rate\_selected)[12] = "Ratio"

##This is the county and rate  
#FL\_daily\_confirm\_rate\_selected[,c(11,12)]

# joint the daily confirmed cases with the shape file  
FL\_daily\_confirm\_rate\_selected\_joint <- left\_join(FL\_daily\_confirm\_rate\_selected, FL\_shape\_data, by = "county\_fips")

# find the lower and upper limits of the ratio  
range(FL\_daily\_confirm\_rate\_selected\$Ratio\*100, na.rm = T)

## [1] 0.007820579 0.271280886

# calculate the 7-day averaged daily confirm cases  
FL\_daily\_confirm\_avg = FL\_daily\_confirm  
FL\_daily\_confirm\_avg[, 13:dim(FL\_daily\_confirm)[2]] = t(apply(FL\_daily\_confirm[,13:dim(FL\_daily\_confirm)[2]], 1,  
data\_seven\_day\_smoothing))

# extract the averaged daily confirmed cases on the selected date  
FL\_daily\_confirm\_avg\_selected = FL\_daily\_confirm\_avg[, c(1:11, 11+ which(all\_dates == date\_for\_map))]

# the Ratio = averaged daily confirmed cases / population  
FL\_daily\_confirm\_rate\_avg\_selected = FL\_daily\_confirm\_avg\_selected  
FL\_daily\_confirm\_rate\_avg\_selected[, 12] = FL\_daily\_confirm\_avg\_selected[, 12]/FL\_population  
FL\_daily\_confirm\_rate\_avg\_selected[,12][FL\_daily\_confirm\_rate\_avg\_selected[,12]==0]=NA  
colnames(FL\_daily\_confirm\_rate\_avg\_selected)[12] = "Ratio"

## county and rate  
#FL\_daily\_confirm\_rate\_avg\_selected[,c(11,12)]

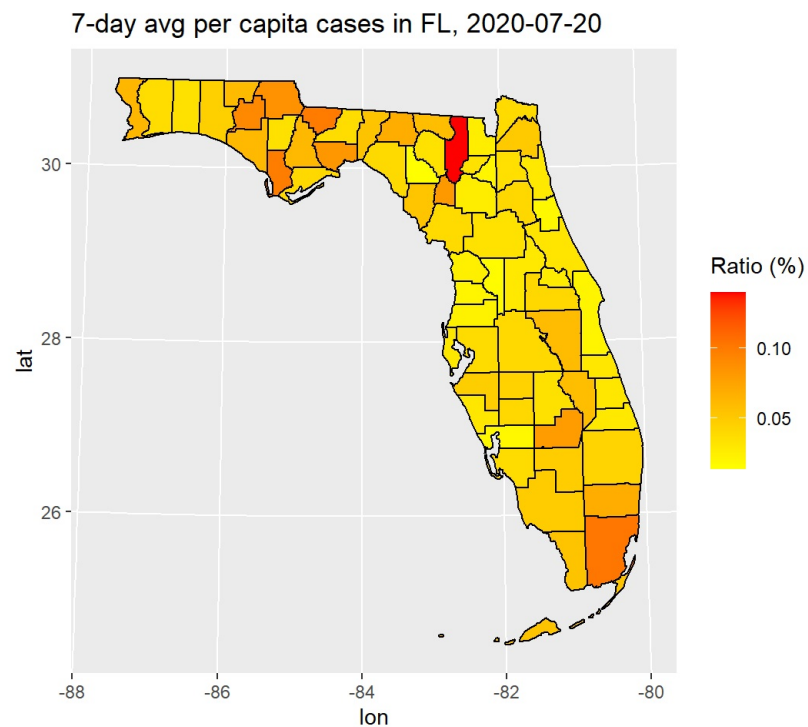
index\_largest=which(FL\_daily\_confirm\_rate\_avg\_selected[,c(12)]==max(FL\_daily\_confirm\_rate\_avg\_selected[,c(12)]))  
#FL\_daily\_confirm\_rate\_avg\_selected[index\_largest,c(11,12)]

# joint the averaged daily confirmed cases with the shape file  
FL\_daily\_confirm\_rate\_avg\_selected\_joint <- left\_join(FL\_daily\_confirm\_rate\_avg\_selected, FL\_shape\_data, by = "county\_fips")

# find the lower and upper limits of the ratio  
range(FL\_daily\_confirm\_rate\_avg\_selected\$Ratio\*100, na.rm = T)

```
## [1] 0.0135699 0.1404937
```

```
FL_daily_confirm_rate_avg_selected_joint %>%  
  ggplot(aes(long, lat, group = group, fill = Ratio*100)) +  
  scale_fill_gradient(low = "yellow", high = "red", na.value = "grey90")+  
  geom_polygon(col = "black") +  
  coord_map(projection = "albers", lat0 = 10, lat1 = 45) +  
  labs(fill = expression("Ratio (%)")) +  
  ggtitle(paste0("7-day avg per capita cases in FL", " ", " ", date_for_map))+  
  xlab("lon") +ylab("lat" )
```





```
# 3.b.2 FL cty map Oct 22 -----
```

```
###Let's see whether we can make a map for MI about the confirmed cases over county population  
### at a particular date
```

```
# set the date for map the end date you selected before  
date_for_map = as.Date("2020-10-22")
```

```
##Start of state-specific code
```

```
##you can make the daily death cases to be zero if it is smaller than zero  
#us_daily_death_clean[us_daily_death_clean<0] = 0
```

```
# select the daily confirm cases for CA counties  
FL_daily_confirm = us_daily_confirm_clean%>%  
  filter(Province_State == "Florida")
```

```
# This shape file contains the coordinates for county boundaries  
FL_shape_data = counties %>%  
  filter(state_name == "Florida")
```

```
# get the population for CA counties  
FL_population = us_daily_death_clean %>%  
  filter(Province_State == "Florida")%>%  
  dplyr::select(Population)  
FL_population = as.numeric(FL_population$Population)
```

```
# extract the daily confirmed cases on the selected date date_for_map  
FL_daily_confirm_selected = FL_daily_confirm[, c(1:11, 11+ which(all_dates == date_for_map))]
```

```
# the Ratio = daily confirmed cases / population  
FL_daily_confirm_rate_selected = FL_daily_confirm_selected  
FL_daily_confirm_rate_selected[,12] = FL_daily_confirm_selected[,12]/FL_population  
FL_daily_confirm_rate_selected[,12][FL_daily_confirm_rate_selected[,12]==0]=NA  
colnames(FL_daily_confirm_rate_selected)[12] = "Ratio"
```

```
##This is the county and rate  
#FL_daily_confirm_rate_selected[,c(11,12)]
```

```
# joint the daily confirmed cases with the shape file  
FL_daily_confirm_rate_selected_joint <- left_join(FL_daily_confirm_rate_selected, FL_shape_data, by = "county_fips")
```

```
# find the lower and upper limits of the ratio  
range(FL_daily_confirm_rate_selected$Ratio*100, na.rm = T)
```

```
## [1] 0.005407451 0.123711340
```

```
# calculate the 7-day averaged daily confirm cases  
FL_daily_confirm_avg = FL_daily_confirm  
FL_daily_confirm_avg[, 13:dim(FL_daily_confirm)[2]] = t(apply(FL_daily_confirm[,13:dim(FL_daily_confirm)[2]], 1,  
data_seven_day_smoothing))
```

```
# extract the averaged daily confirmed cases on the selected date  
FL_daily_confirm_avg_selected = FL_daily_confirm_avg[, c(1:11, 11+ which(all_dates == date_for_map))]
```

```
# the Ratio = averaged daily confirmed cases / population  
FL_daily_confirm_rate_avg_selected = FL_daily_confirm_avg_selected  
FL_daily_confirm_rate_avg_selected[, 12] = FL_daily_confirm_avg_selected[, 12]/FL_population  
FL_daily_confirm_rate_avg_selected[,12][FL_daily_confirm_rate_avg_selected[,12]==0]=NA  
colnames(FL_daily_confirm_rate_avg_selected)[12] = "Ratio"
```

```
## county and rate  
#FL_daily_confirm_rate_avg_selected[,c(11,12)]
```

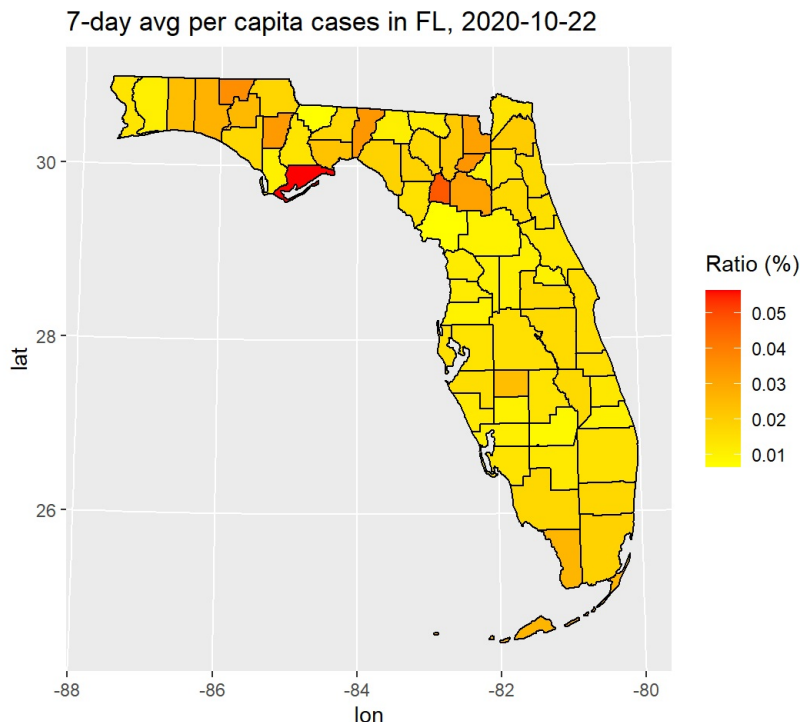
```
index_largest=which(FL_daily_confirm_rate_avg_selected[,c(12)]==max(FL_daily_confirm_rate_avg_selected[,c(12)]))  
#FL_daily_confirm_rate_avg_selected[index_largest,c(11,12)]
```

```
# joint the averaged daily confirmed cases with the shape file  
FL_daily_confirm_rate_avg_selected_joint <- left_join(FL_daily_confirm_rate_avg_selected, FL_shape_data, by = "county_fips")
```

```
# find the lower and upper limits of the ratio  
range(FL_daily_confirm_rate_avg_selected$Ratio*100, na.rm = T)
```

```
## [1] 0.006539975 0.056553756
```

```
FL_daily_confirm_rate_avg_selected_joint %>%  
  ggplot(aes(long, lat, group = group, fill = Ratio*100)) +  
  scale_fill_gradient(low = "yellow", high = "red", na.value = "grey90")+  
  geom_polygon(col = "black") +  
  coord_map(projection = "albers", lat0 = 10, lat1 = 45) +  
  labs(fill = expression("Ratio (%)")) +  
  ggtitle(paste0("7-day avg per capita cases in FL", " ", " ", date_for_map))+  
  xlab("lon") +ylab("lat" )
```



Michigan is worse off on October 22, 2020. It has many regions hovering around the .05% cases per capita range and concentrated areas of .1% cases per capita. Meanwhile, Florida's state-worst Franklin County is no more than a .07% case per capita. Most counties are under the .04% rate.

One possible reason for this could be Michigan's colder climate pushing many unavoidable business interactions in-doors, where the virus collects in the air. A reduced rate of mask participation could also cause the discrepancy, although it is unsure what the relative mask participation rates are between Florida and Michigan. The simplest explanation probably is: this is the virus' first wave in Michigan, while Florida is between its first and second waves of infections.

3d) As of October 22, I would be eyeing the northwestern counties and Miami-Dade county. The former have consistently had more infections than other regions in the state and the latter is at the highest risk for a second wave due to its concentrated population. I would implement some of the tamer parts of my Michigan plan: a specific travel advisory warning about travel to the hottest regions (including Franklin County) should be sent out and, to the degree allowed by the budget, I will make testing more widely available.

4a)

```
# 4.a.1 SB cumulative cases Mar21-Oct31 -----

start_date = as.Date("2020-3-21")

end_date = as.Date("2020-10-31")

date_selected=seq.Date(start_date, end_date, by=1)

####county level analysis
##let's look at Santa Barbara
state_name = "California"
state_name_short = "CA"
county_name = "Santa Barbara"

##get the death and confirmed cases
county_death = us_death%>%
  filter(Admin2 == county_name, Province_State == state_name) %>%
  select(starts_with("x"))
county_confirmed = us_confirm %>%
  filter(Admin2 == county_name, Province_State == state_name) %>%
  select(starts_with("x"))

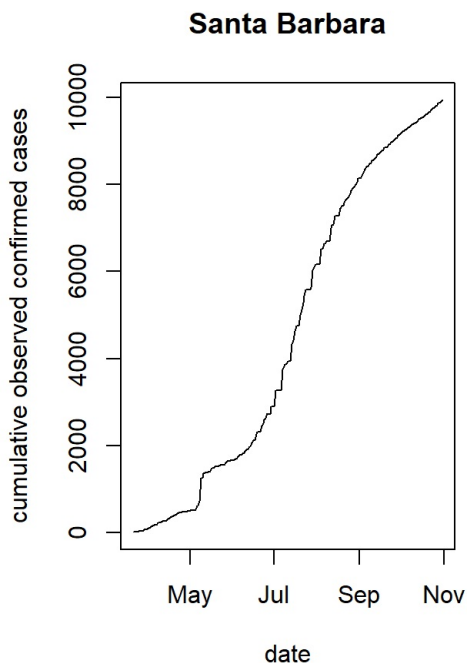
county_death_sum = apply(county_death, 2, sum)

county_confirmed_sum = apply(county_confirmed, 2, sum)

county_death_selected = county_death_sum[which(all_dates %in% seq.Date(start_date, end_date, by=1))]
county_confirmed_selected = county_confirmed_sum[which(all_dates %in% seq.Date(start_date, end_date, by=1))]

county_death_selected=as.numeric(county_death_selected)
county_confirmed_selected=as.numeric(county_confirmed_selected)

par(mfrow=c(1,2))
plot(date_selected,county_confirmed_selected,xlab='date',ylab='cumulative observed confirmed cases',main=county_name,type='l')
```



```
#a1,2, run with part 1
```

```
# 4.a.2 SB daily cases Mar21-Oct31 -----
```

```
##You may plot the daily confirmed by histogram
```

```
county_confirmed_selected_daily=county_confirmed_selected[2:length(county_confirmed_selected)]-county_confirmed_selected[1:(length(county_confirmed_selected)-1)]  
daily_date_selected=date_selected[2:length(date_selected)]
```

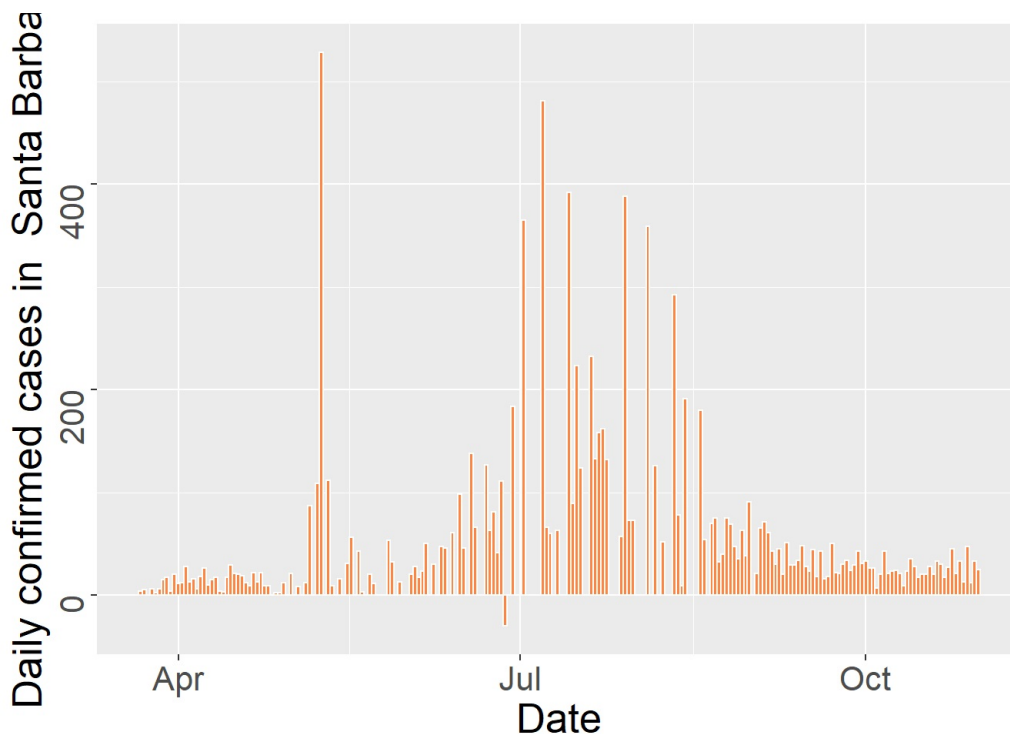
```
daily_confirmed_county_df = data.frame(date = daily_date_selected, value = county_confirmed_selected_daily)
```

```
###plot the daily confirmed cases
```

```
##you can save it as a png
```

```
#png(filename = paste0(file_path, "US_daily_confirmed_cases.png"), width = 900, height = 600)
```

```
daily_confirmed_county_df %>%  
  ggplot(aes(x=date, y=value)) +  
  geom_bar(stat = 'identity', color="white", fill="#ff8540", width = 1) +  
  ylab("Daily confirmed cases in Santa Barbara")+  
  xlab("Date")+  
  theme(text = element_text(size = 20),  
        legend.title = element_text(size = 15),  
        legend.text = element_text(size = 15),  
        legend.key.width=unit(1,"cm"),  
        axis.text.y = element_text(angle=90, hjust=1))
```



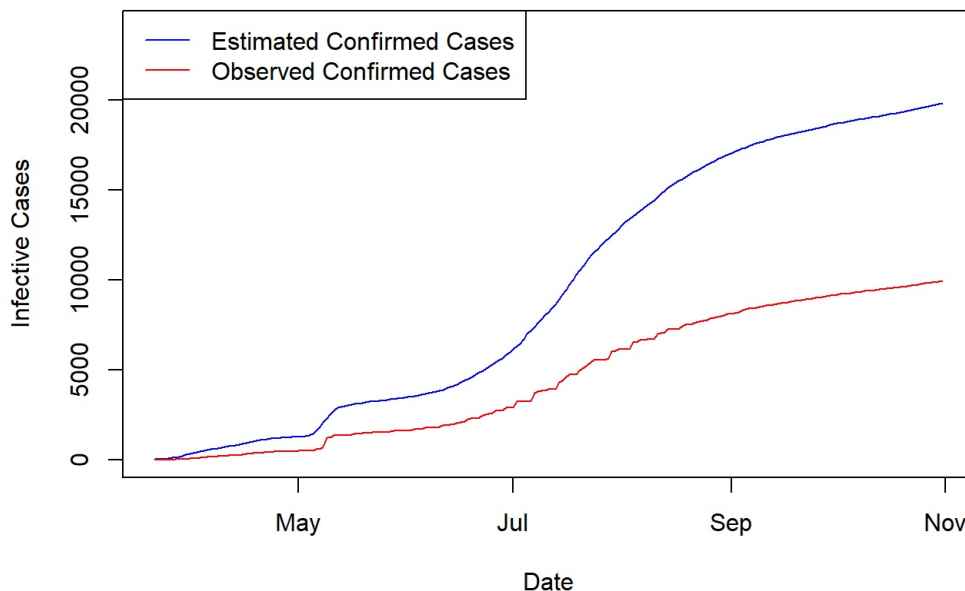
```
##Load Part 2 Environment
```

```
load(file = "C:/Users/David/Desktop/Pstat 120c/120c Report 2/Part2Enviro.RData")
```

```
# 4.a.3 SB fitted SIR Mar21-Oct31 -----
```

```
plot(N-param_record_approx_for_beta[1,]-date_seq, ylim=c(0,24000), type="l", col="blue", xlab = "Date", ylab = "Infective Cases", main = paste0(county_names[each_index], ", population=", round(N/10^6,2),"M", ", Ratio = ", round(ratio_real,3)))  
lines(confirm_selected-date_seq, type="l", col="red")  
legend("topleft", legend = c("Estimated Confirmed Cases", "Observed Confirmed Cases"), lty = c(1,1), col = c("blue", "red"))
```

## Santa Barbara, population=0.45M, Ratio = 0.422



```
##this is the estimated percentage between people infected and confirmed
#confirm_selected[n]/(N-param_record_approx_for_beta[1,n])
```

```
# 4.a.4 Forecast - observed cases SB -----
```

```
#Model prediction of cases by Oct31
N-param_record_approx_for_beta[1, length(date_seq)]
```

```
## [1] 19843.37
```

```
#Calculate difference between model prediction and actual cases
undetected_cases = N-param_record_approx_for_beta[1, length(date_seq)] - confirm_selected[length(date_seq)]
undetected_cases
```

```
## [1] 9899.371
```

Based on the fit, 19,843 individuals have contracted the virus by Oct 31. By comparing this to the observed cases, the model predicts that 9,899 individuals have contracted the virus and have not been counted as an observed case.

4b)

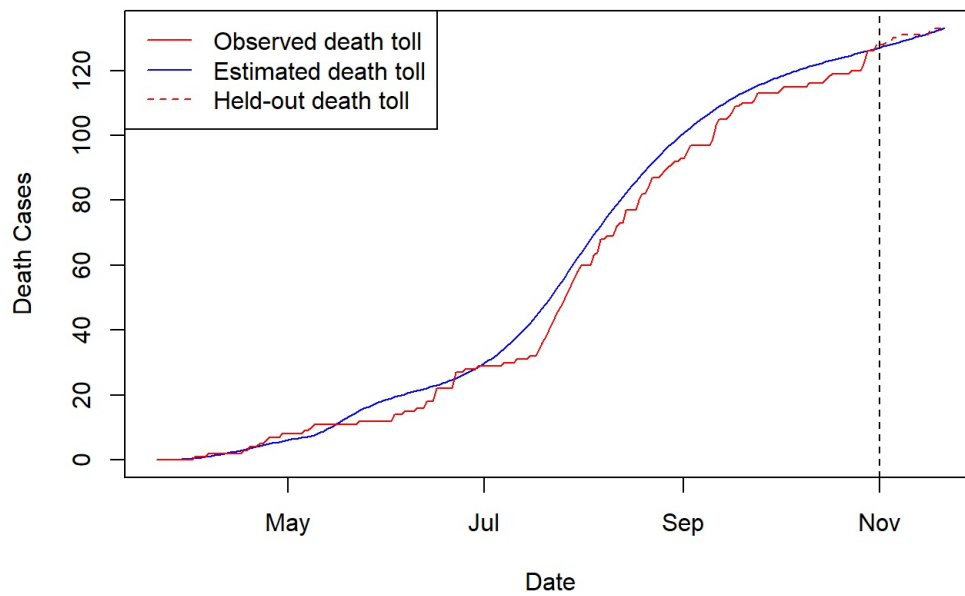
```
# 4.b.1 SB Sim death Nov1-21 -----
```

```
###fitted death and forecast death
date_seq_all = seq.Date(start_date, end_date+prediction_length, by=1)

ylimit_death_all = c(min(param_record_approx_all[4,], death_selected), max(param_record_approx_all[4,], death_selected))

plot(param_record_approx_all[4,]-date_seq_all,ylim = ylimit_death_all, type="l", col="blue", xlab = "Date", ylab = "Death Cases", main = paste0(county_names[each_index], ", population=", round(N/10^6,2),"M", ", Ratio = ", round(ratio_real,3)))
lines(death_selected~date_seq, col = "red")
lines(date_seq_all[(n+1):(n+prediction_length-1)],death_with_county_names_all[[1]][each_index,(n+1):(n+prediction_length-1)],
      col = "red",lty=2)
abline(v = end_date+1, lty=2)
legend("topleft", legend = c("Observed death toll", "Estimated death toll","Held-out death toll"), lty = c(1,1,2),
      col = c("red", "blue",'red'))
```

### Santa Barbara, population=0.45M, Ratio = 0.422



```
#Calculate predicted deaths by Nov21
paste("Predicted deaths by Nov 21:", param_record_approx_all[4,length(date_seq_all)])
```

```
## [1] "Predicted deaths by Nov 21: 133.056231107681"
```

## ##Load Part 1 Environment

```
load(file = "C:/Users/David/Desktop/Pstat 120c/120c Report 2/Part1Enviro.RData")
```

```
#Real death toll Nov1-Nov21
```

```
#b2, run with part 1
```

```
start_date = as.Date("2020-11-1")
```

```
end_date = as.Date("2020-11-21")
```

```
date_selected=seq.Date(start_date, end_date, by=1)
```

```
#####county level analysis
```

```
##let's look at Santa Barbara
```

```
state_name = "California"
```

```
state_name_short = "CA"
```

```
county_name = "Santa Barbara"
```

```
##get the death and confirmed cases
```

```
county_death = us_death%>%
```

```
  filter(Admin2 == county_name, Province_State == state_name) %>%
```

```
  select(starts_with("x"))
```

```
county_confirmed = us_confirm %>%
```

```
  filter(Admin2 == county_name, Province_State == state_name) %>%
```

```
  select(starts_with("x"))
```

```
county_death_sum = apply(county_death, 2, sum)
```

```
county_confirmed_sum = apply(county_confirmed, 2, sum)
```

```
county_death_selected = county_death_sum[which(all_dates %in% seq.Date(start_date, end_date, by=1))]
```

```
county_confirmed_selected = county_confirmed_sum[which(all_dates %in% seq.Date(start_date, end_date, by=1))]
```

```
county_death_selected=as.numeric(county_death_selected)
```

```
county_confirmed_selected=as.numeric(county_confirmed_selected)
```

```
##plot the data
```

```
##There is a jump in death on July 31, see the news:
```

```
##https://www.ksby.com/news/coronavirus/santa-barbara-co-announces-28-previously-unreported-covid-19-related-deaths-discovered-in-data-review
```

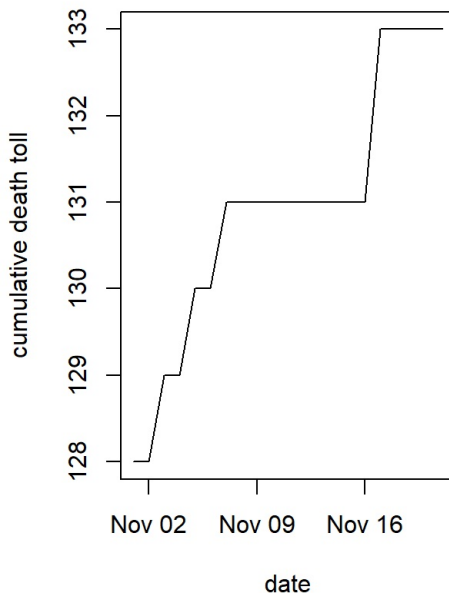
```
par(mfrow=c(1,2))
```

```
#We can also plot cumulative deaths but we don't want that here.
```

```
plot(date_selected, county_death_selected, xlab='date', ylab='cumulative death toll', main=county_name, type='l')
```

```
real_death_Nov = 133
```

## Santa Barbara



The forecasted cumulative deaths and observed cumulative deaths are actually equal at the end of the forecasted period, both at 133 deaths. The observed cumulative deaths increased more in spurts while the forecast was a more gradual increase, reflecting the variability of real life and data collection.

4c)

```
##Load Part 2 Environment

load(file = "C:/Users/David/Desktop/Pstat 120c/120c Report 2/Part2Enviro.RData")

# 4.c.1 5% change in model -----

###work on some simulation if the infectious period decreases
gamma_new = 1/4.75 ###suppose it changes from 5 day to 4.75 days

param_record_approx_for_beta_new = matrix(0, 5, n) # 5 rows: S_t, I_t, R_t, D_t, C_t
param_record_approx_for_beta_new[,1] = init_for_beta
param_record_approx_for_beta_new[1,] = S_t_seq

# record the value of transmission rate
# approx_beta_seq_new = rep(0, n-1)
# we should fix the beta when we change the gamma parameter

# iterative approach for calculating the seq of compartments in SIRDC
for (i in 1:(n-1)){
  S_t_1 = param_record_approx_for_beta_new[1,i]
  I_t_1 = param_record_approx_for_beta_new[2,i]
  R_t_1 = param_record_approx_for_beta_new[3,i]
  D_t_1 = param_record_approx_for_beta_new[4,i]
  C_t_1 = param_record_approx_for_beta_new[5,i]

  beta_t_1_2 = approx_beta_seq[i]

  if(I_t_1<1){
    I_t_1 = 1
  }

  S_t_2 = uniroot(find_root_S_t_2, c(0, N), tol = 0.0001, param = c(S_t_1, beta_t_1_2, I_t_1), N = N, gamma=gamma_new)
  I_t_2 = I_t_1 * exp(beta_t_1_2*(S_t_1 + S_t_2$root)/(2*N) - gamma_new)
  R_t_2 = (2-theta)/(2+theta)*R_t_1 + gamma_new/(2+theta)*(I_t_1+I_t_2)
  D_t_2 = D_t_1 + delta*theta*(R_t_1+R_t_2)/2
  C_t_2 = C_t_1 + (1-delta)*theta*(R_t_1+R_t_2)/2

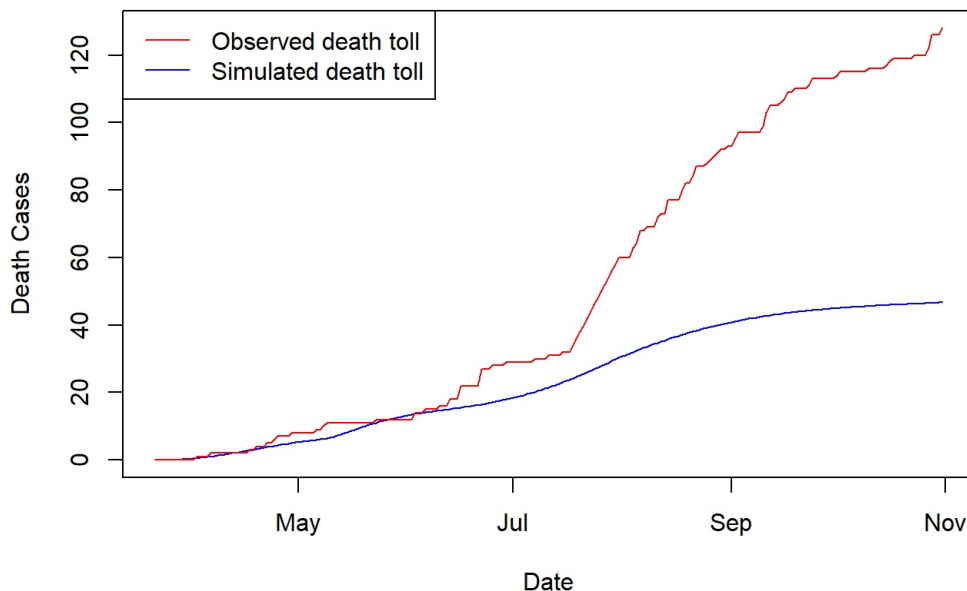
  param_record_approx_for_beta_new[, i+1] = c(S_t_2$root, I_t_2, R_t_2, D_t_2, C_t_2)
}

# calculate the smoothed transmission rate using 7 day average
# approx_beta_seq_smoothed_new = rollapply(approx_beta_seq_new, width = 7, by = 1, FUN = mean, align = "left")

###plot the simulated death
plot(param_record_approx_for_beta_new[4,]-date_seq,ylim = ylimit_death, type="l", col="blue", xlab = "Date", ylab = "Death Cases", main = paste0(county_names[each_index], ", population=", round(N/10^6,2),"M", ", Ratio = ", round(ratio_real,3)))
lines(death_selected~date_seq, col = "red")
legend("topleft", legend = c("Observed death toll", "Simulated death toll"), lty = c(1,1), col = c("red", "blue"))
}
```



## Santa Barbara, population=0.45M, Ratio = 0.422



```
# 4.c.2 Lives saved from 5% reduction -----
```

```
death_toll = param_record_approx_for_beta_new[4, length(date_seq)]  
death_toll
```

```
## [1] 46.69298
```

```
##Load Part 1 Environment
```

```
load(file = "C:/Users/David/Desktop/Pstat 120c/120c Report 2/Part1Enviro.RData")
```

```
#c2, run with part 1
```

```
# 4.c.2 SB cumulative death Oct 31-----
```

```
start_date = as.Date("2020-10-31")
```

```
end_date = as.Date("2020-10-31")
```

```
date_selected=seq.Date(start_date, end_date, by=1)
```

```
#####county level analysis
```

```
##let's look at Santa Barbara
```

```
state_name = "California"
```

```
state_name_short = "CA"
```

```
county_name = "Santa Barbara"
```

```
##get the death and confirmed cases
```

```
county_death = us_death%>%
```

```
  filter(Admin2 == county_name, Province_State == state_name) %>%
```

```
  select(starts_with("x"))
```

```
county_confirmed = us_confirm %>%
```

```
  filter(Admin2 == county_name, Province_State == state_name) %>%
```

```
  select(starts_with("x"))
```

```
county_death_sum = apply(county_death, 2, sum)
```

```
county_confirmed_sum = apply(county_confirmed, 2, sum)
```

```
county_death_selected = county_death_sum[which(all_dates %in% seq.Date(start_date, end_date, by=1))]
```

```
county_confirmed_selected = county_confirmed_sum[which(all_dates %in% seq.Date(start_date, end_date, by=1))]
```

```
county_death_selected=as.numeric(county_death_selected)
```

```
county_confirmed_selected=as.numeric(county_confirmed_selected)
```

```
paste("There were ", county_death_selected, "deaths by ", end_date)
```

```
## [1] "There were 128 deaths by 2020-10-31"
```

```
real_death_Oct = 128
```

The simulated death toll from March 21 to Oct 31 at  $\gamma = 4.75$  is 47 (rounded up from 46.69298). The observed deaths during that period were 128. Therefore, a 5% reduction in infectious period corresponds to a reduction of 81 deaths in our SIRDC model.

4d)

```
##Load Part 2 Environment
```

```
load(file = "C:/Users/David/Desktop/Pstat 120c/120c Report 2/Part2Enviro.RData")
```

```
###work on some simulation if the infectious period decreases
```

```
gamma_new = 1/4.5 ###suppose it changes from 5 day to 4.75 days
```

```
param_record_approx_for_beta_new = matrix(0, 5, n) # 5 rows: S_t, I_t, R_t, D_t, C_t
```

```
param_record_approx_for_beta_new[,1] = init_for_beta
```

```
param_record_approx_for_beta_new[1,] = S_t_seq
```

```
# record the value of transmission rate
```

```
# approx_beta_seq_new = rep(0, n-1)
```

```
# we should fix the beta when we change the gamma parameter
```

```
# iterative approach for calculating the seq of compartments in SIRDC
```

```
for (i in 1:(n-1)){
```

```
  S_t_1 = param_record_approx_for_beta_new[1,i]
```

```
  I_t_1 = param_record_approx_for_beta_new[2,i]
```

```
  R_t_1 = param_record_approx_for_beta_new[3,i]
```

```
  D_t_1 = param_record_approx_for_beta_new[4,i]
```

```
  C_t_1 = param_record_approx_for_beta_new[5,i]
```

```
  beta_t_1_2 = approx_beta_seq[i]
```

```
  if(I_t_1<1){
```

```
    I_t_1 = 1
```

```
  }
```

```
  S_t_2 = uniroot(find_root_S_t_2, c(0, N), tol = 0.0001, param = c(S_t_1, beta_t_1_2, I_t_1), N = N, gamma=gamma_new)
```

```
  I_t_2 = I_t_1 * exp(beta_t_1_2*(S_t_1 + S_t_2$root)/(2*N) - gamma_new)
```

```
  R_t_2 = (2-theta)/(2+theta)*R_t_1 + gamma_new/(2+theta)*(I_t_1+I_t_2)
```

```
  D_t_2 = D_t_1 + delta*theta*(R_t_1+R_t_2)/2
```

```
  C_t_2 = C_t_1 + (1-delta)*theta*(R_t_1+R_t_2)/2
```

```
  param_record_approx_for_beta_new[, i+1] = c(S_t_2$root, I_t_2, R_t_2, D_t_2, C_t_2)
```

```
}
```

```
# calculate the smoothed transmission rate using 7 day average
```

```
# approx_beta_seq_smoothed_new = rollapply(approx_beta_seq_new, width = 7, by = 1, FUN = mean, align = "left")
```

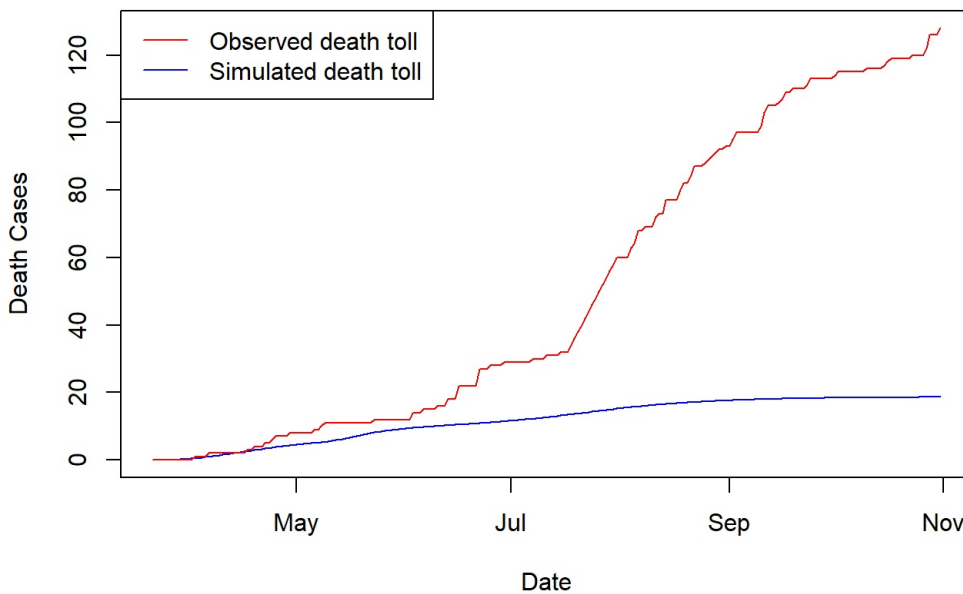
```
###plot the simulated death
```

```
plot(param_record_approx_for_beta_new[4,]~date_seq,ylim = ylimit_death, type="l", col="blue", xlab = "Date", ylab = "Death Cases", main = paste0(county_names[each_index], ", population=", round(N/10^6,2),"M", ", Ratio = ", round(ratio_real,3)))
```

```
lines(death_selected~date_seq, col = "red")
```

```
legend("topleft", legend = c("Observed death toll", "Simulated death toll"), lty = c(1,1), col = c("red", "blue"))
```

## Santa Barbara, population=0.45M, Ratio = 0.422



```
# 4.d.2 Lives saved from 10% reduction-----
death_toll = param_record_approx_for_beta_new[4, length(date_seq)]
death_toll
```

```
## [1] 18.64699
```

The simulated death toll from March 21 to Oct 31 at  $\gamma = 4.5$  is 19 (rounded up from 18.64699). The observed deaths during that period were 128. Therefore, a 10% reduction in infectious period corresponds to a reduction of 109 deaths in our SIRDC model.

4e)

Barring a mutation in Covid-19, the only methods we have to reduce the infectious period of the disease are medical treatments and bureaucratic controls. In the former, we can hope for development or adaptation of a drug to reduce the incubation period or length of the symptomatic period, which would depress the overall length of the infectious period. I'm sure many compounds are being looked at hopefully in labs around the world. In the latter, the government could impose shift work where half of the population is ordered to stay at home while the other half is allowed to work, then switching these assignments every week. Bureaucratic controls like these have been implemented in some schools, to unknown effects. The idea is to cut into the 14-day incubation period by halting transmission in the last 7 days of this period. There might be a more nuanced and/or effective type of bureaucratic control than this, which should be studied in the coming months. Perhaps shifts of 3,4, or 5 days might be more effective than weekly shifts.

5a) In the SIRDC model, if an effective vaccine is widely administered, the susceptible population compartment reduces. The amount of susceptible population that would become a part of the recovered population is  $(\text{susceptible population})(\text{vaccine rate})(\text{efficacy rate})$ .