625 Final Project

Descriptive Clean

Yubo SHAO

2021-11-30

Project Targets

1. **Visualization**:

   - State-level incidence & mortality rate (48 + 1; doesn't include Hawaii, Alaska, Virgin Islands, Puerto Rico and Northern Mariana Islands);

     States are divided into "blue state" (Democratic Party) and "red state" (Republican Party). "Rate" will be shown in the shades of the color.

     Use animation; on monthly data;

   - Democratic and Republican states incidence rate and mortality rate trend, by line plot. (48 + 1 states; 319 days)

2. **Prediction Model**:

   - Use random forest;
   - Predictors: 55

     Demographic: (1) percent female; (2) percent black; (3) percent_asian; (3) percent_hispanic; (4) percent minorities;

     Health related: (1) "poor/fair health" rate; (2) average No. of physically unhealthy days; (3) average No. of mentally unhealthy days; (4) smoking rate; (5) adult obesity rate; (6) food environment index; (7) physically inactive rate (8) access to exercise opportunities rate; (9) excessive drinking rate; (10) chlamydia rate; (11) life expectancy; (12) percent frequent physical distress; (13) percent frequent mental distress; (14) percent adults with diabetes; (15) hiv prevalence rate; (16) percent food insecure; (17) percent limited access to healthy foods; (18) drug overdose mortality rate; (19) percent insufficient sleep; (20) percent disabled; (21) percent low birthweight;

     Socioeconomic status related: (1) teen birth rate; (2) uninsured rate; (3) primary care physicians rate; (4) dentist rate; (5) mental health provider rate; (6) preventable hospitalization rate; (7) vaccinated rate; (8) high school graduation rate; (9) unemployed CHR rate; (10) percent children in poverty; (11) percent single parent households CHR; (12) social association rate; (13) violent crime rate; (14) injury death rate; (15) percent severe housing problems; (16) severe housing cost burden; (17) inadequate facilities; (18) percent long commute drives alone; (19) median household income; (20) firearm fatalities rate; (21) juvenile arrest rate; (22) percent homeowners; (23) percent no vehicle; (24) child mortality rate; (25) infant mortality rate; (26) motor vehicle mortality rate; (27) segregation index; (28) percent limited english abilities;

     Environmental related: (1) average daily pm 2.5; (2) presence of water violation;

   - Outcome: Incidence rate & mortality rate of each county.
   - County-level data (3106 unique counties);

Data Preprocessing

Import the dataset

`data.table::fread` runs much faster than `readr::read_csv`, because it can use all possible threads.

```
> data.table::setDTthreads(2)
> load.data <- fread("US_counties_COVID19_health_weather_data.csv")
```

Select 48 + 1 states

- Exclude: Hawaii, Alaska, Virgin Islands, Puerto Rico and Northern Mariana Islands

```
> state.names <- as.vector(unique(load.data$state))
> exclude.state.names <- c("Hawaii", "Alaska", "Virgin Islands",
+                          "Puerto Rico", "Northern Mariana Islands")
> selected.state.names <- state.names[(state.names %in% exclude.state.names) == FALSE]
> selected.state.data <- load.data[which(load.data$state %in% selected.state.names == TRUE), ]
```

Select Features

```
> data.initial <-
+   selected.state.data[, c(
+     "date",
+     "county",
+     "state",
+     "total_population",
+     "cases",
+     "deaths",
+     "percent_fair_or_poor_health",
+     "average_number_of_physically_unhealthy_days",
+     "average_number_of_mentally_unhealthy_days",
+     "percent_low_birthweight",
+     "percent_smokers",
+     "percent_adults_with_obesity",
+     "food_environment_index",
+     "percent_physically_inactive",
+     "percent_with_access_to_exercise_opportunities",
+     "percent_excessive_drinking",
+     "chlamydia_rate",
+     "teen_birth_rate",
+     "percent_uninsured",
+     "primary_care_physicians_rate",
+     "dentist_rate",
+     "mental_health_provider_rate",
+     "preventable_hospitalization_rate",
+     "percent_vaccinated",
+     "high_school_graduation_rate",
+     "percent_unemployed_CHR",
+     "percent_children_in_poverty",
+     "percent_single_parent_households_CHR",
+     "social_association_rate",
+     "violent_crime_rate",
+     "injury_death_rate",
+     "presence_of_water_violation",
```

```
+     "average_daily_pm2_5",
+     "percent_severe_housing_problems",
+     "severe_housing_cost_burden",
+     "inadequate_facilities",
+     "percent_long_commute_drives_alone",
+     "life_expectancy",
+     "child_mortality_rate",
+     "infant_mortality_rate",
+     "percent_frequent_physical_distress",
+     "percent_frequent_mental_distress",
+     "percent_adults_with_diabetes",
+     "hiv_prevalence_rate",
+     "percent_food_insecure",
+     "percent_limited_access_to_healthy_foods",
+     "drug_overdose_mortality_rate",
+     "motor_vehicle_mortality_rate",
+     "percent_insufficient_sleep",
+     "median_household_income",
+     "segregation_index",
+     "homicide_rate",
+     "firearm_fatalities_rate",
+     "juvenile_arrest_rate",
+     "percent_homeowners",
+     "percent_black",
+     "percent_asian",
+     "percent_hispanic",
+     "percent_female",
+     "percent_rural",
+     "percent_disabled",
+     "percent_minorities",
+     "percent_no_vehicle",
+     "percent_limited_english_abilities"
+   )]
```

Unique counties

```
> ## 3106 unique counties
> data.unique1 <-  unique(data.initial %>%
+                           select(-c("date", "cases", "deaths")))
> data.unique2 <- unique(data.unique1 %>% select(c("county", "state")))
> nrow(data.unique2)
## [1] 3106
```

- *Note: Some counties in different states might have the same name:*

```
> kable(data.unique2[data.unique2$county ==  "Adams"], align = c("c", "c"),
+       caption = "'Adams' county in different state")
```

Table 1: 'Adams' county in different state

county	state
Adams	Indiana
Adams	Colorado
Adams	Nebraska

county	state
Adams	Pennsylvania
Adams	Illinois
Adams	Mississippi
Adams	Washington
Adams	Idaho
Adams	Wisconsin
Adams	Ohio
Adams	Iowa
Adams	North Dakota

Task1: Visualization

1.1 Pandemic Situation Visualization across States

```
> Visualization.used.data <- data.initial[, c("date", "county", "state",
+                                              "cases", "deaths", "total_population")]
```

calculate state population

```
> unique.county.population <- Visualization.used.data[!duplicated(Visualization.used.data,
+                                                                  by = c("county", "state"),
+                                                                  fromLast = T)]
> ## do not need parallel computing, "for-loop" is fast enough.
> for (i in selected.state.names) {
+   state.population <-
+     sum(unique.county.population[unique.county.population$state == i, total_population], na.rm = T)
+   Visualization.used.data$state.population[Visualization.used.data$state == i] <-
+     state.population
+ }
```

seperate monthly data (per 3 months)

```
> Jau <- as.Date("2020-01-01")
> Feb <- as.Date("2020-02-01")
> #Mar <- as.Date("2020-03-01")
> Apr <- as.Date("2020-04-01")
> #May <- as.Date("2020-05-01")
> #Jun <- as.Date("2020-06-01")
> Jul <- as.Date("2020-07-01")
> #Aug <- as.Date("2020-08-01")
> #Sept <- as.Date("2020-09-01")
> Oct <- as.Date("2020-10-01")
> #Nov <- as.Date("2020-11-01")
> #Dec <- as.Date("2020-12-01")
>
> first.season.data <- Visualization.used.data[which(Visualization.used.data$date >= Feb &
+                                                      Visualization.used.data$date < Apr),]
> second.season.data <- Visualization.used.data[which(Visualization.used.data$date >= Apr &
+                                                       Visualization.used.data$date < Jul),]
> third.season.data <- Visualization.used.data[which(Visualization.used.data$date >= Jul &
```

```
+                                                       Visualization.used.data$date < Oct),]
> fourth.season.data <- Visualization.used.data[which(Visualization.used.data$date >= Oct),]
```

Incidence Rate (No. per 100,000)

```
> ## a function that help calculate the incidence rate of each state
> Inci.rate.first.season <- function(dataset, state_name) {
+     temp.data <- dataset[dataset$state == state_name,]
+     last.day.data <- temp.data %>% group_by(county) %>% slice(which.max(date))
+     sum.cases <- sum(last.day.data$cases, na.rm = T)
+     Incidence.Rate <- (sum.cases/last.day.data$state.population[1]) * 100000
+     return(Incidence.Rate)
+ }
>
> ## use parallel computing
> cl.cores <- detectCores()
> cl <- makeCluster(cl.cores - 1)
> clusterEvalQ(cl, c(library(dplyr)))
> incidence.table <-
+   cbind(
+     parSapply(cl, c(selected.state.names), Inci.rate.first.season, dataset = first.season.data),
+     parSapply(cl, c(selected.state.names), Inci.rate.first.season, dataset = second.season.data),
+     parSapply(cl, c(selected.state.names), Inci.rate.first.season, dataset = third.season.data),
+     parSapply(cl, c(selected.state.names), Inci.rate.first.season, dataset = fourth.season.data)
+   )
> stopCluster(cl)
> colnames(incidence.table) <- c("2020/03/31", "2020/06/30", "2020/09/30", "2020/12/04")
```

```
> kable(head(incidence.table, n = 10), align = c("c", "c", "c", "c"),
+       caption = "Incidence rate (per 100,000 people)")
```

Table 2: Incidence rate (per 100,000 people)

	2020/03/31	2020/06/30	2020/09/30	2020/12/04
Washington	71.56646	488.3400	1294.4452	2583.801
Illinois	46.57755	1117.4255	2291.7075	6004.629
California	22.20457	600.5892	2124.2837	3410.079
Arizona	19.29085	1180.0266	3247.5069	5276.807
Massachusetts	93.21962	1610.6748	1920.0127	3511.747
Wisconsin	23.47606	507.1420	2244.5966	7510.550
Texas	13.31037	618.4163	2908.4966	4906.847
Nebraska	10.57802	1020.2742	2413.6496	7247.753
Utah	29.43943	758.5061	2476.8800	7090.560
Oregon	17.32681	218.3430	842.5854	2045.091

Mortality Rate (No. per 100,000 people)

```
> ## a function that help calculate the incidence rate of each state
> Mort.rate.first.season <- function(dataset, state_name) {
+     temp.data <- dataset[dataset$state == state_name,]
+     last.day.data <- temp.data %>% group_by(county) %>% slice(which.max(date))
+     sum.deaths <- sum(last.day.data$deaths, na.rm = T)
```

```
+     Mortality.Rate <- (sum.deaths/last.day.data$state.population[1]) * 100000
+     return(Mortality.Rate)
+ }
>
> ## use parallel computing
> cl.cores <- detectCores()
> cl <- makeCluster(cl.cores - 1)
> clusterEvalQ(cl, c(library(dplyr)))
> mortality.table <-
+   cbind(
+     parSapply(cl, c(selected.state.names), Mort.rate.first.season, dataset = first.season.data),
+     parSapply(cl, c(selected.state.names), Mort.rate.first.season, dataset = second.season.data),
+     parSapply(cl, c(selected.state.names), Mort.rate.first.season, dataset = third.season.data),
+     parSapply(cl, c(selected.state.names), Mort.rate.first.season, dataset = fourth.season.data)
+   )
> stopCluster(cl)
> colnames(mortality.table) <-
+   c("2020/03/31", "2020/06/30", "2020/09/30", "2020/12/04")
```

```
> kable(head(mortality.table, n = 10), align = c("c", "c", "c", "c"),
+       caption = "Mortality rate (per 100,000 people)")
```

Table 3: Mortality rate (per 100,000 people)

	2020/03/31	2020/06/30	2020/09/30	2020/12/04
Washington	3.1951836	18.845928	31.38632	43.26222
Illinois	0.8325757	53.961800	67.65650	107.44117
California	0.4760155	15.736968	41.12877	51.20012
Arizona	0.2526537	24.447963	84.02965	102.32476
Massachusetts	1.3200551	119.353743	140.17798	161.72899
Wisconsin	0.4344201	13.658168	23.07640	66.84857
Texas	0.2114523	9.207449	59.73342	84.91479
Nebraska	0.2126236	14.830494	26.20585	63.73391
Utah	0.0678328	5.460539	15.56762	31.37266
Oregon	0.4520038	5.273378	14.06234	25.21177

Fatality Rate (per 1,000 patients)

```
> Fatality.table <- round(mortality.table/incidence.table * 1000, digits = 2)
> kable(head(Fatality.table, n = 10), align = c("c", "c", "c", "c"),
+       caption = "Fatality rate (per 1,000 patients)")
```

Table 4: Fatality rate (per 1,000 patients)

	2020/03/31	2020/06/30	2020/09/30	2020/12/04
Washington	44.65	38.59	24.25	16.74
Illinois	17.88	48.29	29.52	17.89
California	21.44	26.20	19.36	15.01
Arizona	13.10	20.72	25.88	19.39
Massachusetts	14.16	74.10	73.01	46.05
Wisconsin	18.50	26.93	10.28	8.90
Texas	15.89	14.89	20.54	17.31

	2020/03/31	2020/06/30	2020/09/30	2020/12/04
Nebraska	20.10	14.54	10.86	8.79
Utah	2.30	7.20	6.29	4.42
Oregon	26.09	24.15	16.69	12.33

Plot

1.2 Line Chart of Incidence Rate and Mortality Rate across Country

define two parities

- based on the results of presidential election 2020

```
> blue.state.names <-
+   c(
+     "Washington",
+     "Illinois",
+     "California",
+     "Arizona",
+     "Massachusetts",
+     "Wisconsin",
+     "Oregon",
+     "New York",
+     "Georgia",
+     "New Hampshire",
+     "New Jersey",
+     "Colorado",
+     "Maryland",
+     "Nevada",
+     "Minnesota",
+     "Pennsylvania",
+     "District of Columbia",
+     "Vermont",
+     "Virginia",
+     "Connecticut",
+     "Michigan",
+     "Delaware",
+     "New Mexico",
+     "Maine",
+     "Rhode Island"
+   )
>
> red.state.names <-
+   c(
+     "Texas",
+     "Nebraska",
+     "Utah",
+     "Florida",
+     "North Carolina",
+     "Tennessee",
+     "Indiana",
+     "Kentucky",
+     "Oklahoma",
+     "South Carolina",
```

```
+     "Kansas",
+     "Missouri",
+     "Iowa",
+     "Louisiana",
+     "Ohio",
+     "South Dakota",
+     "Arkansas",
+     "Mississippi",
+     "North Dakota",
+     "Wyoming",
+     "Alabama",
+     "Idaho",
+     "Montana",
+     "West Virginia"
+   )
```

calculate blue and red state total population

```
> calculate.party.population <- unique(Visualization.used.data %>% select(c("state", "state.population"
> blue.state.population <- sum(calculate.party.population[which(calculate.party.population$state %in%
+                                                                  blue.state.names), "state.population"
> red.state.population <- sum(calculate.party.population[which(calculate.party.population$state %in%
+                                                                 red.state.names), "state.population"])
```

data for line chart

```
> line.chart.data <- Visualization.used.data %>% select(c("date", "state", "cases", "deaths"))
> line.chart.data$party[line.chart.data$state %in% blue.state.names] <- "Democratic"
> line.chart.data$party[line.chart.data$state %in% red.state.names] <- "Republican"
> line.chart.used.data <- line.chart.data %>% select(-c("state"))
>
> unique.date <- unique(line.chart.used.data$date)
```

```
> accmulate.cases <- function(which.date) {
+   temp.data <- line.chart.used.data[line.chart.used.data$date == which.date]
+   Republican.inci.rate <- round(sum(temp.data[which(temp.data$party == "Republican"), "cases"])/
+                                   red.state.population * 100000, digits = 3)
+   Democratic.inci.rate <- round(sum(temp.data[which(temp.data$party == "Democratic"), "cases"])/
+                                   blue.state.population * 100000, digits = 3)
+   rate <- as.vector(c(as.numeric(which.date), Democratic.inci.rate, Republican.inci.rate))
+   return(rate)
+ }
>
> inci.matrix <- matrix(rep(NA, 3 * length(unique.date)), ncol = 3)
> j <- 0
> for (i in unique.date) {
+   j <- j + 1
+   inci.matrix[j, ] <- accmulate.cases(i)
+ }
> inci.dataframe <- as.data.frame(inci.matrix)
> colnames(inci.dataframe) <- c("date", "Democratic", "Republican")
```

```
> inci.dataframe$date <- as.Date(inci.dataframe$date, origin="1970-01-01")
```

```
> kable(tail(inci.dataframe, n = 10), align = c("c", "c", "c"),
+       caption = "Incidence rate on different date (per 100,000 people)")
```

Incidence Rate

Table 5: Incidence rate on different date (per 100,000 people)

	date	Democratic	Republican
310	2020-11-25	3524.744	4630.604
311	2020-11-26	3562.012	4654.750
312	2020-11-27	3618.197	4726.999
313	2020-11-28	3665.235	4771.615
314	2020-11-29	3705.673	4818.318
315	2020-11-30	3752.682	4876.996
316	2020-12-01	3808.313	4936.354
317	2020-12-02	3867.997	5006.161
318	2020-12-03	3933.545	5075.919
319	2020-12-04	4005.795	5149.713

```
> accmulate.deaths <- function(which.date) {
+   temp.data <- line.chart.used.data[line.chart.used.data$date == which.date]
+   Republican.inci.rate <- round(sum(temp.data[which(temp.data$party == "Republican"), "deaths"])/
+                                 red.state.population * 100000, digits = 3)
+   Democratic.inci.rate <- round(sum(temp.data[which(temp.data$party == "Democratic"), "deaths"])/
+                                 blue.state.population * 100000, digits = 3)
+   rate <- as.vector(c(as.numeric(which.date), Democratic.inci.rate, Republican.inci.rate))
+   return(rate)
+ }
>
> death.matrix <- matrix(rep(NA, 3 * length(unique.date)), ncol = 3)
> j <- 0
> for (i in unique.date) {
+   j <- j + 1
+   death.matrix[j, ] <- accmulate.deaths(i)
+ }
> death.dataframe <- as.data.frame(death.matrix)
> colnames(death.dataframe) <- c("date", "Democratic", "Republican")
> death.dataframe$date <- as.Date(death.dataframe$date, origin="1970-01-01")
```

```
> kable(tail(death.dataframe, n = 10), align = c("c", "c", "c"),
+       caption = "Mortality rate on different date (per 100,000 people)")
```

Mortality Rate

Table 6: Mortality rate on different date (per 100,000 people)

	date	Democratic	Republican
310	2020-11-25	88.398	73.321
311	2020-11-26	88.783	73.664
312	2020-11-27	89.217	74.088
313	2020-11-28	89.602	74.441
314	2020-11-29	89.840	74.714
315	2020-11-30	90.210	75.127
316	2020-12-01	90.969	76.042
317	2020-12-02	91.805	77.023
318	2020-12-03	92.693	77.930
319	2020-12-04	93.393	78.861

```
> Fatality.matrix <- matrix(rep(NA, 2 * length(unique.date)), ncol = 2)
> Fatality.matrix[, 1] <- death.dataframe$Democratic/inci.dataframe$Democratic * 1000
> Fatality.matrix[, 2] <- death.dataframe$Republican/inci.dataframe$Republican * 1000
> Fatality.dataframe <- cbind(death.dataframe$date, as.data.frame(Fatality.matrix))
> colnames(Fatality.dataframe) <- c("date", "Democratic", "Republican")
```

```
> kable(tail(Fatality.dataframe, n = 10), align = c("c", "c", "c"),
+       caption = "Fatality rate on different date (per 1,000 patients)")
```

Fatality Rate

Table 7: Fatality rate on different date (per 1,000 patients)

	date	Democratic	Republican
310	2020-11-25	25.07927	15.83400
311	2020-11-26	24.92496	15.82555
312	2020-11-27	24.65786	15.67337
313	2020-11-28	24.44645	15.60080
314	2020-11-29	24.24391	15.50624
315	2020-11-30	24.03881	15.40436
316	2020-12-01	23.88695	15.40449
317	2020-12-02	23.73451	15.38564
318	2020-12-03	23.56475	15.35288
319	2020-12-04	23.31447	15.31367

Task2: Prediction Model

2.0 Clean the dataset

- Only use the last day data; (It may or may not be the data on 2020/12/04, because some counties will stop collecting data early.)

```
> Gener.pred.model.data <- function(state_name) {
+     temp.data <- data.initial[data.initial$state == state_name,]
+     county.last.day.data <- temp.data %>% group_by(county) %>% slice(which.max(date))
+     return(county.last.day.data)
```

```
+ }
>
> ## parallel computing
> cl <- makeCluster(cl.cores - 1)
> registerDoParallel(cl)
> clusterEvalQ(cl, c(library(dplyr)))
> prediction.data <- foreach (i = selected.state.names, .combine = "rbind") %dopar% {
+   Gener.pred.model.data(i)
+ }
> stopCluster(cl)
> prediction.data$incidence.rate <-
+   prediction.data$cases / prediction.data$total_population * 100000
> prediction.data$mortality.rate <-
+   prediction.data$deaths / prediction.data$total_population * 100000
> prediction.data$fatality.rate <-
+   prediction.data$mortality.rate / prediction.data$incidence.rate * 1000
>
> Prediction.model.data <- prediction.data %>% select(-c("total_population", "cases", "deaths"))
```

2.1 Logistic Regression

2.2 Random Forest