Midterm MA615 2023

Mingrui Du

2023-11-06

```
#install.packages("rfema", repos = "https://ropensci.r-universe.dev")
library(httr)
library(jsonlite)
library(rfema)
suppressMessages(library(dplyr))
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ forcats   1.0.0     ✓ readr     2.1.4
## ✓ ggplot2   3.4.3     ✓ stringr   1.5.0
## ✓ lubridate 1.9.3     ✓ tibble    3.2.1
## ✓ purrr     1.0.2     ✓ tidyr     1.3.0
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter()  masks stats::filter()
## ✖ purrr::flatten() masks jsonlite::flatten()
## ✖ dplyr::lag()     masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to be
come errors
```

```
library(stringr)
suppressMessages(library(maps))
```

```
## Warning: 程辑包'maps'是用R版本4.3.2 来建造的
```

```
suppressMessages(library(plotly))
```

```
## Warning: 程辑包'plotly'是用R版本4.3.2 来建造的
```

```
suppressMessages(library(usmap))
```

```
## Warning: 程辑包'usmap'是用R版本4.3.2 来建造的
```

Data acquisition and assessment

Flood Data

Two data sets obtained from Federal Emergency Management Agency (FEMA) (https://www.fema.gov/about/openfema/data-sets), v1 containing financial assistance and v2 for all declared disasters.

v1 (Fema Web Disaster Summaries) (https://www.fema.gov/openfema-data-page/fema-web-disaster-summaries-v1) is a 14-column, formatted data frame containing: sequential disaster number; total approved number of Individual Assistance (IA); approved amount($) for the Individual and Households Program (IHP), which includes Housing Assistance (HA) and Other Needs Assistance (ONA); amount of Public Assistance grant funding (PA) and Hazard Mitigation Grant Program (HMGP) (https://www.fema.gov/grants/mitigation/hazard-mitigation).

v2 (Disaster Declarations Summaries) (https://www.fema.gov/openfema-data-page/disaster-declarations-summaries-v2) contains disaster time, counties, incident type, disaster descriptions and assistance programs. Among these, column “declarationTitle” is a concatenation of all incidents included in one declared disaster.

The two data sets are connected by disaster number. Since the assignment focuses on year 2020-2021, the two data set will be filtered respectively for further investigation.

CENSUS Data

The three files of CENSUS data posted on bb are originally obtained from U.S. Census Bureau (https://www.census.gov/data/developers/data-sets/acs-5year.html), while S1701 (https://www.census.gov/acs/www/data/data-tables-and-tools/subject-tables/) stands for poverty status in the past 12 Months, DP05 (https://data.census.gov/table/ACSDP5Y2021.DP05) for demographic and housing estimates, B25001 (https://data.census.gov/table?q=B25001) for housing units.

Storm Data

Obtained from NOAA (https://www.ncei.noaa.gov/pub/data/swdi/stormevents/csvfiles/), the three data sets in each year record locations, details and fatalities. “Locations” records magnitude and coordinates of the incident. “Details” contains begin/end time, duration, event type, location, damage and narratives. “Fatalities” sadly includes age and gender of the departed, as well as their incident location.

The three data sets could be matched by EVENT_ID, an ID assigned by NWS for each individual storm event contained within a storm episode; one storm episode may contain several events. However, I haven’t figure out how to link fatalities to the other two table with uncertain, one-to-one or many-to-one connection, thus quit this part.

Initial Questions

- Locate flooding area according to FEMA data sets.
- How many household are affected by floods in each region?
- Where are the highly risky locations when incidents happen?

Data Cleaning and Organization

Data Loading

A total of 8 data sets are read via local files, or API. API for OpenFEMA data set (https://www.fema.gov/about/openfema/api). Disaster summery data (fema_v2) are filtered by incident begin date from 1/1/2020 to before 1/1/2022, incident type limited in “Flood”. CENSUS data are loaded from local csv files, single-value columns removed at very beginning to simplify observation.

```
## via API
base_fema <- 'https://www.fema.gov/api/open/'
version_2 <- 'v2/'
entity <- 'DisasterDeclarationsSummaries'
info_key <- "?$filter=incidentBeginDate%20ge%20'2020-01-01'%20and%20incidentBeginDate%20lt%20
'2022-01-01'%20and%20incidentType%20eq%20'Flood'"

url_fema <- paste0(base_fema, version_2, entity, info_key)
rawd <- GET(url_fema)
raw_text <- content(rawd, as = "text", encoding = "UTF-8")
df <- fromJSON(raw_text, flatten = TRUE)
fema_v2 <- data.frame(df$DisasterDeclarationsSummaries)

## via local files
fema_v1 <- read.csv("FemaWebDisasterSummaries.csv", header = T)
## S1701: Poverty
pov_20 <- read.csv("ACSST5Y2020.S1701-Data.csv", header = T)
pov_20 <- pov_20 %>% select_if(~ length(unique(.)) > 2)
pov_21 <- read.csv("ACSST5Y2021.S1701-Data.csv", header = T)
pov_21 <- pov_21 %>% select_if(~ length(unique(.)) > 2)
## all.equal(colnames(pov_20), colnames(pov_21))

## DP05: ACS DEMOGRAPHIC AND HOUSING ESTIMATES
demo_20 <- read.csv("ACSDP5Y2020.DP05-Data.csv", header = T)
demo_20 <- demo_20 %>% select_if(~ length(unique(.)) > 2)
demo_21 <- read.csv("ACSDP5Y2021.DP05-Data.csv", header = T)
demo_21 <- demo_21 %>% select_if(~ length(unique(.)) > 2)

## B25001: Housing Units
haus_uni_20 <- read.csv("ACSDT5Y2020.B25001-Data.csv", header = T)
haus_uni_20 <- haus_uni_20 %>% select_if(~ length(unique(.)) > 2)
colnames(haus_uni_20) <- unlist(haus_uni_20[1, ])
haus_uni_20 <- haus_uni_20[-1,]

haus_uni_21 <- read.csv("ACSDT5Y2021.B25001-Data.csv", header = T)
haus_uni_21 <- haus_uni_21 %>% select_if(~ length(unique(.)) > 2)
colnames(haus_uni_21) <- unlist(haus_uni_21[1, ])
haus_uni_21 <- haus_uni_21[-1,]
```

Data Cleaning and Organization

FEMA v1 records fiscal information long long ago. Since only year 2020 and 2021 are taken into consideration, this data set is filtered by disaster number in FEMA v2. Unused column removed.

```
## v1
fema_v1 <- fema_v1 %>%
  select(-hash, -lastRefresh, -id)
## select data by disaster number in v2:
fema_v1 <- fema_v1 |>
  semi_join(fema_v2, by = "disasterNumber")
```

FEMA v2: remove unused columns, split BeginDate, Area, DeclarationTitle then separate by year

```
## v2
fema_v2 <- fema_v2 %>%
  select(-hash, -lastRefresh, -id, -femaDeclarationString) %>%
  select(iaProgramDeclared, where(~length(unique(.)) > 1))
## 'iaProgramDeclared' saved as a funding program indicator

## Split concatenation columns:
### incidentBeginDate into Year, Month, Date
fema_v2$incidentBeginDate <- as.character(fema_v2$incidentBeginDate)
fema_v2$incidentBeginDate <- gsub("T00:00:00.000Z", "", fema_v2$incidentBeginDate)
fema_v2 <- fema_v2 |>
                separate_wider_delim(cols = incidentBeginDate,
                                     delim = "-",
                                     names = c("Year",
                                               "Month",
                                               "Date"))

### declarationTitle
fema_v2$declarationTitle <- gsub(", AND", ",", fema_v2$declarationTitle)
fema_v2$declarationTitle <- gsub("AND ", ", ", fema_v2$declarationTitle)
fema_v2$declarationTitle <- gsub("SEVERE, STORMS", "SEVERE STORMS", fema_v2$declarationTitle)
### unique(grepl(" AND ", fema_v2$declarationTitle))
fema_v2 <- fema_v2 |>
                separate_wider_delim(cols = declarationTitle,
                                     delim = ",",
                                     names = c("temp1",
                                               "temp2",
                                               "temp3",
                                               "temp4",
                                               "temp5"),
                                     too_many = "error",
                                     too_few = "align_start")
fema_v2 <- fema_v2 %>% mutate_at(vars(temp1:temp4), ~trimws(.))

## create FIPS code column
fema_v2$fips <- paste0(fema_v2$fipsStateCode, fema_v2$fipsCountyCode, sep = "")
fema_v2$fips <- ifelse(grepl("000", fema_v2$fips), NA, fema_v2$fips)

## Unify designatedArea
## Delete "(County)", and "()" for territories
fema_v2$designatedArea <- as.character(fema_v2$designatedArea)
fema_v2$designatedArea <- fema_v2$designatedArea |>
  str_replace_all(" \\(County\\)", "")
fema_v2$designatedArea <- gsub("[()]", "", fema_v2$designatedArea)

## Separate FEMA v2 by year 2020 and 2021
fema_v2_20 <- fema_v2 %>% filter(Year == '2020')
fema_v2_21 <- fema_v2 %>% filter(Year == '2021')
```

CENSUS: Split poverty data set by Estimates and Percent

```
## census data
## Poverty
Area <- as.data.frame(pov_20$NAME)|>
    separate_wider_delim(cols = `pov_20$NAME`,
                         delim = ",",
                         names = c("County","State"),
                         too_few = "align_start",
                         too_many = "error")
Area[1,1] <- "County"
Area[1,2] <- "State"
Area$County <- trimws(Area$County)
Area$State <- trimws(Area$State)
## Divide Poverty Data
## Remove all non-estimate columns
pov_20 <- pov_20[, grepl("E$",colnames(pov_20))]
## Total
pov_20_totl <- data.frame(Area, pov_20[ ,grepl("C01", colnames(pov_20))])
## Below Poverty Level
pov_20_blo <- data.frame(Area, pov_20[ ,grepl("C02", colnames(pov_20))])
## Percent of Blo
pov_20_blopc <- data.frame(Area, pov_20[ ,grepl("C03", colnames(pov_20))])

## Replace col names by first row
colnames(pov_20_totl) <- unlist(pov_20_totl[1, ])
pov_20_totl <- pov_20_totl[-1,]

## age
colnames(pov_20_totl) <-
  sapply(colnames(pov_20_totl),
         function(x) gsub("Population for whom poverty status is determined!!AGE!!",
                                                "age_", x))
colnames(pov_20_totl) <-
  sapply(colnames(pov_20_totl),
         function(x) gsub("Under 18 years!!", "", x))
colnames(pov_20_totl) <-
  sapply(colnames(pov_20_totl),
         function(x) gsub("18 to 64 years!!", "", x))

## Education
colnames(pov_20_totl) <-
  sapply(colnames(pov_20_totl),
         function(x) gsub("Population for whom poverty status is determined!!EDUCATIONAL ATTA
INMENT!!",
                          "edu_", x))
colnames(pov_20_totl) <-
  sapply(colnames(pov_20_totl),
         function(x) gsub("Population 25 years and over!!", "", x))

## Employment
colnames(pov_20_totl) <-
  sapply(colnames(pov_20_totl),
```

```
        function(x) gsub("Population for whom poverty status is determined!!EMPLOYMENT STATU
S!!",
                         "emp_", x))
colnames(pov_20_totl) <-
  sapply(colnames(pov_20_totl),
        function(x) gsub("Civilian labor force 16 years and over!!", "", x))
```

Split Area into County and State

```
## CENSUS
cty_state <- function(c){
  c <- c |> separate_wider_delim(cols = `Geographic Area Name`,
                                 delim = ",",
                                 names = c("County",
                                           "State"))
  c$County <- trimws(c$County)
  c$State <- trimws(c$State)
  return(c)
}
#pov_20 <- cty_state(pov_20); pov_21 <- cty_state(pov_21)
#demo_20 <- cty_state(demo_20); demo_21 <- cty_state(demo_21)
haus_uni_20 <- cty_state(haus_uni_20); haus_uni_21 <- cty_state(haus_uni_21)
haus_uni_20$County <- gsub(" County", "", haus_uni_20$County)
haus_uni_21$County <- gsub(" County", "", haus_uni_21$County)
```

EDA

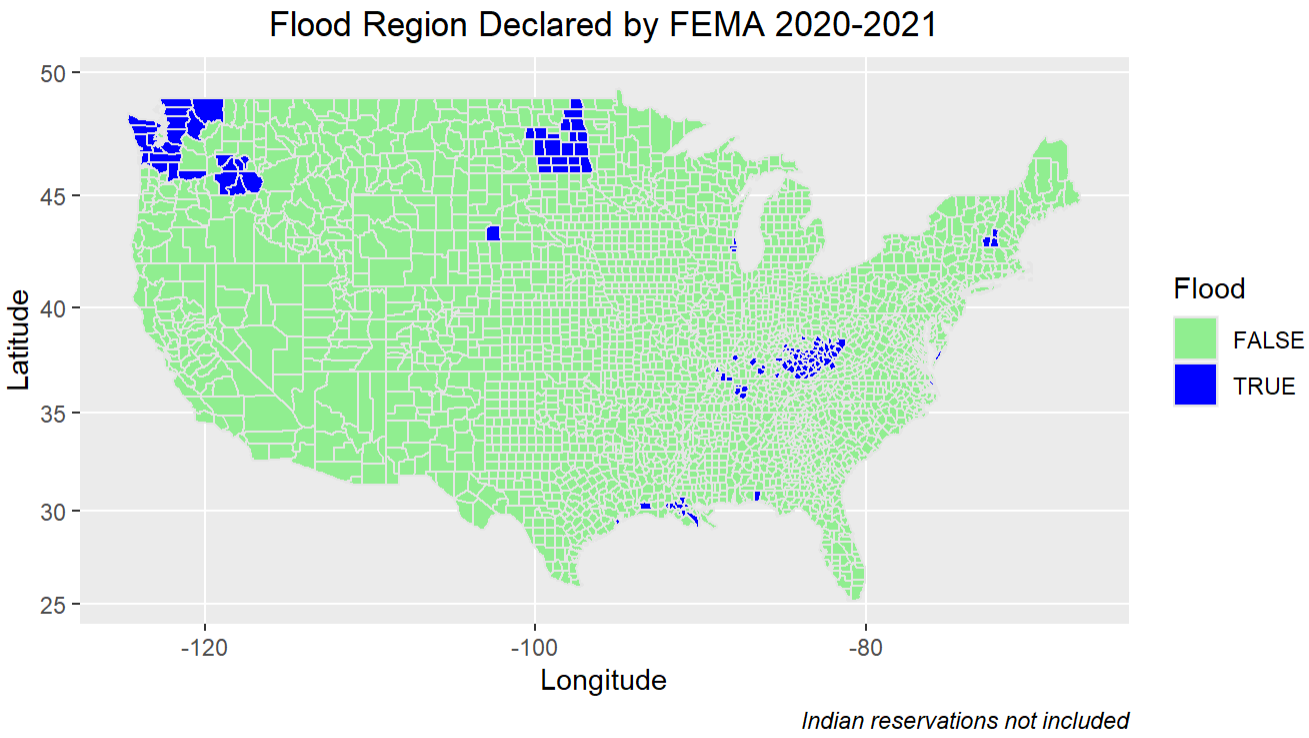
```
## 1 FEMA reported flooding region
dfips <- maps::county.fips %>%
  as.tibble %>%
  extract(polyname, c("region", "subregion"), "^([^,]+),([^,]+)$")
```

```
## Warning: `as.tibble()` was deprecated in tibble 2.0.0.
## ℹ Please use `as_tibble()` instead.
## ℹ The signature and semantics have changed, see `?as_tibble`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
dall <- map_data("county") %>% left_join(dfips)
```

```
## Joining with `by = join_by(region, subregion)`
```

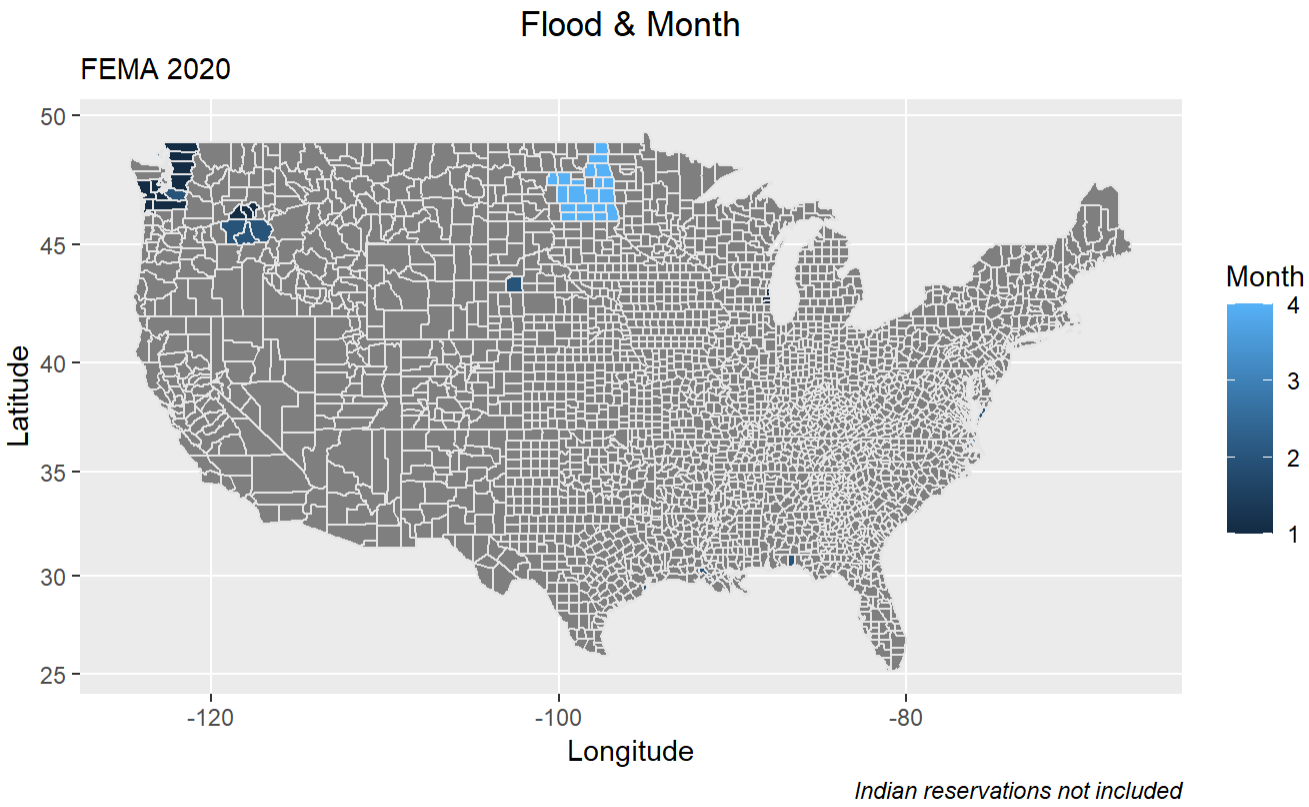
```
dall_2 <- dall
dall_3 <- dall
plot1 <- dall %>%
  mutate(fld_place = fips %in% fema_v2$fips) %>%
  ggplot(aes(long, lat, group = group)) +
  geom_polygon(aes(fill = fld_place), color="grey90") +
  labs(x = "Longitude",
       y = "Latitude",
       fill = "Flood",
       title = "Flood Region Declared by FEMA 2020-2021",
       caption = "Indian reservations not included") +
  scale_fill_manual(values = c("TRUE" = "blue", "FALSE" = "lightgreen")) +
  coord_map()+
  theme(plot.title = element_text(hjust = 0.5),
        plot.caption = element_text(face = "italic"))
plot1
```



The blue-colored locations are determinded by FIPS code. Indian reservation are not included because their places have no FIPS code. I have no idea how to add them in except manually introducing latitude and longitude info.

Another question aroused is, since flooding areas locate at different parts on the map, will there be any seasonal patterns?

```
## Season & Flood
fema_v2_20$Month <- as.numeric(fema_v2_20$Month)
fema_v2_20$fips <- as.numeric(fema_v2_20$fips)
dall <- dall %>%
  left_join(fema_v2_20, by = "fips")
plot2 <- dall %>%
  mutate(fld_place = fips %in% fema_v2_20$fips) %>%
  ggplot(aes(long, lat, group = group, color = Month)) +
  geom_polygon(aes(fill=Month), color="grey90") +
  labs(x = "Longitude",
       y = "Latitude",
       fill = "Month",
       title = "Flood & Month",
       subtitle = "FEMA 2020",
       caption = "Indian reservations not included") +
  scale_color_viridis_c(option = "inferno", direction = 1) +
  coord_map()+
  theme(plot.title = element_text(hjust = 0.5),
        plot.caption = element_text(face = "italic"))
plot2
```



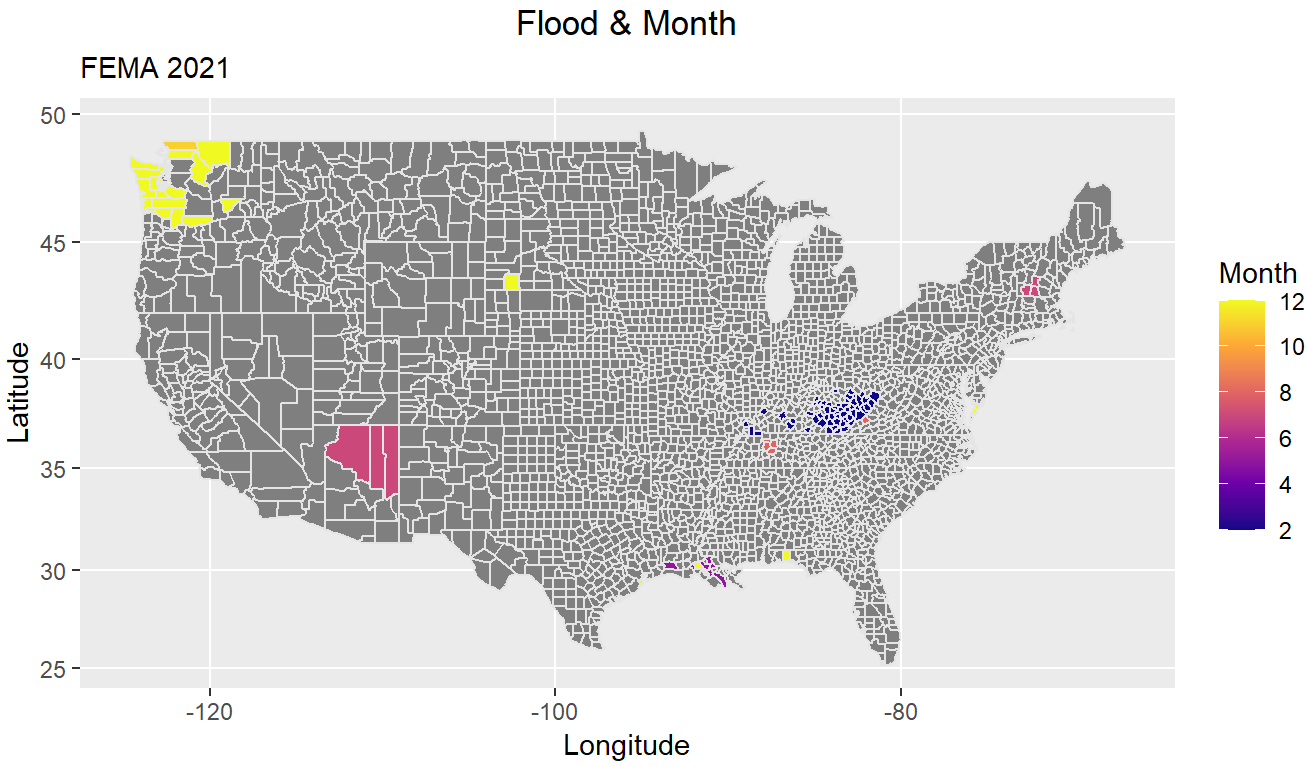
As shown above, in 2020, FEMA declared flooding occurred from Jan. to Apr., colors telling the month.

And another one for Year 2021.

```
fema_v2_21$Month <- as.numeric(fema_v2_21$Month)
fema_v2_21$fips <- as.numeric(fema_v2_21$fips)
dall_2 <- dall_2 %>%
  left_join(fema_v2_21, by = "fips")
```

```
## Warning in left_join(., fema_v2_21, by = "fips"): Detected an unexpected many-to-many rela
tionship between `x` and `y`.
## ℹ Row 13528 of `x` matches multiple rows in `y`.
## ℹ Row 42 of `y` matches multiple rows in `x`.
## ℹ If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
plot3 <- dall_2 %>%
  mutate(fld_place = fips %in% fema_v2_21$fips) %>%
  ggplot(aes(long, lat, group = group, color = Month)) +
  geom_polygon(aes(fill=Month), color="grey90") +
  labs(x = "Longitude",
       y = "Latitude",
       fill = "Month",
       title = "Flood & Month",
       subtitle = "FEMA 2021",
       caption = "Indian reservations not included") +
  scale_fill_viridis_c(option = "C") +
  coord_map()+
  theme(plot.title = element_text(hjust = 0.5),
        plot.caption = element_text(face = "italic"))
plot3
```



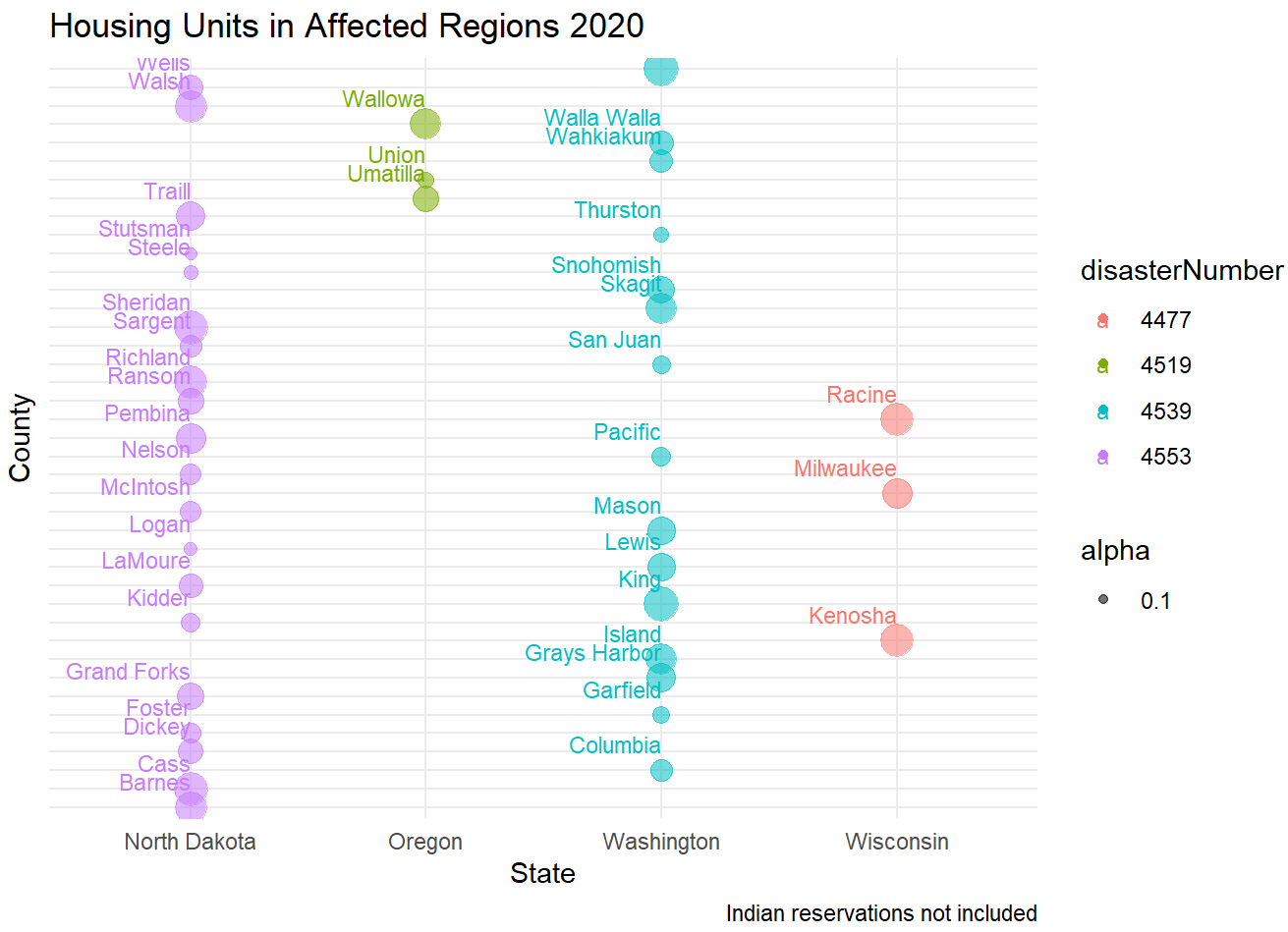
Indian reservations not included

```
## 3 Housing Units
## add Housing Units info to fema_v2, matched by designatedArea and County
## FEMA v2 contains counties with reused names in several states.
## Thus create dictionary between state abbreviation and full name.
state_dict <- list(
  AL = "Alabama",
  AK = "Alaska",
  AZ = "Arizona",
  AR = "Arkansas",
  CA = "California",
  CO = "Colorado",
  CT = "Connecticut",
  DE = "Delaware",
  FL = "Florida",
  GA = "Georgia",
  HI = "Hawaii",
  ID = "Idaho",
  IL = "Illinois",
  IN = "Indiana",
  IA = "Iowa",
  KS = "Kansas",
  KY = "Kentucky",
  LA = "Louisiana",
  ME = "Maine",
  MD = "Maryland",
  MA = "Massachusetts",
  MI = "Michigan",
  MN = "Minnesota",
  MS = "Mississippi",
  MO = "Missouri",
  MT = "Montana",
  NE = "Nebraska",
  NV = "Nevada",
  NH = "New Hampshire",
  NJ = "New Jersey",
  NM = "New Mexico",
  NY = "New York",
  NC = "North Carolina",
  ND = "North Dakota",
  OH = "Ohio",
  OK = "Oklahoma",
  OR = "Oregon",
  PA = "Pennsylvania",
  RI = "Rhode Island",
  SC = "South Carolina",
  SD = "South Dakota",
  TN = "Tennessee",
  TX = "Texas",
  UT = "Utah",
  VT = "Vermont",
  VA = "Virginia",
```

```
  WA = "Washington",
  WV = "West Virginia",
  WI = "Wisconsin",
  WY = "Wyoming"
)
fema_v2_20$StFull <- as.character(state_dict[fema_v2_20$state])
fema_v2_20 <- fema_v2_20 %>%
  inner_join(haus_uni_20, by = c("designatedArea"= "County", "StFull" = "State"))
fema_v2_21$StFull <- as.character(state_dict[fema_v2_21$state])
fema_v2_21 <- fema_v2_21 %>%
  inner_join(haus_uni_21, by = c("designatedArea"= "County", "StFull" = "State"))
fema_v2_20$disasterNumber <- as.character(fema_v2_20$disasterNumber)

plot4 <- ggplot(fema_v2_20) +
  aes(x = StFull,
      y = designatedArea,
      size = `Estimate!!Total`,
      color = disasterNumber) +
  geom_point(aes(alpha = 0.1),
             shape = "circle", fill = "white") +
  geom_text(aes(label = designatedArea), size = 3, vjust = -1, hjust = 1) +
  labs(x = "State",
       y = "County",
       title = "Housing Units in Affected Regions 2020",
       caption = "Indian reservations not included") +
  scale_y_discrete(labels = NULL) +
  theme(plot.title = element_text(hjust = 0.5),
        plot.caption = element_text(face = "italic")) +
  theme_minimal() +
  guides(fill = FALSE, size = FALSE)

plot4
```



This plot visualizes estimates housing units at each affected area in 2020.

References

R FEMA (https://docs.ropensci.org/rfema/#installation)

NOAA Storm Events Database (https://www.ncdc.noaa.gov/stormevents/ftp.jsp)

OpenFEMA Data Sets (https://www.fema.gov/about/openfema/data-sets)

API in R (https://statisticsglobe.com/api-in-r)

Geographic Codes Explained (https://www.census.gov/library/reference/code-lists/ansi.html)

Federal American Indian Reservations (https://hub.arcgis.com/datasets/41a17452810f4b6f819924f8638c520f_0/about)