MA615_Midterm

Bingtian Ye

2023-10-27

1.read data

v1 (OpenFEMA Dataset: FEMA Web Disaster Summaries - v1, link = https://www.fema.gov/openfema-data-page/fema-web-disaster-summaries-v1) and v2 (OpenFEMA Dataset: Disaster Declarations Summaries - v2, link = https://www.fema.gov/openfema-data-page/disaster-declarations-summaries-v2). I observed some data provided by FEMA, and some of the data has nothing to do with flooding. For example, the content in WebDisasterDeclarations that is related to flooding is included in v2. So only the above two data are used.

```r
v1=read.csv("v1.csv",header=T)
v2=read.csv("v2.csv",header=T)
```

1.Data Fields

v1 FEMA Web Disaster Summaries

Name	Description
disasterNumber	Unique identifier for each disaster. Can be used to merge datasets
totalNumberIaApproved	Approved applications for Individual Assistance
totalAmountIhpApproved	Dollars approved for Individual and Households Program
totalAmountHaApproved	Dollars approved for Housing Assistance
totalAmountOnaApproved	Dollars approved for Other Needs Assistance
totalObligatedAmountPa	Dollars available for Public Assistance grants
totalObligatedAmountCatAb	Dollars for Emergency Work Public Assistance (Categories A & B)
totalObligatedAmountCatC2g	Dollars for Permanent Work Public Assistance (Categories C to G)
paLoadDate	Date Public Assistance data was updated
iaLoadDate	Date Individual Assistance data was updated
totalObligatedAmountHmgp	Dollars obligated for the Hazard Mitigation Grant Program
hash	MD5 hash for the record's integrity
lastRefresh	Date the record was last refreshed
id	Unique ID for the record

v2 Disaster Declarations Summaries

Name	Description
femaDeclarationString	Agency standard method for uniquely identifying Stafford Act declarations.
disasterNumber	The number of the disaster, unique. Can be used to merge datasets
state	The name or phrase describing the U.S. state, district, or territory
declarationType	Two character code that defines the disaster declaration type.
declarationDate	Date the disaster was declared

Name	Description
fyDeclared	Fiscal year in which the disaster was declared
incidentType	Type of incident such as fire or flood.
declarationTitle	Title for the disaster
ihProgramDeclared	Whether the Individuals and Households program was declared for this disaster.
iaProgramDeclared	Whether the Individual Assistance program was declared for this disaster.
paProgramDeclared	Whether the Public Assistance program was declared for this disaster.
hmProgramDeclared	Whether the Hazard Mitigation program was declared for this disaster.
incidentBeginDate	Date the incident itself began
incidentEndDate	Date the incident itself ended
disasterCloseoutDate	Date all financial transactions for all programs are completed
tribalRequest	Whether a declaration request was submitted by a Tribal Nation.
fipsStateCode	FIPS code used to identify US states and territories
fipsCountyCode	FIPS code used to identify US counties and equivalents
placeCode	FEMA's internal code system to recognize locations
designatedArea	Geographic area included in the declaration
declarationRequestNumber	Number assigned to the declaration request
lastIAFilingDate	Last date when IA requests can be filed.
lastRefresh	Date the record was last updated in the API data store
hash	MD5 Hash of the fields and values of the record
id	Unique ID assigned to the record

Data preparing

By observing the contents of the data table, I decided to use v2 as the main database, and then merged the two databases based on the disasterNumber field. ### Data preparing for v2 First select the required columns. Based on observation, the selected columns are as follows: disasterNumber, state, declarationType, declarationDate, incidentType, declarationTitle, ihProgramDeclared, iaProgramDeclared, paProgramDeclared, hmProgramDeclared, fipsStateCode, fipsCountyCode, id

```
v2 <- v2 |>
  select(disasterNumber, state, declarationType, declarationDate, incidentType, declarationTitle, ihProg
```

Split column declarationDate and select the years in 2020 and 2021

```
v2 <- v2 |>
  mutate(year = substr(declarationDate, 1, 4),
         month = substr(declarationDate, 6, 7),
         day = substr(declarationDate, 9, 10)) |>
  filter(year == "2020" | year == "2021") |>
  select(!declarationDate)
```

Filter the rows where incidentType is flood, and replace the abbreviation of declarationType with the completion name (DR = Major Disaster Declaration, EM = Emergency Declaration, FM = Fire Management Assistance Declaration).

```
v2 <- v2 |>
  filter(incidentType == "Flood") |>
  mutate(declarationType = case_when(
    declarationType == "DR" ~ "Major Disaster Declaration",
    declarationType == "EM" ~ "Emergency Declaration",
    declarationType == "FM" ~ "Fire Management Assistance Declaration"
  ))
```

Remove columns whose values are the same

```
v2 <- v2 |>
  select_if(~ n_distinct(.) > 1)
```

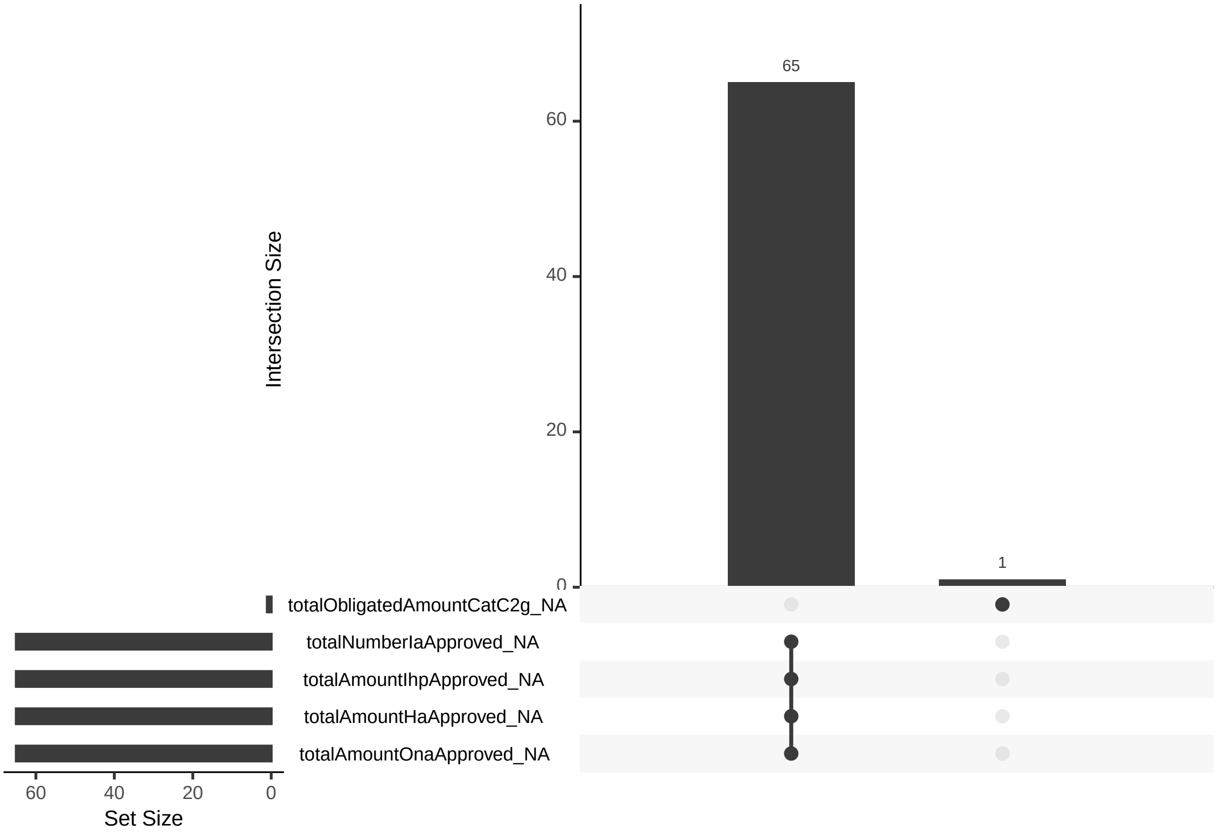
Data preparing for v1 and Merge two dataset

Remove the following columns: paLoadDate, iaLoadDate, hash, lastRefresh, id. And then merge to dataset by using key “diasasterNumber”.

```
v1 <- v1 |>
  select(-paLoadDate, -iaLoadDate, -hash, -lastRefresh, -id)
new_data <- v2 |>
  left_join(v1, by = "disasterNumber")
colnames(new_data)
```

```
##  [1] "disasterNumber"             "state"
##  [3] "declarationTitle"           "ihProgramDeclared"
##  [5] "paProgramDeclared"          "hmProgramDeclared"
##  [7] "fipsStateCode"              "fipsCountyCode"
##  [9] "id"                         "year"
## [11] "month"                      "day"
## [13] "totalNumberIaApproved"      "totalAmountIhpApproved"
## [15] "totalAmountHaApproved"      "totalAmountOnaApproved"
## [17] "totalObligatedAmountPa"     "totalObligatedAmountCatAb"
## [19] "totalObligatedAmountCatC2g" "totalObligatedAmountHmgp"
```

```
# check the NA value (missing data)
gg_miss_upset(new_data)
```



When the values of the four columns totalNumberIaApproved, totalAmountIhpApproved, totalAmountHaApproved, and totalAmountOnaApproved are NA, we observe that the value of ihProgramDeclared is 0, so we believe that they have not received donations from the Individuals and Households program. Therefore, the missing values in these four columns are set to 0.

```
#Verify whether the guess is correct
rows_with_na <- new_data |>
  filter(
    is.na(totalNumberIaApproved) |
    is.na(totalAmountIhpApproved) |
    is.na(totalAmountHaApproved) |
    is.na(totalAmountOnaApproved)
  )

# If the result is TRUE, the guess is correct, otherwise, the guess is incorrect.
all(rows_with_na$ihProgramDeclared == 0)
```

```
## [1] TRUE
```

```
#Since the value of above is TRUE, so we set four columns' NA value to 0.
new_data <- new_data |>
  mutate(
    totalNumberIaApproved = ifelse(is.na(totalNumberIaApproved), 0, totalNumberIaApproved),
    totalAmountIhpApproved = ifelse(is.na(totalAmountIhpApproved), 0, totalAmountIhpApproved),
    totalAmountHaApproved = ifelse(is.na(totalAmountHaApproved), 0, totalAmountHaApproved),
    totalAmountOnaApproved = ifelse(is.na(totalAmountOnaApproved), 0, totalAmountOnaApproved)
  )

# Since disaster 4571 is the only disaster that totalObligatedAmountPa, totalObligatedAmountCatAb, tota
new_data <- new_data |>
  mutate(
    totalObligatedAmountPa = ifelse(is.na(totalObligatedAmountPa), 0, totalObligatedAmountPa),
    totalObligatedAmountCatAb = ifelse(is.na(totalObligatedAmountCatAb), 0, totalObligatedAmountCatAb),
    totalObligatedAmountCatC2g = ifelse(is.na(totalObligatedAmountCatC2g), 0, totalObligatedAmountCatC2g
  )

#check the NA value again, if the value is FALSE, the dataset haven's any NA value.
any(is.na(new_data))
```

```
## [1] FALSE
```

```
#The value is NA, which means the dataset haven't NA value. We can continue to the follow steps.
```

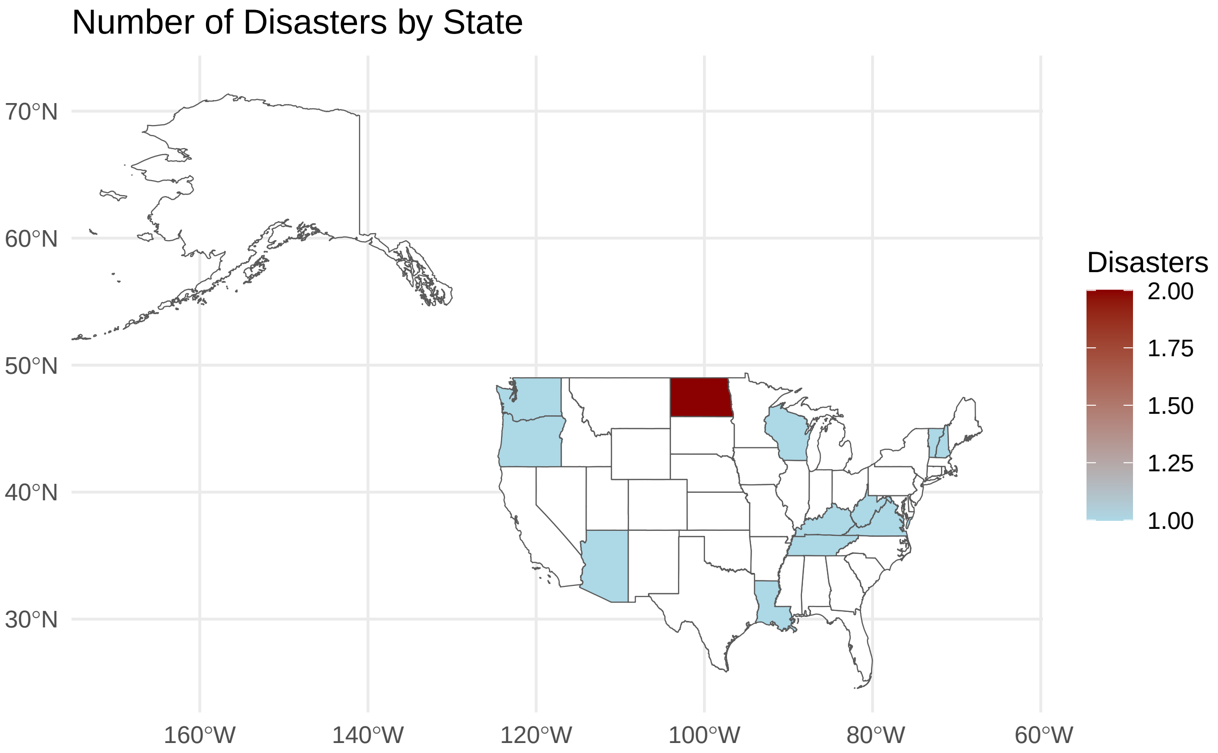
EDA

Disaster by States

```
us_states <- st_read("https://eric.clst.org/assets/wiki/uploads/Stuff/gz_2010_us_040_00_5m.json")
```

```
## Reading layer `gz_2010_us_040_00_5m' from data source
##   `https://eric.clst.org/assets/wiki/uploads/Stuff/gz_2010_us_040_00_5m.json'
##   using driver `GeoJSON'
## Simple feature collection with 52 features and 5 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -179.1473 ymin: 17.92688 xmax: 179.7785 ymax: 71.35256
## Geodetic CRS:  WGS 84
```

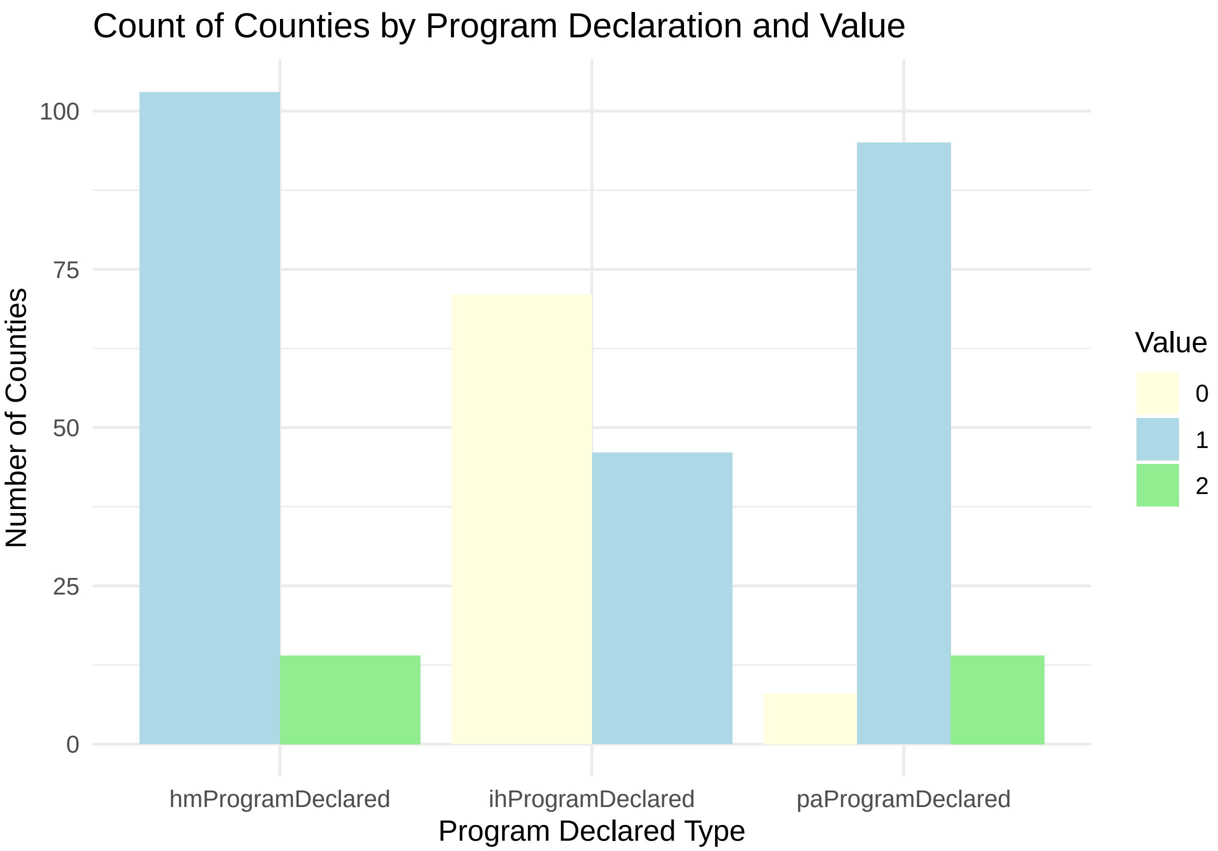
```
# Read the disaster data
disaster_data <- new_data

# Process the disaster data
# Dropping duplicates to ensure we count each disaster only once per state
state_disaster_counts <- disaster_data |>
  distinct(disasterNumber, fipsStateCode, .keep_all = TRUE) |>
  count(fipsStateCode) |>
  mutate(fipsStateCode = as.character(fipsStateCode)) |>
  rename(StateFIPS = fipsStateCode, Disasters = n)

# Make sure the FIPS codes have leading zeros
state_disaster_counts$StateFIPS <- str_pad(state_disaster_counts$StateFIPS, width = 2, side = "left", pa

# Join the disaster data with the state geometries
merged_data <- left_join(us_states, state_disaster_counts, by = c("STATE" = "StateFIPS"))

# Plot the map
plot1 <- ggplot(data = merged_data) +
  geom_sf(aes(fill = Disasters)) +
  scale_fill_gradient(low = "lightblue", high = "darkred", na.value = "white") +
  theme_minimal() +
  labs(title = "Number of Disasters by State", fill = "Disasters") +
  coord_sf(xlim = c(-170, -65), ylim = c(25, 72))

# Display the plot
print(plot1)
```



According to the results, there will be fewer flood disasters in the United States in 2020 and 2021, and the distribution will be more scattered.

Count of Counties by Program Declaration and Value

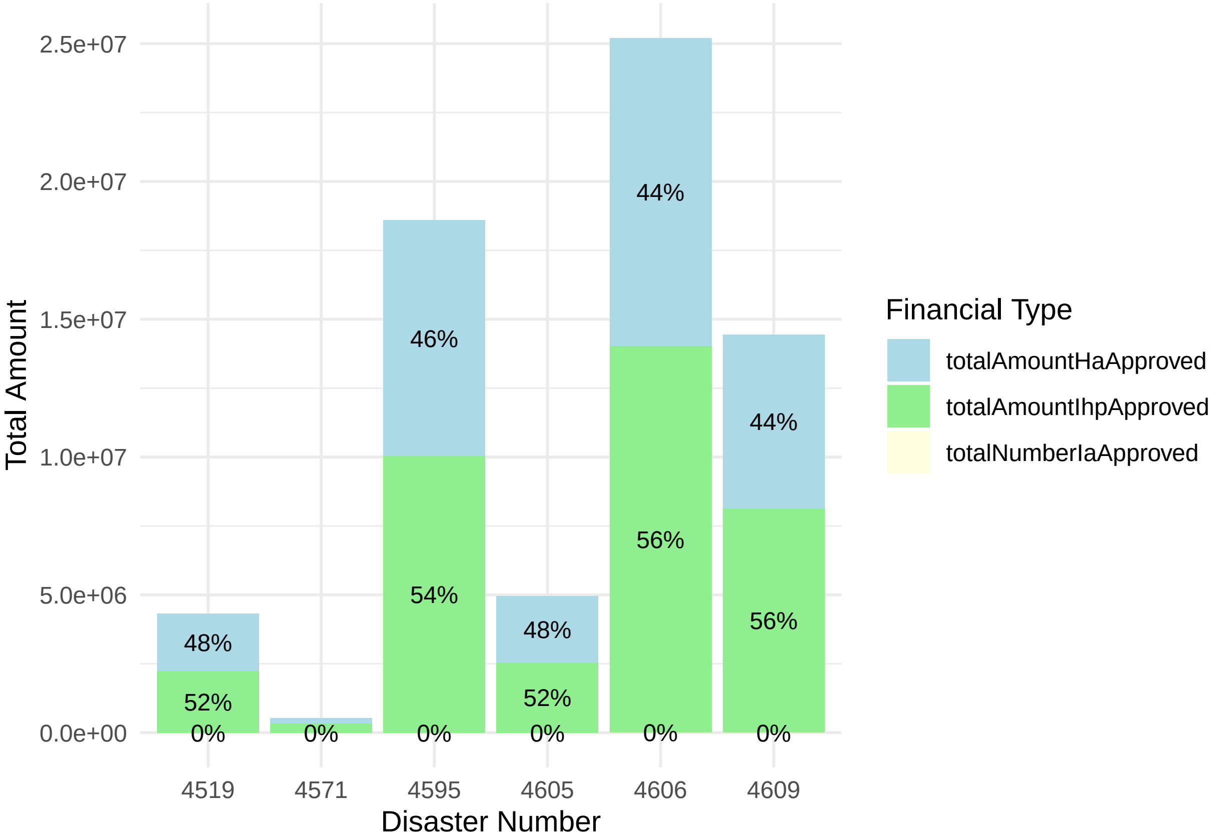
```r
declared_data <- new_data |>
  group_by(fipsStateCode, fipsCountyCode) |>
  summarise(
    ihProgramDeclared = sum(ihProgramDeclared, na.rm = TRUE),
    paProgramDeclared = sum(paProgramDeclared, na.rm = TRUE),
    hmProgramDeclared = sum(hmProgramDeclared, na.rm = TRUE),
    .groups = "drop"
  )|>
  mutate(
    fipsStateCode = str_pad(fipsStateCode, width = 2, pad = "0"),
    fipsCountyCode = str_pad(fipsCountyCode, width = 3, pad = "0"),
    FipsCode = paste0(fipsStateCode, fipsCountyCode)
  )|>
  select(
    -fipsStateCode,
    -fipsCountyCode
  )
# fipscode.csv is from https://transition.fcc.gov/oet/info/maps/census/fips/fips.txt and has been proce
fips_codes <- read_csv("fipscode.csv", col_names = TRUE)

fips_codes <- fips_codes |>
  mutate(`FIPS code` = as.character(`FIPS code`))

declared_data_with_names <- declared_data |>
  left_join(fips_codes, by = c("FipsCode" = "FIPS code"))|>
  distinct(name, .keep_all = TRUE)
count_ihProgramDeclared <- declared_data_with_names |>
  count(ihProgramDeclared, name = "Count") |>
  mutate(Program = 'IH Program')

count_paProgramDeclared <- declared_data_with_names |>
  count(paProgramDeclared, name = "Count") |>
  mutate(Program = 'PA Program')

count_hmProgramDeclared <- declared_data_with_names |>
  count(hmProgramDeclared, name = "Count") |>
  mutate(Program = 'HM Program')

program_counts <- bind_rows(count_ihProgramDeclared, count_paProgramDeclared, count_hmProgramDeclared)|
  pivot_longer(
    cols = c(ihProgramDeclared, paProgramDeclared, hmProgramDeclared),
    names_to = "ProgramDeclaredType",
    values_to = "Value"
  )|>
  filter(!is.na(Value))

ggplot(program_counts, aes(x = ProgramDeclaredType, y = Count, fill = as.factor(Value))) +
  geom_bar(stat = "identity", position = position_dodge()) +
  labs(title = "Count of Counties by Program Declaration and Value",
       x = "Program Declared Type",
       y = "Number of Counties",
       fill = "Value") +
  theme_minimal() +
  scale_fill_manual(values = c("lightyellow", "lightblue", "lightgreen"))
```



It can be seen that a county generally receives less than or equal to one aid plan from the same source. And the number of assistance programs from the Individuals and Households program and Public Assistance program is greater than that from the Individual Assistance program

The total amount approved from different program

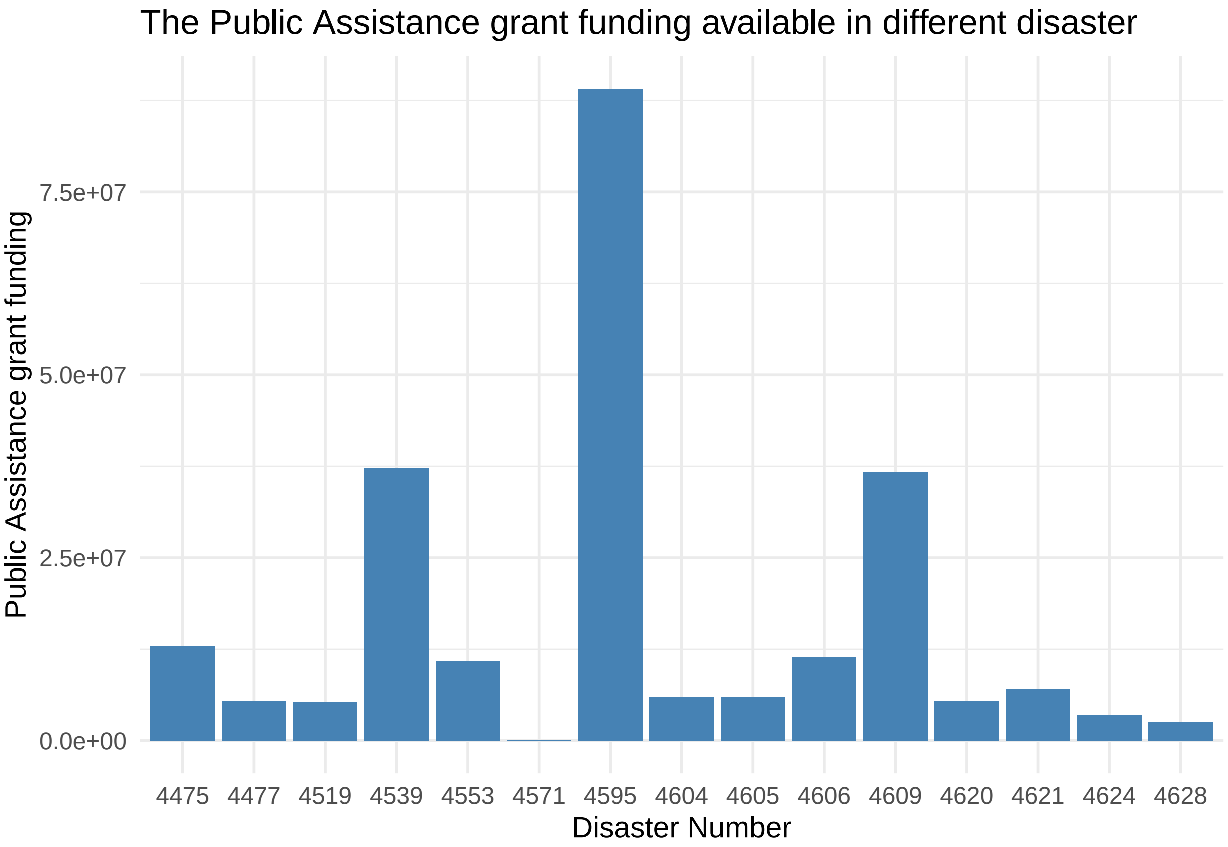
```
Approved_data <- new_data |>
  distinct(disasterNumber,.keep_all = TRUE) |>
  mutate(as.factor(disasterNumber))|>
  filter(totalNumberIaApproved != 0, totalAmountIhpApproved != 0, totalAmountHaApproved != 0)|>
  select(disasterNumber, totalNumberIaApproved, totalAmountIhpApproved, totalAmountHaApproved) |>
  pivot_longer(
    cols = -disasterNumber,
    names_to = "FinancialType",
    values_to = "Amount"
  )|>
  group_by(disasterNumber) |>
  mutate(Proportion = Amount / sum(Amount),
         LabelPosition = cumsum(Amount) - (0.5 * Amount)) |>
  ungroup()

ggplot(Approved_data, aes(x = as.factor(disasterNumber), y = Amount, fill = FinancialType)) +
  geom_col() +
  scale_fill_manual(values = c("lightblue", "lightgreen", "lightyellow")) +
  geom_text(
    aes(label = scales::percent(Proportion, accuracy = 1), y = LabelPosition),
    color = "black",
    size = 3,
    vjust = 0.5,
    check_overlap = TRUE
  ) +
  theme_minimal() +
  labs(x = "Disaster Number", y = "Total Amount", fill = "Financial Type")
```



The total amount approved for Housing Assistance is similar to The total amount approved for the Individual and Households Program, and basically no The number of disaster assistance applications that were approved for Individual Assistance.

The Public Assistance grant funding available in different disaster.

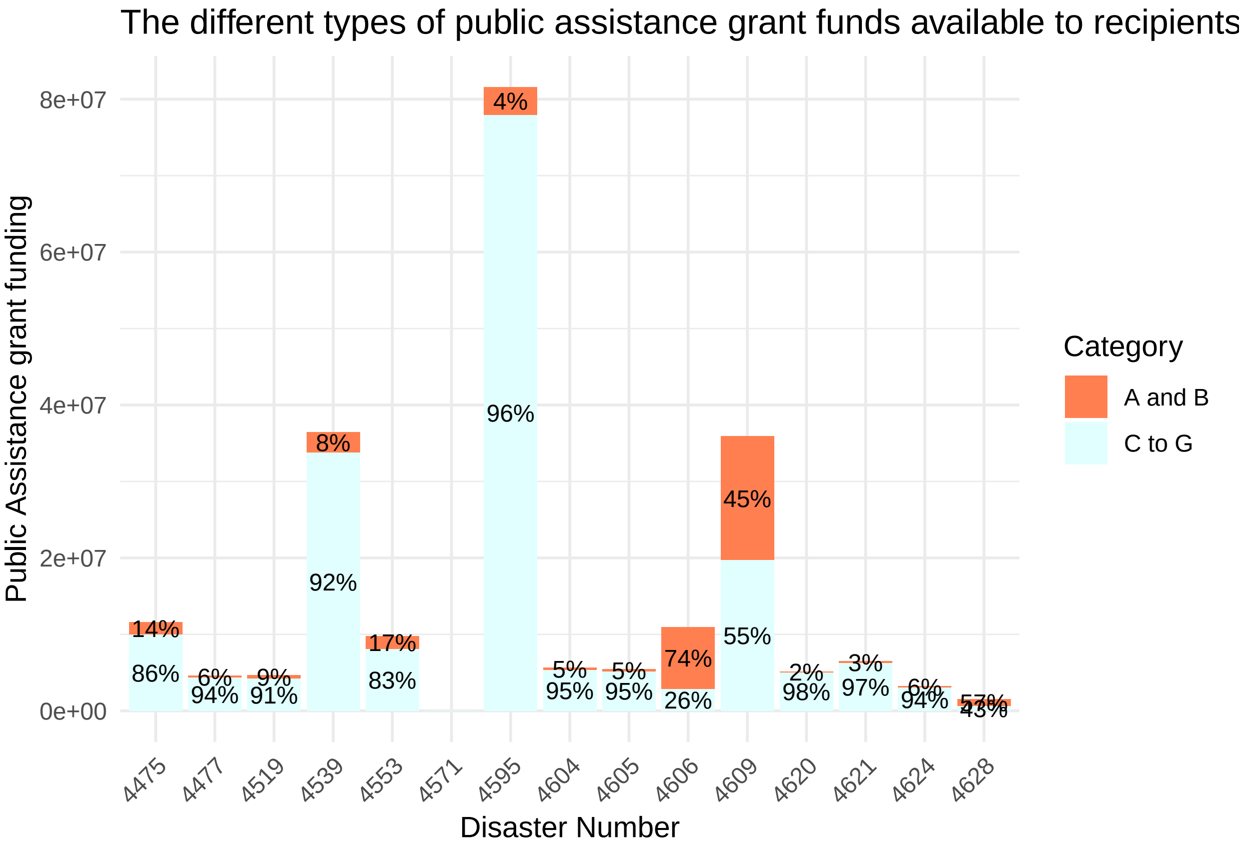
```
AmongPA <- new_data |>
  distinct(disasterNumber,.keep_all = TRUE) |>
  select(disasterNumber,totalObligatedAmountPa)|>
  mutate(as.factor(disasterNumber))

ggplot(AmongPA) +
  aes(
    x = `as.factor(disasterNumber)`,
    y = totalObligatedAmountPa
  ) +
  geom_col(fill = "#4682B4") +
  labs(x = "Disaster Number", y = "Public Assistance grant funding", title = "The Public Assistance gran
  theme_minimal()
```



The different types of public assistance grant funds available to recipients in U.S. dollars

```
Funding <- new_data |>
  distinct(disasterNumber, .keep_all = TRUE) |>
  mutate(disasterNumber = as.factor(disasterNumber)) |>
  select(disasterNumber, totalObligatedAmountCatAb, totalObligatedAmountCatC2g) |>
  pivot_longer(
    cols = starts_with("totalObligatedAmount"),
    names_to = "Category",
    values_to = "Amount"
  )|>
  mutate(Category = recode(Category,
                            "totalObligatedAmountCatAb" = "A and B",
                            "totalObligatedAmountCatC2g" = "C to G"))
Funding_long <- Funding |>
  group_by(disasterNumber) |>
  mutate(TotalAmount = sum(Amount)) |>
  ungroup()

ggplot(Funding_long, aes(x = as.factor(disasterNumber), y = Amount, fill = Category)) +
  geom_col() +
  scale_fill_manual(values = c("#FF7F50", "#E1FFFF"),
                    labels = c("A and B", "C to G")) +
  geom_text(
    aes(label = scales::percent(Amount / TotalAmount, accuracy = 1)),
    position = position_stack(vjust = 0.5),
    size = 3,
    color = "black"
  ) +
  theme_minimal() +
  labs(x = "Disaster Number", y = "Public Assistance grant funding", fill = "Category",title="The diffe
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Warning: Removed 2 rows containing missing values (`geom_text()`).
```



It can be seen that most public assistance grant funds available are of type c to g.

Combine Flood Information With Census

read data and combine data

```
# read data
houseunit2020 <- read_csv("houseunit2020.csv")
houseunit2021 <- read_csv("houseunit2021.csv")
population2020 <- read_csv("population2020.csv")
population2021 <- read_csv("population2021.csv")
poverty2020 <- read_csv("poverty2020.csv")
poverty2021 <- read_csv("poverty2021.csv")
```

```r
# combine three datasets with same year, get Fipscode
data_2020 <- population2020 |>
  left_join(houseunit2020, by = "Geography") |>
  left_join(poverty2020, by = "Geography") |>
  mutate(Fipscode = substr(Geography, nchar(Geography)-4, nchar(Geography))) |>
  select(Fipscode, everything()) |>
  select(-Geography)

data_2021 <- population2021 |>
  left_join(houseunit2021, by = "Geography") |>
  left_join(poverty2021, by = "Geography") |>
  mutate(Fipscode = substr(Geography, nchar(Geography)-4, nchar(Geography))) |>
  select(Fipscode, everything()) |>
  select(-Geography)

# combine datasets with flood information dataset. If year is 2020, combine data_2020, otherwise, combi
flood_data <- new_data |>
  mutate(
    fipsStateCode = str_pad(fipsStateCode, width = 2, pad = "0"),
    fipsCountyCode = str_pad(fipsCountyCode, width = 3, pad = "0"),
    Fipscode = paste0(fipsStateCode, fipsCountyCode)
  ) |>
  select(Fipscode, everything())
# Split the flood_data into two data frames
flood_data_2020 <- filter(flood_data, year == 2020)
flood_data_2021 <- filter(flood_data, year == 2021)
# Join with data_2020 and data_2021
flood_data_2020 <- flood_data_2020 |>
  left_join(data_2020, by = "Fipscode")
flood_data_2021 <- flood_data_2021 |>
  left_join(data_2021, by = "Fipscode")

# Combine the two data frames
final_data <- bind_rows(flood_data_2020, flood_data_2021)
```

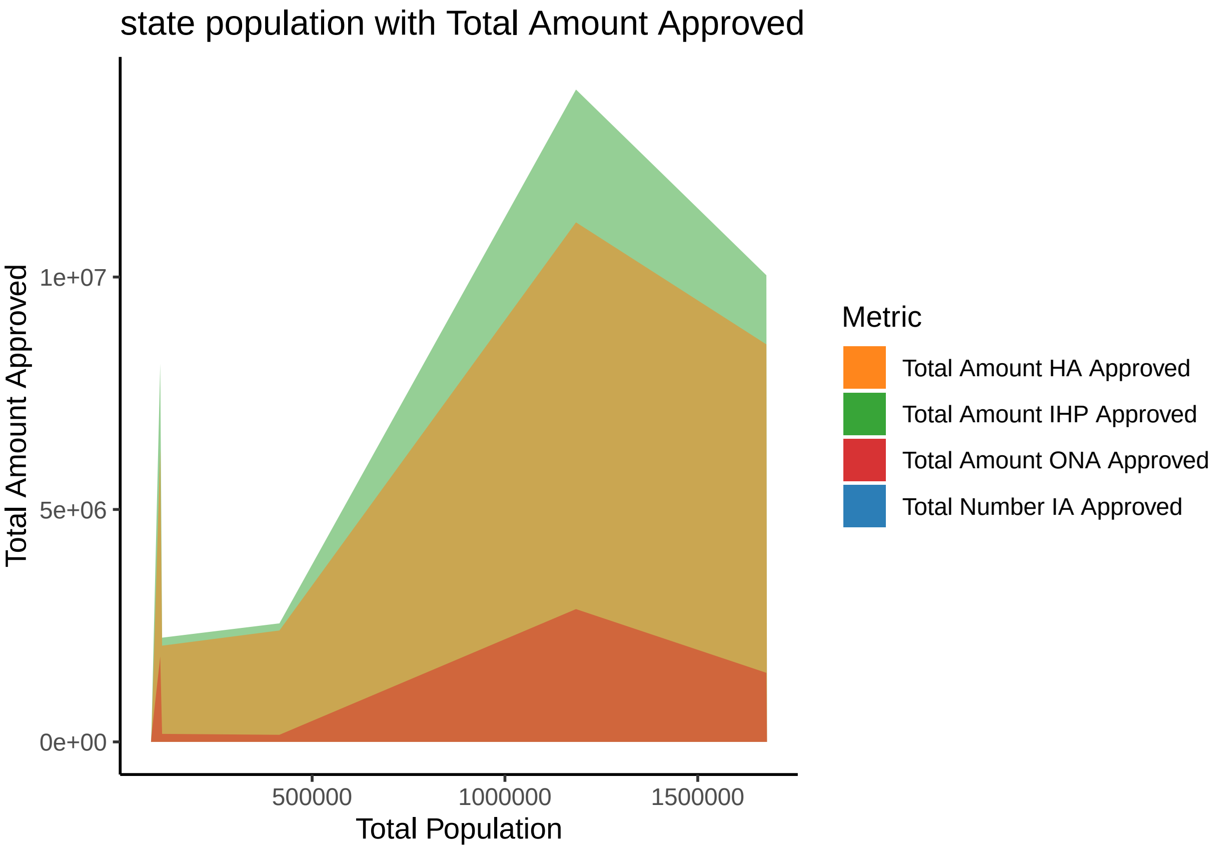
Group by state and disasterNumber

```r
data_withflood <- final_data|>
  group_by(disasterNumber, state) %>%
  summarise(
    Total_population = sum(`Total population`, na.rm = TRUE),
    HOUSE_UNIT = sum(`HOUSE UNIT`, na.rm = TRUE),
    Population_poverty_status_determined = sum(`Population for whom poverty status is determined`, na.r
    totalNumberIaApproved = first(totalNumberIaApproved),
    totalAmountIhpApproved = first(totalAmountIhpApproved),
    totalAmountHaApproved = first(totalAmountHaApproved),
    totalAmountOnaApproved = first(totalAmountOnaApproved),
    totalObligatedAmountPa = first(totalObligatedAmountPa),
    totalObligatedAmountCatAb = first(totalObligatedAmountCatAb),
    totalObligatedAmountCatC2g = first(totalObligatedAmountCatC2g),
    totalObligatedAmountHmgp = first(totalObligatedAmountHmgp),
    Sum_Total_population = sum(`Total population`, na.rm = TRUE),
    Sum_HOUSE_UNIT = sum(`HOUSE UNIT`, na.rm = TRUE),
    Sum_Population_poverty_status_determined = sum(`Population for whom poverty status is determined`,
  )
```

EDA after add datasets

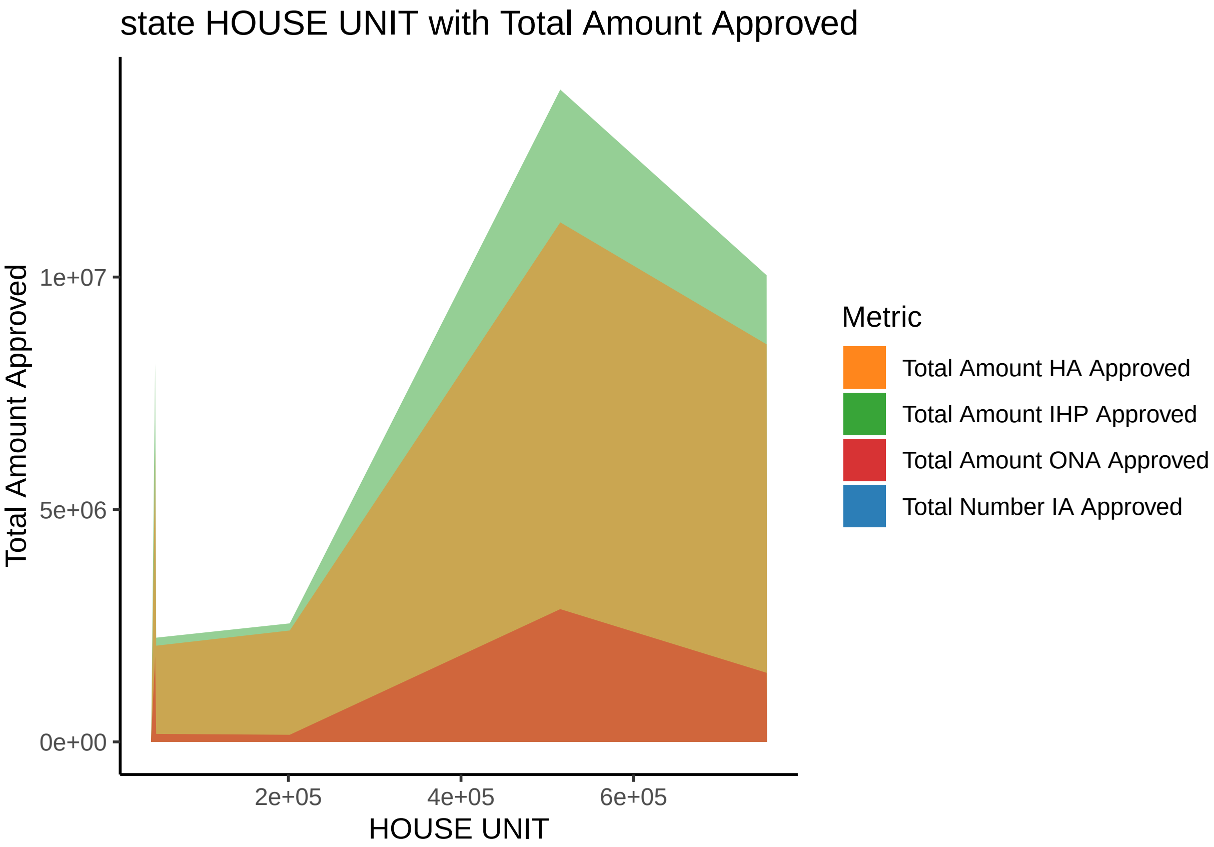
state population with Total Amount Approved

```
amount_data <- data_withflood |>
  filter(!(totalNumberIaApproved == 0 &
           totalAmountIhpApproved == 0 &
           totalAmountHaApproved == 0 &
           totalAmountOnaApproved == 0))

ggplot(amount_data, aes(x = Sum_Total_population)) +
  geom_area(aes(y = totalNumberIaApproved, fill = 'totalNumberIaApproved'), alpha = 0.5) +
  geom_area(aes(y = totalAmountIhpApproved, fill = 'totalAmountIhpApproved'), alpha = 0.5) +
  geom_area(aes(y = totalAmountHaApproved, fill = 'totalAmountHaApproved'), alpha = 0.5) +
  geom_area(aes(y = totalAmountOnaApproved, fill = 'totalAmountOnaApproved'), alpha = 0.5) +
  scale_fill_manual(values = c('totalNumberIaApproved' = '#1f77b4',
                               'totalAmountIhpApproved' = '#2ca02c',
                               'totalAmountHaApproved' = '#ff7f0e',
                               'totalAmountOnaApproved' = '#d62728'),
                    labels = c('totalNumberIaApproved' = 'Total Number IA Approved',
                               'totalAmountIhpApproved' = 'Total Amount IHP Approved',
                               'totalAmountHaApproved' = 'Total Amount HA Approved',
                               'totalAmountOnaApproved' = 'Total Amount ONA Approved')) +
  labs(x = "Total Population", y = "Total Amount Approved", fill = "Metric" ,title = "state population
  theme_classic()
```



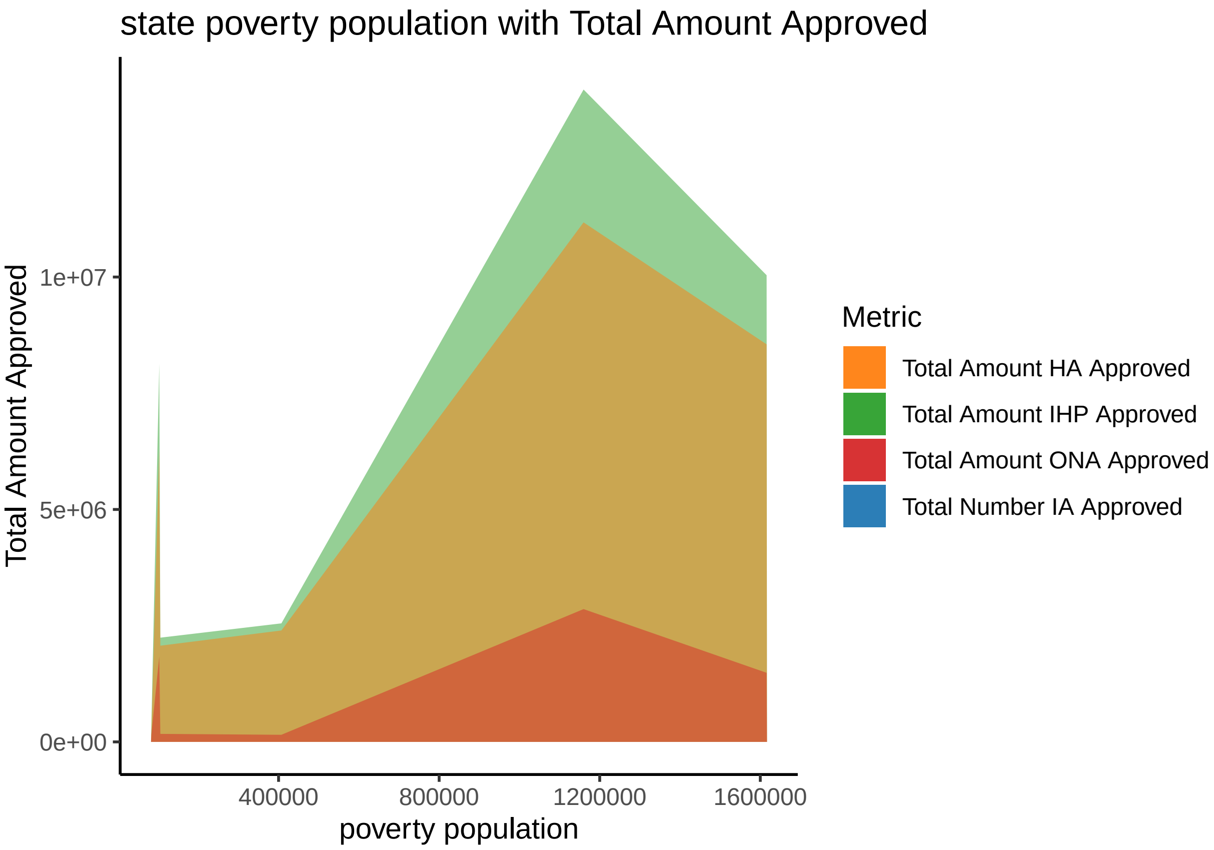
state HOUSE UNIT with Total Amount Approved

```
ggplot(amount_data, aes(x = Sum_HOUSE_UNIT)) +
  geom_area(aes(y = totalNumberIaApproved, fill = 'totalNumberIaApproved'), alpha = 0.5) +
  geom_area(aes(y = totalAmountIhpApproved, fill = 'totalAmountIhpApproved'), alpha = 0.5) +
  geom_area(aes(y = totalAmountHaApproved, fill = 'totalAmountHaApproved'), alpha = 0.5) +
  geom_area(aes(y = totalAmountOnaApproved, fill = 'totalAmountOnaApproved'), alpha = 0.5) +
  scale_fill_manual(values = c('totalNumberIaApproved' = '#1f77b4',
                               'totalAmountIhpApproved' = '#2ca02c',
                               'totalAmountHaApproved' = '#ff7f0e',
                               'totalAmountOnaApproved' = '#d62728'),
                    labels = c('totalNumberIaApproved' = 'Total Number IA Approved',
                               'totalAmountIhpApproved' = 'Total Amount IHP Approved',
                               'totalAmountHaApproved' = 'Total Amount HA Approved',
                               'totalAmountOnaApproved' = 'Total Amount ONA Approved')) +
  labs(x = "HOUSE UNIT", y = "Total Amount Approved", fill = "Metric" ,title = "state HOUSE UNIT with T
  theme_classic()
```



state poverty population with Total Amount Approved

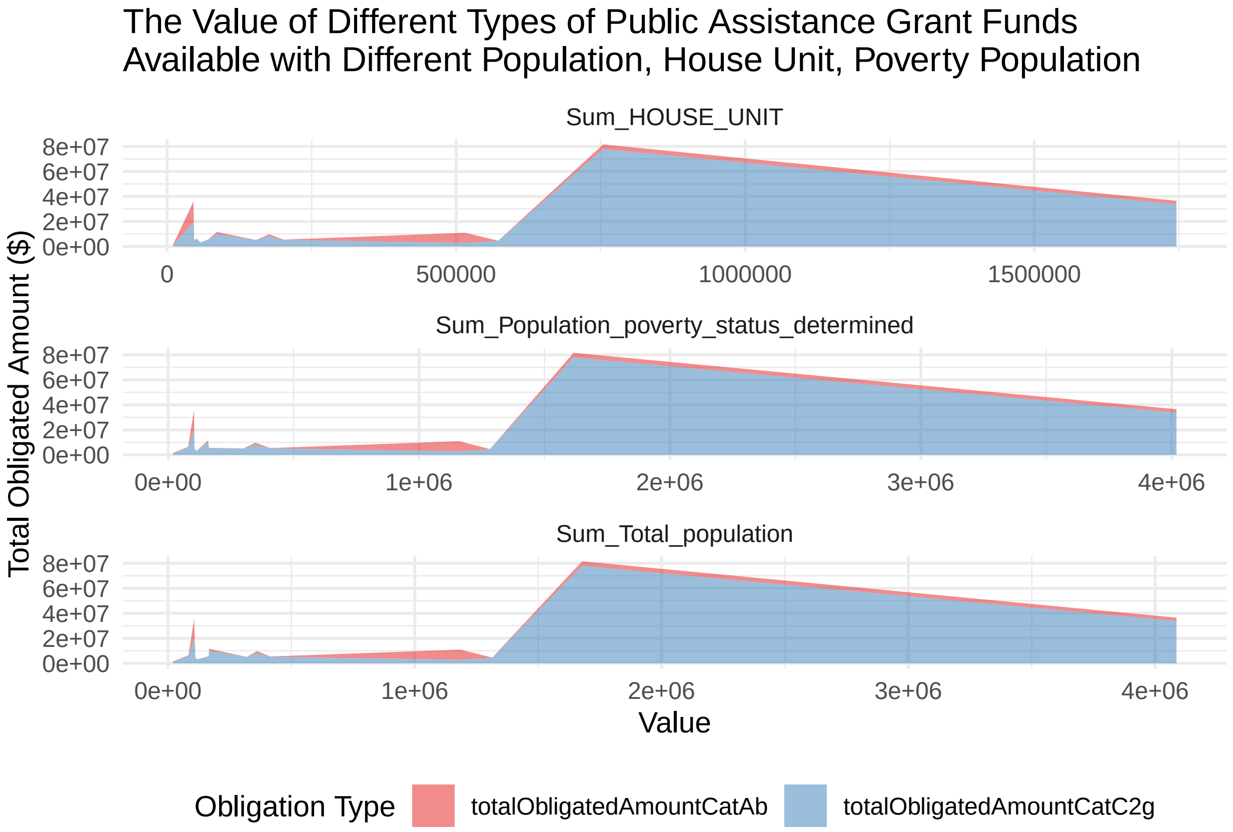
```r
ggplot(amount_data, aes(x = Sum_Population_poverty_status_determined)) +
  geom_area(aes(y = totalNumberIaApproved, fill = 'totalNumberIaApproved'), alpha = 0.5) +
  geom_area(aes(y = totalAmountIhpApproved, fill = 'totalAmountIhpApproved'), alpha = 0.5) +
  geom_area(aes(y = totalAmountHaApproved, fill = 'totalAmountHaApproved'), alpha = 0.5) +
  geom_area(aes(y = totalAmountOnaApproved, fill = 'totalAmountOnaApproved'), alpha = 0.5) +
  scale_fill_manual(values = c('totalNumberIaApproved' = '#1f77b4',
                               'totalAmountIhpApproved' = '#2ca02c',
                               'totalAmountHaApproved' = '#ff7f0e',
                               'totalAmountOnaApproved' = '#d62728'),
                    labels = c('totalNumberIaApproved' = 'Total Number IA Approved',
                               'totalAmountIhpApproved' = 'Total Amount IHP Approved',
                               'totalAmountHaApproved' = 'Total Amount HA Approved',
                               'totalAmountOnaApproved' = 'Total Amount ONA Approved')) +
  labs(x = "poverty population", y = "Total Amount Approved", fill = "Metric" ,title = "state poverty po
  theme_classic()
```



The above three figures all illustrate that the value of The total amount approved will increase with the increase of population, the number of house units and the poverty population.

the value of different types of public assistance grant funds available with different population, house unit, poverty population

```
fund_data <- data_withflood |>
  filter(!(totalObligatedAmountCatAb == 0 &
           totalObligatedAmountCatC2g == 0))|>
  pivot_longer(
    cols = c(Sum_Total_population, Sum_HOUSE_UNIT, Sum_Population_poverty_status_determined),
    names_to = "TotalType",
    values_to = "TypeValue"
  ) |>
  select(
    TotalType,
    TypeValue,
    totalObligatedAmountCatAb,
    totalObligatedAmountCatC2g
  ) |>
  pivot_longer(
    cols = starts_with("totalObligatedAmount"),
    names_to = "ObligationType",
    values_to = "Amount"
  )

ggplot(fund_data, aes(x = TypeValue, y = Amount, fill = ObligationType)) +
  geom_area(position = 'stack', alpha = 0.5) +
  facet_wrap(~TotalType, scales = 'free_x', ncol = 1) +
  scale_fill_brewer(palette = "Set1", name = "Obligation Type") +
  labs(
    x = "Value",
    y = "Total Obligated Amount ($)",
    title = "The Value of Different Types of Public Assistance Grant Funds\nAvailable with Different Pop
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```



The above three figures all illustrate that the value of The Public Assistance grant funding will increase with the increase in population, the number of house units, and the number of poor people.