

# MA615-Extended Mapping Assignment

Chenxun Li && Minqi Li

11/9/2020

## The Number of Hurricanes in Each Area

### Data processing

At first, we need to read the data from mac.

```
PublicAssistanceFundedProjectsDetails <- read_csv("PublicAssistanceFundedProjectsDetails.csv")
```

Then, we use 'filter()' to extract 'hurricane' from the data set.

```
#select the 'hurricane'  
original_hurricane <- filter(PublicAssistanceFundedProjectsDetails, incidentType=="Hurricane")
```

Next, we select the columns which we are interested in.

```
#select specific variables  
hurr_num <- original_hurricane %>%  
  select(disasterNumber, declarationDate, county, countyCode, state, stateCode, stateNumberCode)  
  
#way1: use 'separate()' to select year  
hurr_num %<>% separate('declarationDate', c("Year", "Month", "Day"), sep = "-")  
hurr_num %<>% select(-c(Month, Day))
```

Finally, we use summarize to get the final data set for graph and mapping.

```
# data set for bar chart  
hurr_num_bar <- hurr_num %>%  
  group_by(state) %>%  
  summarize(hurricanes=n_distinct(disasterNumber)) %>%  
  arrange(desc(hurricanes))  
  
#data set for mapping  
hurr_num_map<- hurr_num %>%  
  group_by(state,stateCode,stateNumberCode,county,countyCode,Year) %>%  
  summarize(hurricanes=n_distinct(disasterNumber)) %>%  
  arrange(desc(hurricanes))
```

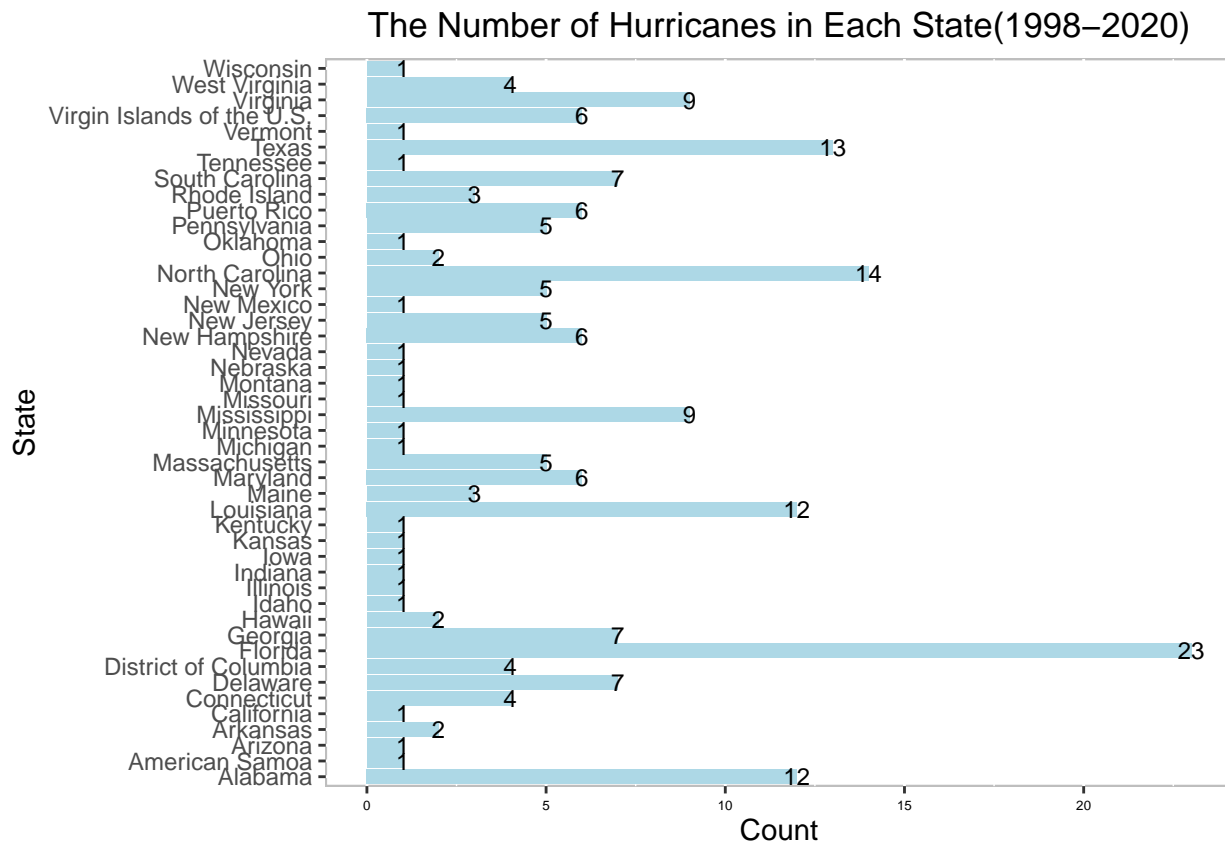
### Bar chart

```
ggplot(hurr_num_bar, aes(x=state, y=hurricanes))+  
  geom_bar(stat = "identity", fill = "lightblue")+  
  #change to horizontal visualization  
  coord_flip()+  
  #add numbers of hurricanes  
  geom_text(aes(label = hurricanes), vjust = 0.5, colour = "black", position = position_dodge(.9),
```

```

    size = 3))+
  theme(panel.background=element_rect(fill='transparent',
                                       color = "gray"),
        axis.text.x = element_text(hjust = 0.5,
                                    vjust = 0.5,color = "black",size=5))+
  #change the name of x, y, and title
  xlab("State")+ylab("Count")+
  ggtitle("The Number of Hurricanes in Each State(1998-2020)")+
  theme(plot.title = element_text(hjust = 0.5))

```



## Mapping

First, we load the data of longitude and latitude, and join them with our data

```

#get the data set which includes all longitude and latitude of the county/state.
MainStates <- map_data("state")
AllCounty <- map_data("county")

#make the letter of county lower
hurr_num_map$county %<>% tolower()

#combine the data set to get the longitude and latitude
hurr_map <- left_join(hurr_num_map, AllCounty, by = c("county"="subregion"))

```

Then, we can use the nes data set to draw a map.

```

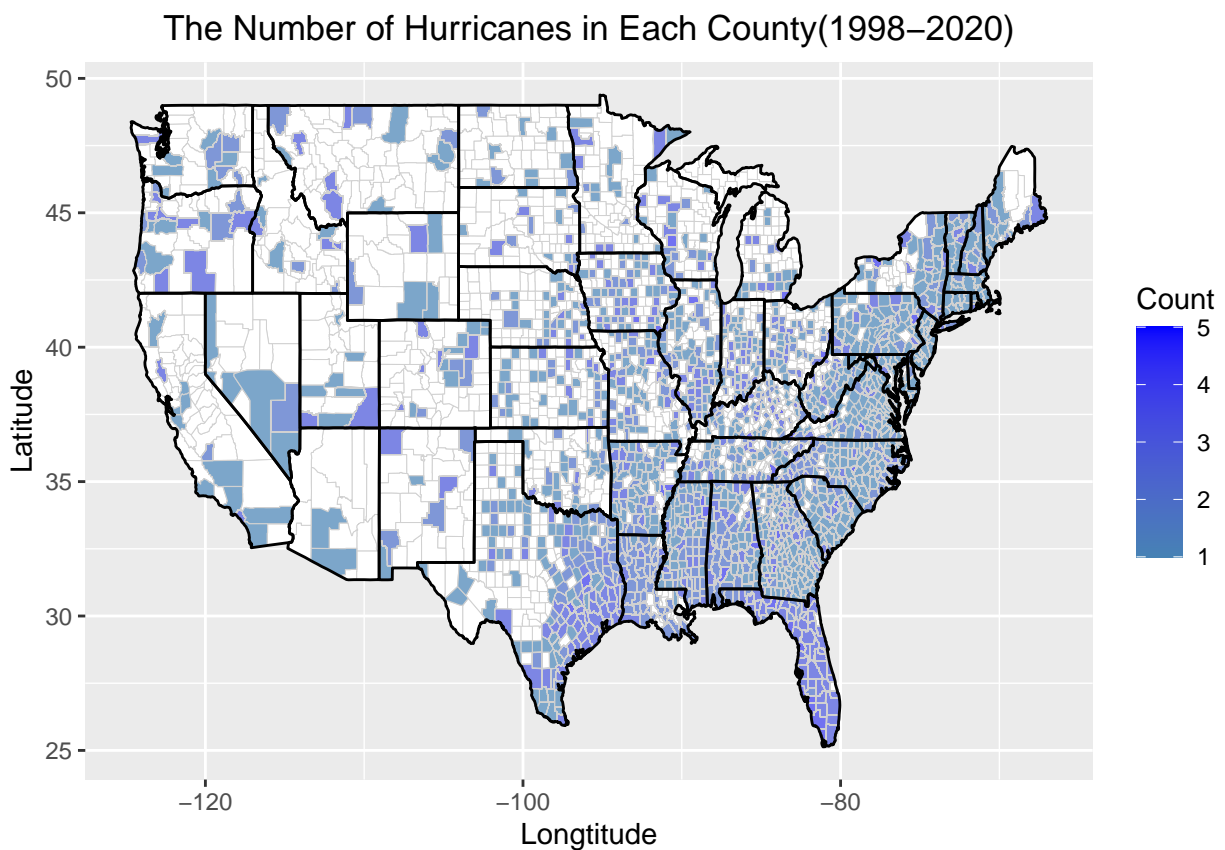
ggplot() +
  #get the map of county

```

```

geom_polygon(data=AllCounty, aes(x=long, y=lat, group=group),
             color="gray", fill="white", size = .1 ) +
#add the data set we have
geom_polygon(data = hurr_map, aes(x = long, y = lat, group = group, fill = 'hurricanes'),
             color = "grey", size = 0.2, alpha = 1.6) +
#display discrete values on a map
scale_fill_gradient(low="steelblue",high="blue")+
#add the state borders
geom_polygon(data=MainStates, aes(x=long, y=lat, group=group),color="black", fill="white",
             size = .5, alpha = .3) +
#change the name of x, y, and title
ggtitle("The Number of Hurricanes in Each County(1998-2020)") +
xlab("Longitude")+ylab("Latitude")+labs(fill="Count")+
theme(plot.title = element_text(hjust = 0.5))

```



## Different Expenses in Each Damage Category and Area

### Data processing

First, we need select and summarize the specific columns.

```

#select information we need
hurricane_damage_cate_sum <- original_hurricane%>%
  group_by(damageCategoryCode)%>%
  summarise(projectAmount_sum=sum(projectAmount)/1000000,
            federalShareObligated_sum=sum(federalShareObligated)/1000000,
            totalObligated_sum=sum(totalObligated)/1000000)

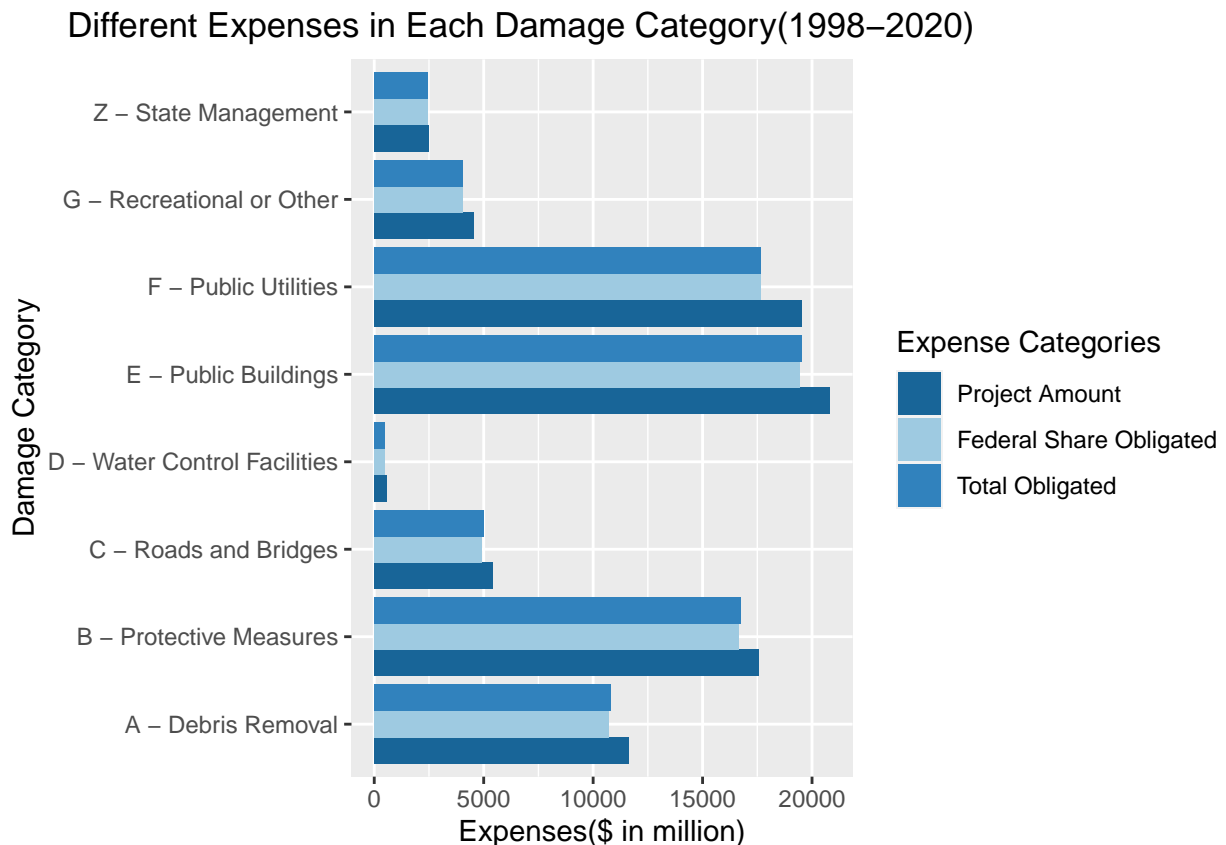
```

Then, we use 'melt()' to change the format of the table.

```
#get final data set
final_hurricane <- melt(hurricane_damage_cate_sum,id.vars = c("damageCategoryCode"),
  variable.name = "cost_cate", value.name= "cost_dollar")
```

## Bar chart

```
ggplot(final_hurricane, mapping = aes(x=damageCategoryCode, y=cost_dollar,fill=cost_cate))+
  #change to horizontal visualization
  coord_flip()+
  geom_bar(stat = "identity",position = "dodge")+
  #change the name of x, y, and title
  ggtitle("Different Expenses in Each Damage Category(1998-2020)")+
  xlab("Damage Category")+
  ylab("Expenses($ in million)")+
  #change the name of fill
  labs(fill="Expense Categories")+
  #change the color and labels
  scale_fill_manual(values = c("#186598","#9ecae1","#3182bd"),
    labels =c("Project Amount","Federal Share Obligated","Total Obligated"))+
  theme(plot.title = element_text(hjust = 0.7))
```



## Mapping

First, we still summarize the columns we want.

```

#way2: use 'substring()' to select year
hurricane_ob <- original_hurricane%>%
  group_by(region=state,subregion=county,year=substring(declarationDate,1,4))%>%
  srvyr::summarize(totalObligated_sum=sum(totalObligated),count=n())

#add new variable which is computed in million
hurricane_ob$totalObligated_sum_million <- hurricane_ob$totalObligated_sum/1000000

#make the letteew to lowercase to merge
hurricane_ob$region <- tolower(hurricane_ob$region)
hurricane_ob$subregion <- tolower(hurricane_ob$subregion)

```

Then, we still add the county's longitude and latitude to our data set.

```

#turn data from the maps package in to a data frame suitable for plotting with ggplot2
map_states <- map_data("county", unique(hurricane_ob$state))

#add the state border data
map_states_border <- map_data("state",unique(hurricane_ob$state))

#get the final data set
final_hurricane_ob <- merge(hurricane_ob,map_states,by=c("region","subregion"))

```

Finally, we can plot the map.

(1).Total Obligated Expenses in Each Area

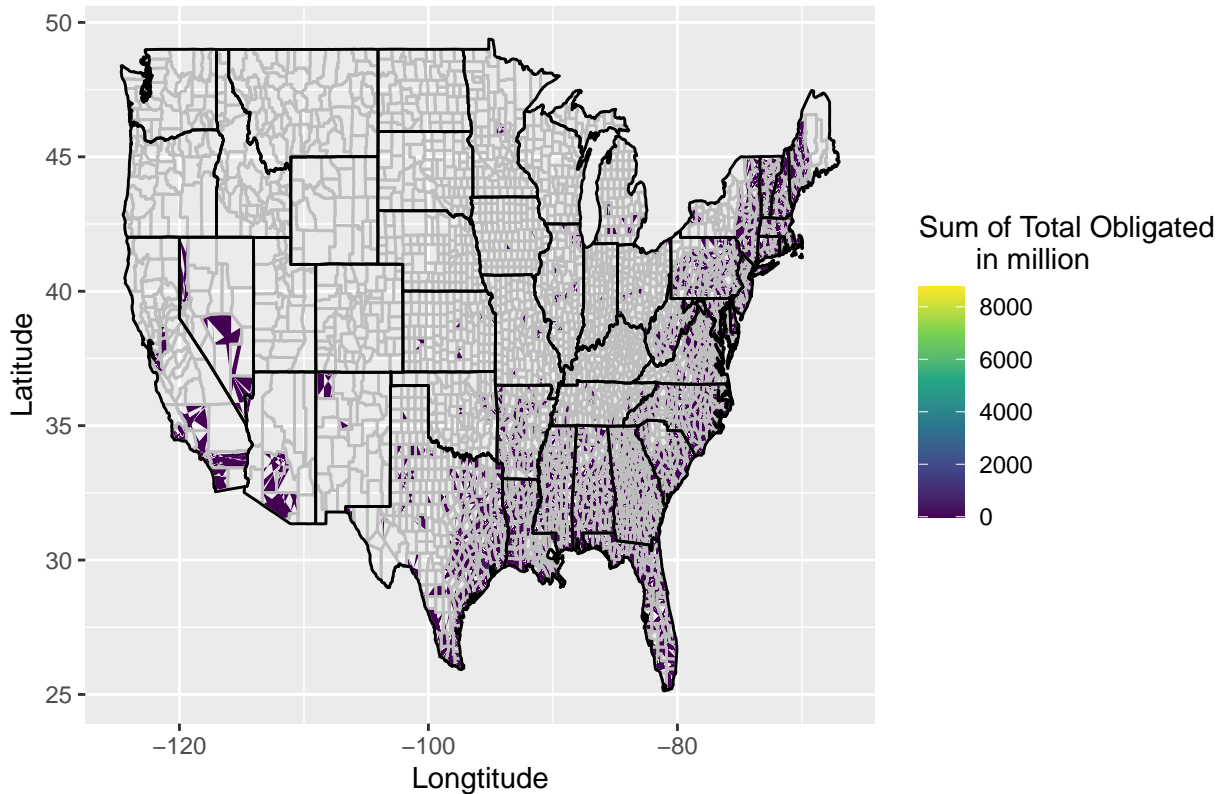
```

#use package 'viridis' to maintain gradual change visual effect
options(
  ggplot2.continuous.colour = "viridis",
  ggplot2.continuous.fill = "viridis"
)

ggplot()+
  geom_polygon(final_hurricane_ob,
    mapping = aes(x=long,y=lat,group=group,fill=totalObligated_sum_million))+
  #connects the observations in the order in which they appear in the 'map_states'
  geom_path(map_states, mapping=aes(x=long, y=lat, group=group),color="grey")+
  #state borders
  geom_path(map_states_border, mapping=aes(x=long, y=lat, group=group),color="black")+
  #change the range of x and y
  xlim(min(map_states$long),max(map_states$long))+
  ylim(min(map_states$lat),max(map_states$lat))+
  #change the name of x, y, fill and title
  xlab("Longitude")+ylab("Latitude")+
  labs(fill="Sum of Total Obligated
    in million")+
  ggtitle("Total Obligated Expenses in Each Area(1998-2020)")+
  theme(plot.title = element_text(hjust = 0.5))

```

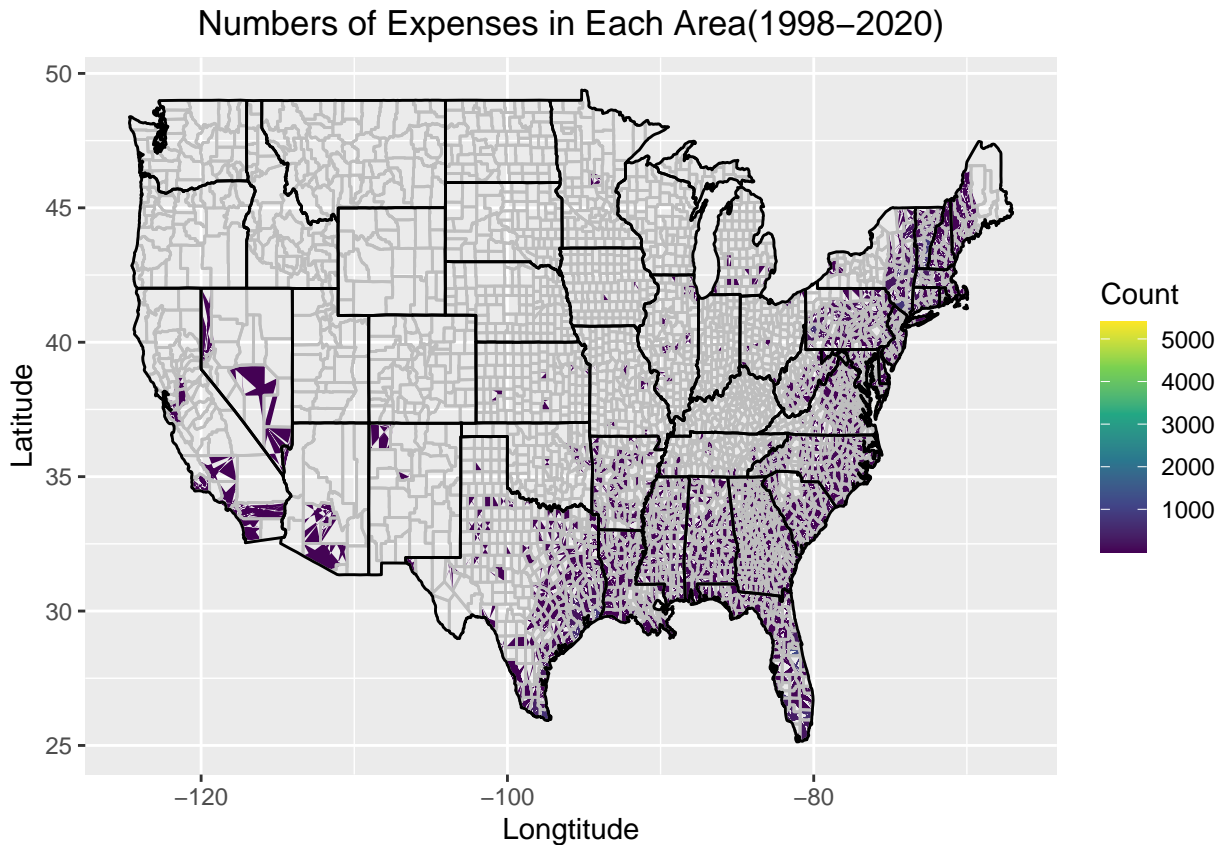
## Total Obligated Expenses in Each Area(1998–2020)



(2).Numbers of Expenses in Each Area

```
#use package 'viridis' to maintain gradual change visual effect
options(
  ggplot2.continuous.colour = "viridis",
  ggplot2.continuous.fill = "viridis"
)

ggplot()+
  geom_polygon(final_hurricane_ob, mapping = aes(x=long,y=lat,group=group, fill=count))+
  #connects the observations in the order in which they appear in the 'map_states'
  geom_path(map_states, mapping=aes(x=long, y=lat, group=group),color="grey")+
  #state borders
  geom_path(map_states_border, mapping=aes(x=long, y=lat, group=group),color="black")+
  #change the range of x and y
  xlim(min(map_states$long),max(map_states$long))+
  ylim(min(map_states$lat),max(map_states$lat))+
  #change the name of x, y, fill and title
  xlab("Longitude")+ylab("Latitude")+
  ggtitle("Numbers of Expenses in Each Area(1998-2020))+labs(fill="Count")+
  theme(plot.title = element_text(hjust = 0.5))
```



## Discussion

- Based on the bar chart of number of hurricanes in each state, we can conclude that hurricanes happened more in Virginia, North Carolina, Texas, Mississippi, Louisiana, Alabama and Florida than other states from 1998 to 2020. Among these states, hurricanes happened 23 times in Florida which is the most.
- According to three maps, we can find that the more hurricanes happen, the more public assistance programs and expenses are.
- Based on the bar chart of different expenses in each damage category, we can conclude that public assistance programs' expenses are mainly used in public utilities, public buildings, protective measures and debris removal.

## References

1. Yihui Xie (2020). knitr: A General-Purpose Package for Dynamic Report Generation in R. R package version 1.29. <https://github.com/yihui/knitr>
2. Hadley Wickham (2019). tidyverse: Easily Install and Load the 'Tidyverse'. R package version 1.3.0. <https://cloud.r-project.org/package=tidyverse>
3. Greg Freedman Ellis, Thomas Lumley, Tomasz Żółtak and Ben Schneider (2020). srvyr: 'dplyr'-Like Syntax for Summary Statistics of SURvey Data. R package version 0.4.0. <https://cloud.r-project.org/package=srvyr>
4. Hadley Wickham (2020). reshape2: Flexibly Reshape Data: A Reboot of the Reshape Package. R package version 1.4.4. <https://github.com/hadley/reshape>
5. Stefan Milton Bache and Hadley Wickham (2014). magrittr: A Forward-Pipe Operator for R. R package version 1.5. <https://cran.rstudio.com/web/packages/magrittr/index.html>