MA615 Midterm Project - Flood Analysis

Fengyuan Shen (Vincent)

2023-11-06

Introduction

Floods are one of the most common and destructive natural disasters in the world. In the United States, floods not only threaten people's lives, but also have a significant impact on society and economy. Many parts of the world experienced severe flood events during 2020-2021, which highlighted the need for a deep understanding of flood impacts.

This report aims to examine the dangers of flooding, the economic costs, and what types of communities are more likely to suffer damage through exploratory data analysis of data provided by National Oceanic and Atmospheric Administration (NOAA) and the Federal Emergency Management Agency (FEMA).

Goal

This report aims to achieve the following research objectives through an in-depth analysis of flood data for 2020-2021:

1. Flood frequency analysis - Create time series maps of flood events that clearly show the frequency of floods to identify possible seasonal or long-term trends.
2. Flood Duration Assessment - Assess the duration of flood events and their range of variation to understand the potential impact of the persistence of different flood types on communities.
3. State-level Flood Event analysis - Create a bar chart of the number of flood events by state to compare the frequency and severity of flood events across states.
4. Geographic Distribution Research - The production of geographic maps of flood events to visualize patterns of flood occurrence and high-risk areas in different regions.
5. Flood Type and Distribution Exploration - Analyze the distribution of different types of flood events in each state to reveal the possible flood risk factors in a specific area.

6. Quantification of Economic losses - calculates and analyzes property and crop losses due to flooding in each state and assesses regional differences in economic impact.
7. Casualty Statistics and Factor analysis - Create a statistical map of flood-related deaths and explore the main factors leading to casualties to improve future warning systems and emergency response measures.
8. Flood Cause Distribution Studies - Analyze the cause distribution of flood occurrence in different regions and identify the dominant natural or human factors.
9. Community impact Model analysis - Develop interactive maps to explore the types of communities affected by flooding most.

This report aims to provide actionable insights through detailed data analysis to help organizations and individuals take targeted measures to reduce direct and indirect losses from flood disasters.

Data Acquisition and Assessment

This report uses data sets from the National Oceanic and Atmospheric Administration (NOAA) and the Federal Emergency Management Agency (FEMA), It includes records of flood events and other disasters across the United States from 2020 to 2021.

Datasets overview

1. Disaster declarations summary data (DisasterDeclarationsSummaries.csv) :

   This data contains the disaster number, type, date, and affected area, which can be used to determine the frequency and location of flood events. Data fields such as disaster type and date help analyze seasonal and long-term trends in flooding.

2. FEMA flood compensation overview data (FemaWebDisasterSummaries.csv) :

   Data were provided on the amount of approved personal aid and public aid funding, helping to quantify the economic damage caused by the floods. The loss data for each state can be used to compare losses across states.

3. Storm event details (StormEvents_details-ftp_v1.0_d2020_c20230927.csv, StormEvents_details-ftp_v1.0_d2021_c20231017.csv):

   Including specific storm event details, such as start and end dates, death tolls, and property damage, is critical to understanding the duration and impact of flooding. The data helps create a geographic map of storm events and identify areas at higher risk.

4. Storm event death record data (StormEvents_fatalities-ftp_v1.0_d2020_c20230927.csv, StormEvents_fatalities-ftp_v1.0_d2021_c20231017.csv):

   Detailed information on death events, including the type and location of death, is recorded, which is useful for analyzing deaths and related factors caused by floods. It can help create maps that reflect flood fatality statistics and explore the main factors that lead to casualties.

```r
disaster <- read.csv("data/DisasterDeclarationsSummaries.csv", header = T)
fin_assistance <- read.csv("data/FemaWebDisasterSummaries.csv", header = T)
storm_det_2020 <- read.csv("data/StormEvents_details-ftp_v1.0_d2020_c20230927.csv", header
storm_det_2021 <- read.csv("data/StormEvents_details-ftp_v1.0_d2021_c20231017.csv", header
storm_fat_2020 <- read.csv("data/StormEvents_fatalities-ftp_v1.0_d2020_c20230927.csv",
                           header = T)
storm_fat_2021 <- read.csv("data/StormEvents_fatalities-ftp_v1.0_d2021_c20231017.csv",
                           header = T)
storm_loc_2020 <- read.csv("data/StormEvents_locations-ftp_v1.0_d2020_c20230927.csv",
                           header = T)
storm_loc_2021 <- read.csv("data/StormEvents_locations-ftp_v1.0_d2021_c20231017.csv",
                           header = T)
```

Data Cleaning

1. **Keyword Filtering**:

   Defined a list of keywords related to floods and constructed a regular expression to identify flood-related events from the storm details data for the years `2020` and `2021`.

2. **Data Merging**:

   Merged `StormEvents_details-ftp_v1.0_d2020_c20230927.csv` and `data/StormEvents_details-ftp_` into a single data frame to consolidate the information for analysis.

3. **Time Data Cleaning**:

   The `clean_time` function formats time values into a consistent four-digit representation (HHMM). If a time value is invalid, it defaults to '0000', representing midnight.

4. **Datetime Creation and Localization**:

   Using the `create_datetime` function, datetime objects are created from date and time information, incorporating time zone details for accuracy.

   The `create_datetime_safe` function wraps `create_datetime` in error handling to ensure robust datetime creation. If any errors occur, it returns `NA` to avoid processing disruptions.

These functions are applied to the BEGIN_TIME and END_TIME columns in the flood_details dataframe to create BEGIN_DATETIME and END_DATETIME columns, respectively, with the correct datetime objects for each flood event.

5. **Damage Value Conversion**:

   Implemented a function to convert reported damage values from strings to numeric values, accounting for the use of “K” (thousands) and “M” (millions) in the data, and applied this function to the DAMAGE_PROPERTY and DAMAGE_CROPS columns.

6. **Column Selection and Cleaning**:

   Selected a specific set of columns to retain for the analysis, removing any that were not required, to streamline the data frame

7. **Data Filtering by Geographic Range**:

   Applied a geographic filter to retain data points that fall within the approximate latitude and longitude range of the United States.

8. **Event ID Extraction**:

   Extracted unique event IDs for flood events and used these to filter related fatality data for both 2020 and 2021.

9. **Combining Fatality Data**:

   Merged data/StormEvents_fatalities-ftp_v1.0_d2020_c20230927.csv and StormEvents_fatalities-ftp_v1.0_d2021_c20231017.csv into a combined data frame for comprehensive analysis of flood-related fatalities.

```r
# Define flood-related keywords
flood_related_keywords <- c('Flood', 'Coastal Flood', 'Flash Flood', 'Lakeshore Flood')

# Construct a regular expression, ignoring case
flood_pattern <- paste0('(?i)', paste(flood_related_keywords, collapse = "|"))

# Screening for flood-related events in detailed data for 2020
flood_det_2020 <- storm_det_2020 |>
  filter(str_detect(EVENT_TYPE, flood_pattern))

# Screening for flood-related events in detailed data for 2021
flood_det_2021 <- storm_det_2021 |>
  filter(str_detect(EVENT_TYPE, flood_pattern))

# Merge two years of detailed flood data
flood_details <- bind_rows(flood_det_2020, flood_det_2021)
```

```r
# Display the shape and header of the merged flood detail data to ensure correctness
dim(flood_details)
head(flood_details)
```

```r
# Function of cleaning time data
clean_time <- function(time_val) {
  # Convert to a string, making sure it's in a four-digit format
  time_str <- sprintf("%04s", as.integer(time_val))
  # Validation and formatting time
  tryCatch({
    time <- hm(time_str)
  }, error = function(e) {
    time_str <- '0000'  # Reset to midnight if invalid
  })
  return(time_str)
}

# Apply the cleaning function to the time column
flood_details$BEGIN_TIME <- sapply(flood_details$BEGIN_TIME, clean_time)
flood_details$END_TIME <- sapply(flood_details$END_TIME, clean_time)

# Function that creates a datetime object with a time zone
create_datetime <- function(yearmonth, day, time, tz_string) {
  # Create a datetime object from year, month, day and time
  year <- as.integer(substr(yearmonth, 1, 4))
  month <- as.integer(substr(yearmonth, 5, 6))
  hour <- as.integer(substr(time, 1, 2))
  minute <- as.integer(substr(time, 3, 4))
  dt <- make_datetime(year, month, day, hour, minute)

  # Parse time zone string
  tz_info <- strsplit(tz_string, "-")[[1]]
  tz_name <- tz_info[1]  # Time zone name
  tz_offset <- as.integer(tz_info[2])  # Time zone offset from UTC

  # Localizes the datetime object to the time zone
  dt <- with_tz(dt, tz(tz_offset * -60))

  # Convert the time zone to UTC
  dt <- with_tz(dt, 'UTC')
  return(dt)
}

# Security functions that create datetime objects and deal with potential errors
create_datetime_safe <- function(yearmonth, day, time, tz_string) {
  tryCatch({
    create_datetime(yearmonth, day, time, tz_string)
  }, error = function(e) {
    # Return NA or some default datetime on any error
    return(NA)
  })
}

# Apply the function to the start and end datetime columns, handling errors
flood_details <- flood_details |>
  rowwise() |>
  mutate(
    BEGIN_DATETIME = create_datetime_safe(BEGIN_YEARMONTH, BEGIN_DAY,
                                          BEGIN_TIME, CZ_TIMEZONE),
    END_DATETIME = create_datetime_safe(END_YEARMONTH, END_DAY,
                                        END_TIME, CZ_TIMEZONE)
    ) |>
  ungroup()

# Displays the new datetime column and the cleaned time column
head(select(flood_details, BEGIN_DATETIME, END_DATETIME))
```

```
# Define a function to convert damage values, taking into account thousands and millions
convert_damage_value <- function(damage) {
  if (is.na(damage) || damage == "0.00K") {
    return(0)
  } else if (grepl("K", damage)) {
    return(as.numeric(sub("K", "", damage)) * 1e3)
  } else if (grepl("M", damage)) {
    return(as.numeric(sub("M", "", damage)) * 1e6)
  } else {
    return(as.numeric(damage))
  }
}

# Apply the conversion function to the DAMAGE_CROPS and DAMAGE_PROPERTY column
```

```
flood_details <- flood_details |>
  mutate(DAMAGE_CROPS = sapply(DAMAGE_CROPS, convert_damage_value)) |>
  mutate(DAMAGE_PROPERTY = sapply(DAMAGE_PROPERTY, convert_damage_value))
```

```
# Define the columns to keep
columns_to_keep <- c('BEGIN_DATETIME', 'END_DATETIME',
                     'BEGIN_YEARMONTH', 'BEGIN_DAY', 'BEGIN_TIME',
                     'END_YEARMONTH', 'END_DAY', 'END_TIME',
                     'STATE', 'EVENT_TYPE', 'CZ_TYPE', 'CZ_NAME',
                     'INJURIES_DIRECT', 'INJURIES_INDIRECT',
                     'DEATHS_DIRECT', 'DEATHS_INDIRECT',
                     'DAMAGE_PROPERTY', 'DAMAGE_CROPS', 'FLOOD_CAUSE',
                     'BEGIN_LAT', 'BEGIN_LON', 'END_LAT', 'END_LON',
                     'EVENT_ID')

# Remove unwanted columns
flood_details <- select(flood_details, all_of(columns_to_keep))

# Removes records with missing latitude or longitude data
#flood_details <- drop_na(flood_details, c('BEGIN_LAT', 'BEGIN_LON', 'END_LAT', 'END_LON')
```

```
# Extract the ID of the flood event
flood_event_ids <- unique(flood_details$EVENT_ID)

# The extracted event ids were used to filter out data on flood-related deaths in 2020 and
flood_fatalities_2020 <- filter(storm_fat_2020, EVENT_ID %in% flood_event_ids)
flood_fatalities_2021 <- filter(storm_fat_2021, EVENT_ID %in% flood_event_ids)

# Merge two years of death data
flood_fatalities_combined <- bind_rows(flood_fatalities_2020, flood_fatalities_2021)
```

```
# The approximate latitude and longitude range of the United States is from about 24 to 49
# Sift out the data in this range
flood_details_us <- flood_details |>
  filter(BEGIN_LAT >= 24, BEGIN_LAT <= 49,
         BEGIN_LON >= -125, BEGIN_LON <= -66,
         END_LAT >= 24, END_LAT <= 49,
         END_LON >= -125, END_LON <= -66,
         )
```

Below shows the cleaned data.

```
# Show the cleaned data
head(flood_details)
```

```
# A tibble: 6 x 24
  BEGIN_DATETIME      END_DATETIME        BEGIN_YEARMONTH BEGIN_DAY BEGIN_TIME
  <dttm>              <dttm>                        <int>     <int> <chr>
1 2020-05-25 17:00:00 2020-05-25 20:00:00          202005        25 1700
2 2020-04-13 00:30:00 2020-04-13 03:30:00          202004        13 0030
3 2020-03-28 18:30:00 2020-03-28 19:30:00          202003        28 1830
4 2020-03-28 14:30:00 2020-03-28 15:30:00          202003        28 1430
5 2020-03-03 05:30:00 2020-03-03 06:30:00          202003         3 0530
6 2020-04-13 19:00:00 2020-04-14 05:45:00          202004        13 1900
# i 19 more variables: END_YEARMONTH <int>, END_DAY <int>, END_TIME <chr>,
#   STATE <chr>, EVENT_TYPE <chr>, CZ_TYPE <chr>, CZ_NAME <chr>,
#   INJURIES_DIRECT <int>, INJURIES_INDIRECT <int>, DEATHS_DIRECT <int>,
#   DEATHS_INDIRECT <int>, DAMAGE_PROPERTY <dbl>, DAMAGE_CROPS <dbl>,
#   FLOOD_CAUSE <chr>, BEGIN_LAT <dbl>, BEGIN_LON <dbl>, END_LAT <dbl>,
#   END_LON <dbl>, EVENT_ID <int>
```

```
glimpse(flood_details)
```

```
Rows: 13,661
Columns: 24
$ BEGIN_DATETIME    <dttm> 2020-05-25 17:00:00, 2020-04-13 00:30:00, 2020-03-2~
$ END_DATETIME      <dttm> 2020-05-25 20:00:00, 2020-04-13 03:30:00, 2020-03-2~
$ BEGIN_YEARMONTH   <int> 202005, 202004, 202003, 202003, 202003, 202004, 2020~
$ BEGIN_DAY         <int> 25, 13, 28, 28, 3, 13, 19, 24, 13, 13, 14, 16, 18, 2~
$ BEGIN_TIME        <chr> "1700", "0030", "1830", "1430", "0530", "1900", "091~
$ END_YEARMONTH     <int> 202005, 202004, 202003, 202003, 202003, 202004, 2020~
$ END_DAY           <int> 25, 13, 28, 28, 3, 14, 19, 24, 13, 13, 14, 16, 18, 2~
$ END_TIME          <chr> "2000", "0330", "1930", "1530", "0630", "0545", "180~
$ STATE             <chr> "WEST VIRGINIA", "VIRGINIA", "OHIO", "OHIO", "OHIO",~
$ EVENT_TYPE        <chr> "Flash Flood", "Flood", "Flash Flood", "Flood", "Flo~
$ CZ_TYPE           <chr> "C", "C", "C", "C", "C", "C", "Z", "C", "C", "C", "C~
$ CZ_NAME           <chr> "WEBSTER", "DICKENSON", "DARKE", "LICKING", "MONTGOM~
$ INJURIES_DIRECT   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
$ INJURIES_INDIRECT <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
$ DEATHS_DIRECT     <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
$ DEATHS_INDIRECT   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
$ DAMAGE_PROPERTY   <dbl> 3000, 5000, 0, 0, 0, 2000, 0, 0, 0, 0, 0, 0, 0, 0, 0~
$ DAMAGE_CROPS     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
$ FLOOD_CAUSE      <chr> "Heavy Rain", "Heavy Rain", "Heavy Rain", "Heavy Rai~
$ BEGIN_LAT        <dbl> 38.5367, 37.1988, 40.1000, 39.9700, 39.7500, 39.6391~
$ BEGIN_LON        <dbl> -80.5887, -82.5269, -84.6200, -82.6200, -84.2200, -8~
$ END_LAT          <dbl> 38.5186, 37.2685, 40.0942, 39.9690, 39.7472, 39.6392~
$ END_LON          <dbl> -80.5378, -82.3034, -84.6748, -82.6201, -84.2234, -8~
$ EVENT_ID         <int> 885808, 880416, 874556, 874465, 871084, 883021, 8695~
```

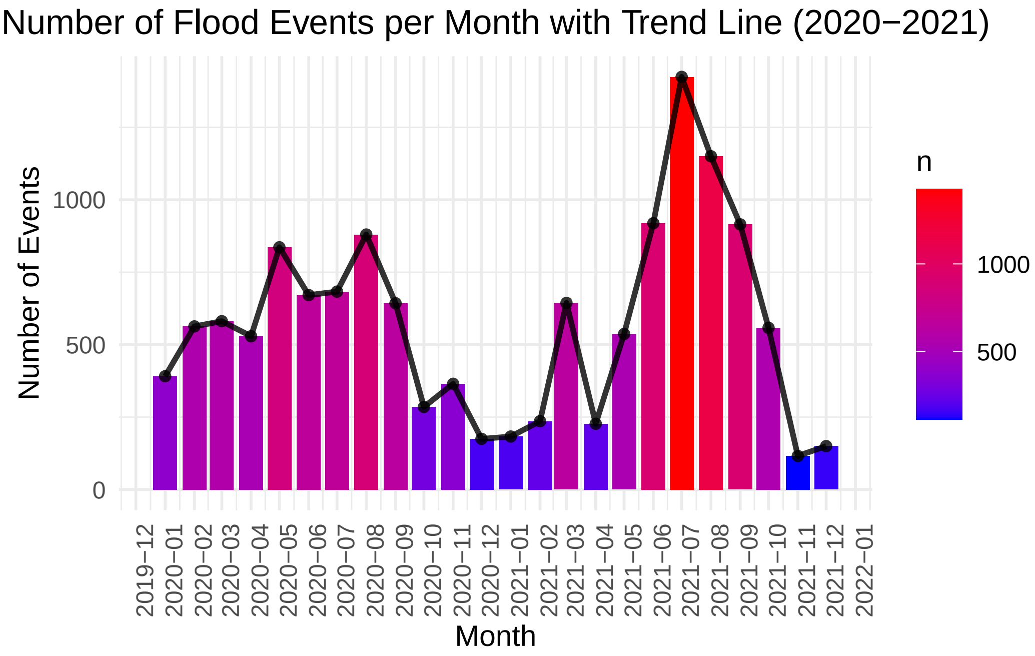
Exploratory Data Analysis (EDA)

After data-cleaning process, a cleaned data set is ready for use.

This section will be divided into 7 parts so as to analyze flood impacts (such as dangers of flooding, the economic costs, etc.) from different aspects.

1. Flood frequency analysis

```r
# Create a new column that combines year and month for plotting
flood_details <- flood_details |>
  mutate(YEAR_MONTH = floor_date(BEGIN_DATETIME, "month")) |>
  mutate(YEAR_MONTH = as.Date(YEAR_MONTH))  # Convert to Date object

# Count the number of flood events for each month
monthly_counts <- flood_details |>
  count(YEAR_MONTH) |>
  arrange(YEAR_MONTH)

# Plot the distribution of flood events over months for 2020 and 2021
ggplot(monthly_counts, aes(x = YEAR_MONTH, y = n, fill = n)) +
  geom_bar(stat = "identity", alpha = 1) +
  geom_line(aes(group = 1), color = "black", linewidth = 1, alpha = 0.8) +
  geom_point(color = "black", alpha = 0.8) +
  scale_x_date(labels = date_format("%Y-%m"), breaks = date_breaks("1 month")) +
  scale_fill_gradient(low = "blue", high = "red") + # Define gradient colors here
  labs(title = 'Number of Flood Events per Month with Trend Line (2020-2021)',
       x = 'Month',
       y = 'Number of Events') +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5),
    axis.text.x = element_text(angle = 90, hjust = 1)
    )
```



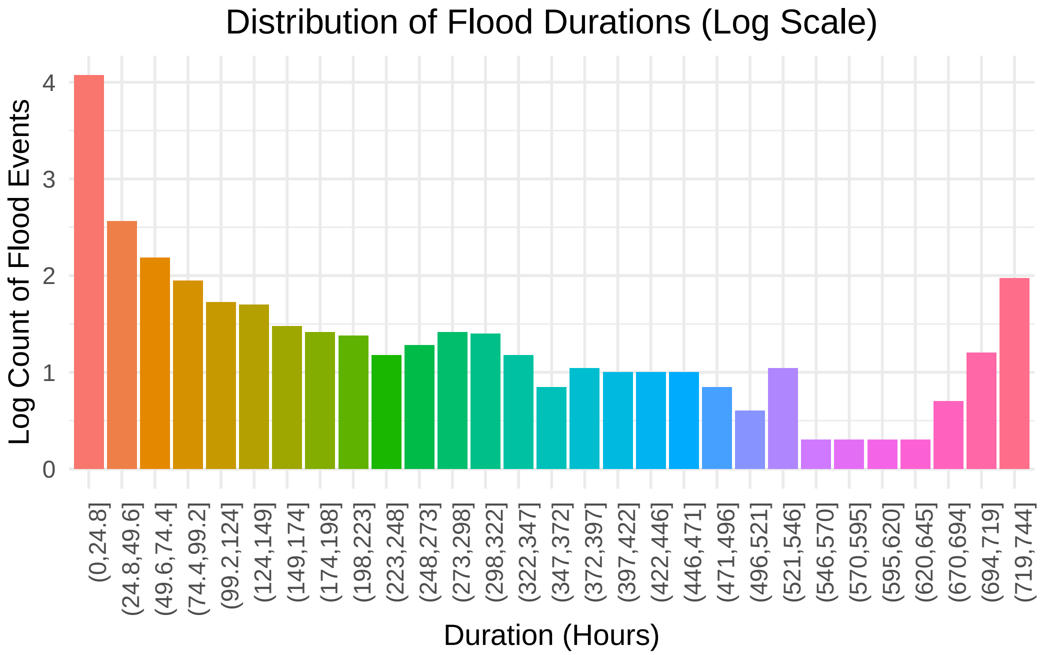
This graph plots the number of flood events in the United States for each month in 2020-2021, with darker columns indicating more floods occurring and the black lines being trend lines.

On average, 534 flood events occur each month with a standard deviation of 341.85, with the highest month having 1420 flood events.

It is obvious from the figure that summer (May to September) is the most frequent period of flood events, while the number of floods occurs less in winter. This may indicate a seasonal trend or a concentration of extreme weather events over a specific time period.

```
# Convert BEGIN_DATETIME and END_DATETIME to datetime objects
flood_details <- flood_details |>
  mutate(flood_details,BEGIN_DATETIME = as.POSIXct(BEGIN_DATETIME,
                                                   format = "%Y-%m-%d %H:%M:%S"),
                        END_DATETIME = as.POSIXct(END_DATETIME,
                                                  format = "%Y-%m-%d %H:%M:%S"))

# Calculate the duration for each flood event in hours
flood_details <- mutate(flood_details,
                        DURATION = as.numeric(difftime(END_DATETIME, BEGIN_DATETIME,
```

```r
                                                       units = "hours")))

# calculate bins
flood_details$bins <- cut(flood_details$DURATION, breaks = seq(min(flood_details$DURATION)
                                                               max(flood_details$DURATION)
                                                               length.out = 31))

# Calculate the count for each bin and perform a logarithmic transformation
duration_count <- flood_details |>
  drop_na(bins) |>
  group_by(bins) |>
  summarise(count = n()) |>
  mutate(log_count = ifelse(count == 0, 0, log10(count)))

# Plot the count after the logarithmic transformation
ggplot(duration_count, aes(x = bins, y = log_count, fill = bins)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  labs(title = 'Distribution of Flood Durations (Log Scale)',
       x = 'Duration (Hours)',
       y = 'Log Count of Flood Events') +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5),
    axis.text.x = element_text(angle = 90, hjust = 1)
    )
```

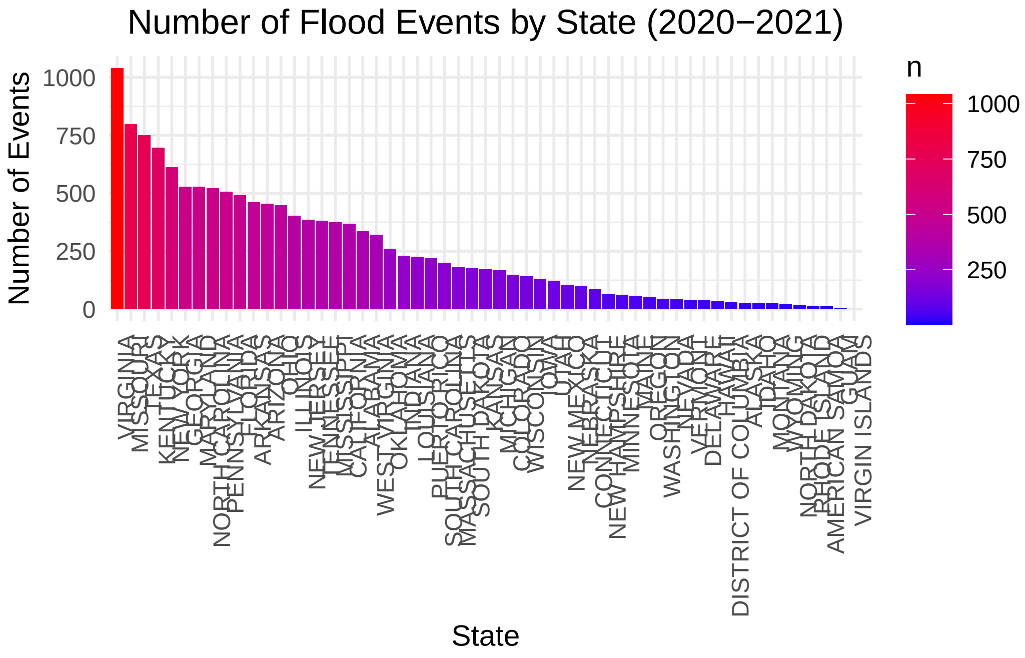


This graph describes the distributions of flood duration, and the ordinate is the logarithmic count.

The duration of flood events varies widely, from a few hours to a few days. Most flood events last no more than a day, but in extreme cases, some flood events can last more than a month, reflecting the potential lasting impact of different flood types on communities.

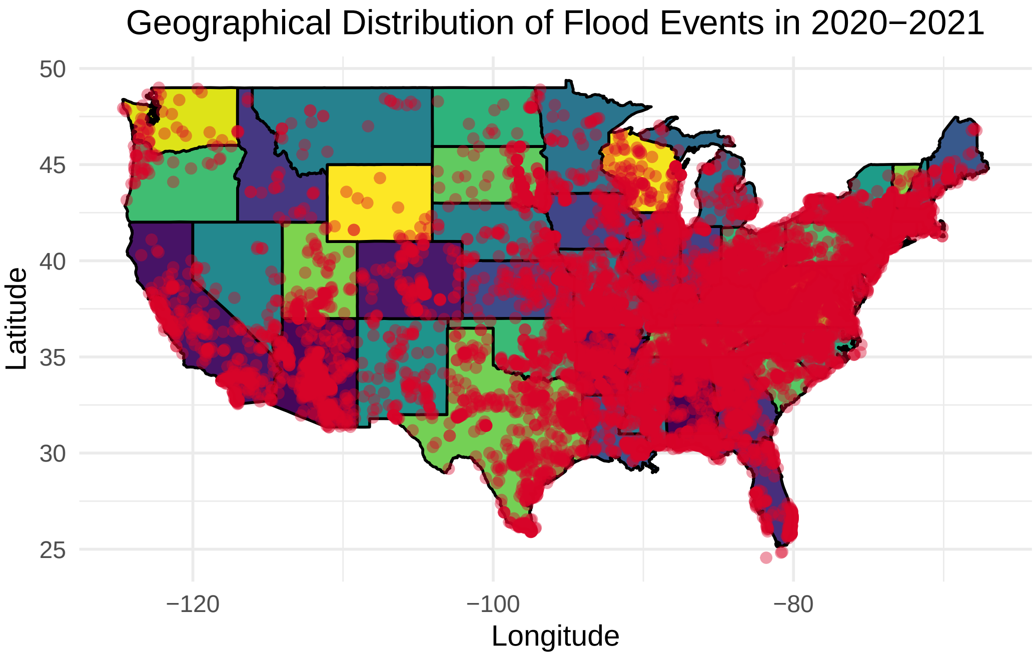
2. Geographic distribution of flood events

```r
# Count the number of flood events by state
state_counts <- flood_details |>
  count(STATE, sort = TRUE) # Sort = TRUE will order the states by the number of events

# Plot the distribution of flood events by state
ggplot(state_counts, aes(x = reorder(STATE, -n), y = n)) +
  geom_bar(stat = "identity", aes(fill = n)) +
  scale_fill_gradient(low = "blue", high = "red") + # Use gradient color for the bars
  labs(title='Number of Flood Events by State (2020-2021)',
       x = 'State', y = 'Number of Events') +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5),
    axis.text.x = element_text(angle = 90, vjust = 1, hjust=1),
    )
```



This graph shows the number of flood events by state. From the graph, Virginia, Missouri and Texas are the top three states with the highest number of flood events. This may be related to the climatic characteristics, topography and hydrological conditions of these areas.

```
# Create a data box for a map of the United States
states_map <- map_data("state")

# Draw the map
ggplot() +
  geom_polygon(data = states_map, aes(x = long, y = lat, group = group, fill=factor(group)
               color="black") +
  scale_fill_viridis_d(guide = FALSE) +
  geom_point(data = flood_details_us, aes(x = BEGIN_LON, y = BEGIN_LAT),
             color="#D90429", alpha=0.4) +
  labs(title = 'Geographical Distribution of Flood Events in 2020-2021',
       x = 'Longitude', y = 'Latitude') +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```



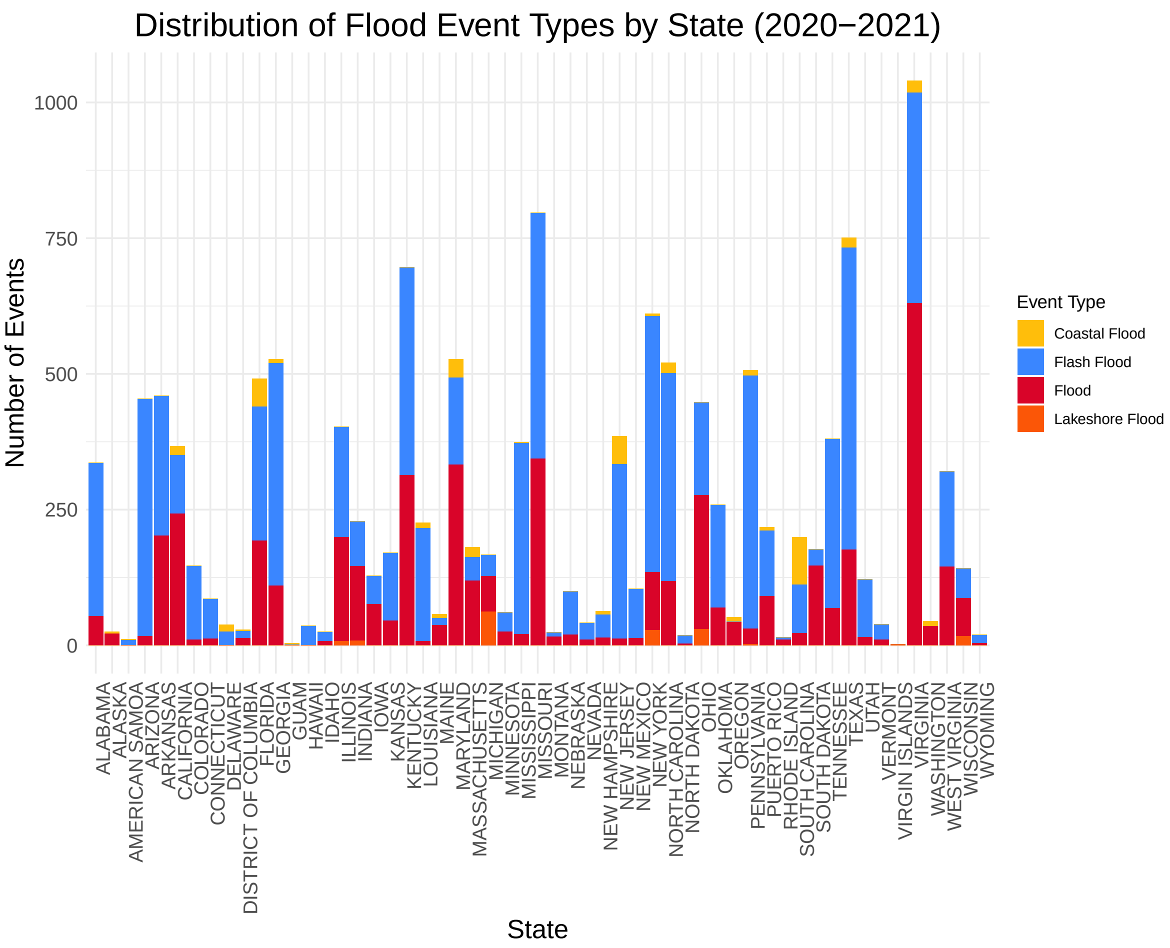
Based on the latitude and longitude coordinates of the data set, we can project the location of the flood to a map of the United States, with a red dot representing a flood event.

As we can see from the chart, the longitude of the flood event ranges from approximately 124.64°W to 67.42°W, with an average longitude of about 87.96°W. This reflects the greater concentration of flood events in the eastern and central United States.

3. Exploration of flood types and distribution

```
# Count the number of each event type in each state
event_type_distribution <- flood_details |>
  count(STATE, EVENT_TYPE) |>
  pivot_wider(names_from = EVENT_TYPE, values_from = n, values_fill = list(n = 0))

# Converts a data frame from a wide format back to a long format for drawing
event_type_distribution_long <- event_type_distribution |>
  pivot_longer(cols = -STATE, names_to = "EVENT_TYPE", values_to = "Number_of_Events")

# Dinfine color
```

```
custom_colors <- c("Flood" = '#D90429', "Flash Flood" = '#3A86FF',
                   "Coastal Flood" = '#FFBE0B', "Lakeshore Flood" = '#FB5607')

# Draw a stacked bar chart
ggplot(event_type_distribution_long, aes(x = STATE, y = Number_of_Events,
                                         fill = EVENT_TYPE)) +
  geom_bar(stat = "identity", position = "stack") +
  scale_fill_manual(values = custom_colors) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 20),
    axis.title.x = element_text(size = 16),
    axis.title.y = element_text(size = 16),
    axis.text.x = element_text(size = 12, angle = 90, vjust = 1, hjust = 1),
    axis.text.y = element_text(size = 12),
    ) +
  labs(x = 'State', y = 'Number of Events', fill = 'Event Type',
       title = 'Distribution of Flood Event Types by State (2020-2021)')
```



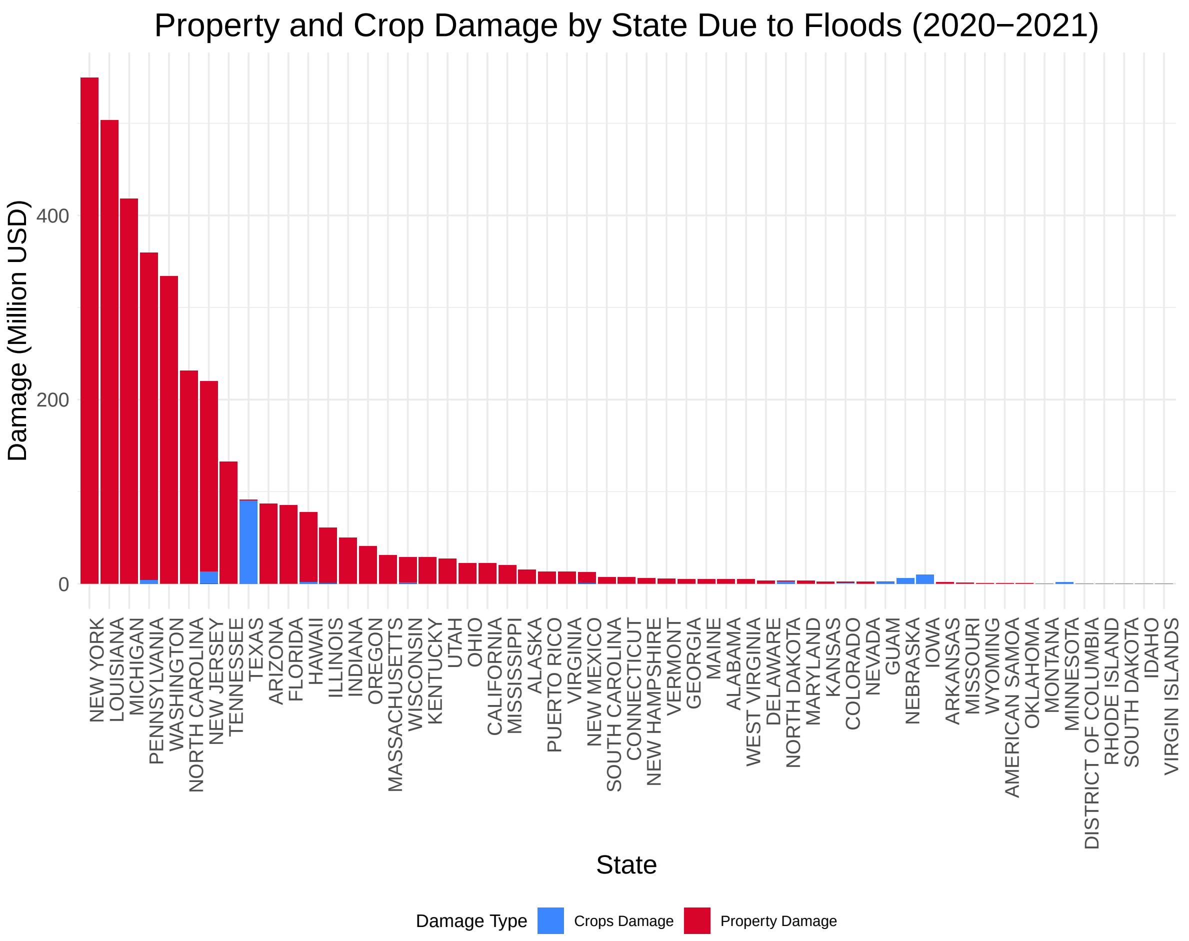
This chart explores how different types of flooding are distributed across U.S. states.

Different types of flood events are unevenly distributed across states. Some states may experience specific types of flooding more frequently, which may be due to state-specific environmental factors and climate conditions.

4. Property and crop damage and loss of life due to flood events

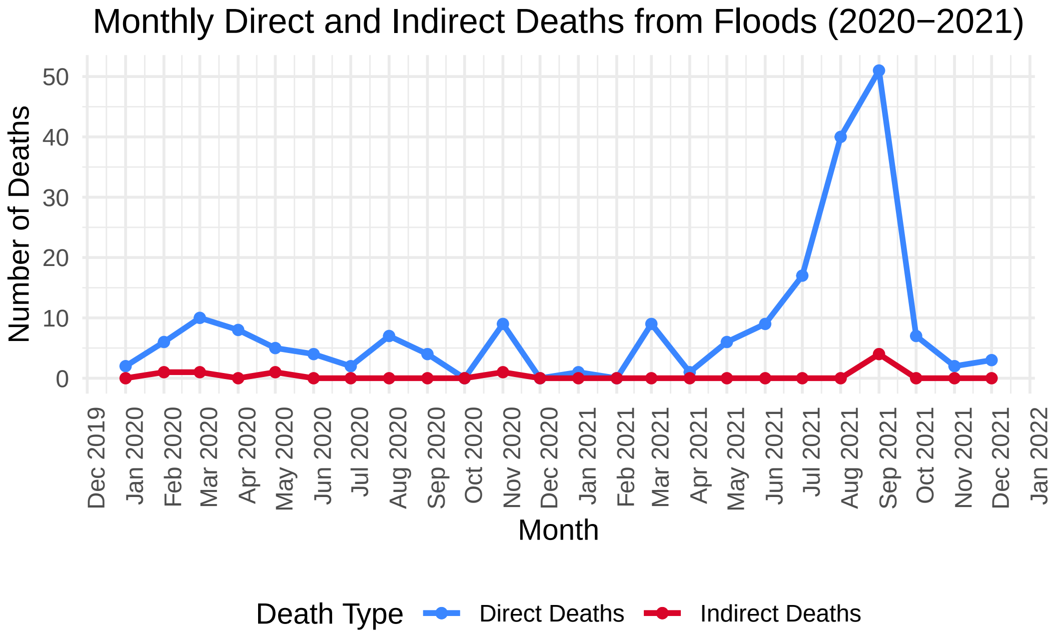
```r
# Use dplyr to group and sum data by state
state_damage <- flood_details |>
  group_by(STATE) |>
  summarise(DAMAGE_PROPERTY = sum(DAMAGE_PROPERTY, na.rm = TRUE),
            DAMAGE_CROPS = sum(DAMAGE_CROPS, na.rm = TRUE)) |>
  ungroup()
```

```r
# Sort data according to DAMAGE_PROPERTY for better visualization
state_damage_sorted <- state_damage |>
  arrange(desc(DAMAGE_PROPERTY))

# draw the plot
ggplot(state_damage_sorted, aes(x=reorder(STATE, -DAMAGE_PROPERTY))) +
  geom_bar(aes(y=DAMAGE_PROPERTY/10^6, fill="Property Damage"),
           stat="identity", position='dodge') +
  geom_bar(aes(y=DAMAGE_CROPS/10^6, fill="Crops Damage"),
           stat="identity", position='dodge') +
  labs(title='Property and Crop Damage by State Due to Floods (2020-2021)',
       x='State', y='Damage (Million USD)', fill="Damage Type") +
  scale_y_continuous(labels=scales::comma) +
  scale_fill_manual(values=c("Property Damage"="#D90429", "Crops Damage"="#3A86FF")) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 20),
    axis.title.x = element_text(size = 16),
    axis.title.y = element_text(size = 16),
    axis.text.x = element_text(size = 12, angle = 90, vjust = 1, hjust = 1),
    axis.text.y = element_text(size = 12),
    legend.position = "bottom"
  )
```



As the chart shows, the floods caused significant property damage in cities such as New York, Louisiana, and Michigan, and also had a great impact on agricultural production in Texas. Total property damage was about $3.385 billion and crop damage was more than $131 million.

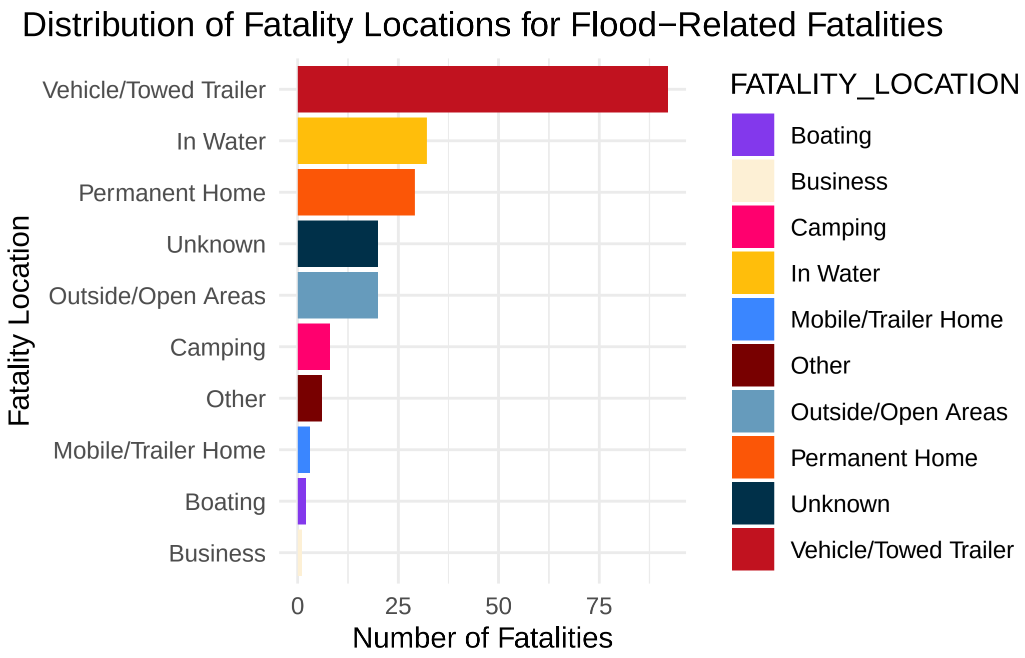
```
# The total number of direct and indirect deaths is grouped by month and year
monthly_deaths <- flood_details |>
  group_by(MONTH = floor_date(YEAR_MONTH, "month")) |>
  summarise(DEATHS_DIRECT = sum(DEATHS_DIRECT, na.rm = TRUE),
            DEATHS_INDIRECT = sum(DEATHS_INDIRECT, na.rm = TRUE))

# Draw the plot
ggplot(monthly_deaths, aes(x = MONTH)) +
  geom_line(aes(y = DEATHS_DIRECT, group = 1, colour = "Direct Deaths"), size = 1) +
  geom_point(aes(y = DEATHS_DIRECT, colour = "Direct Deaths")) +
  geom_line(aes(y = DEATHS_INDIRECT, group = 1, colour = "Indirect Deaths"), size = 1) +
  geom_point(aes(y = DEATHS_INDIRECT, colour = "Indirect Deaths")) +
  scale_colour_manual(values = c("Direct Deaths" = "#3A86FF", "Indirect Deaths" = "#D90429
  labs(title = 'Monthly Direct and Indirect Deaths from Floods (2020-2021)',
       x = 'Month', y = 'Number of Deaths', colour = "Death Type") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 90, vjust = 1, hjust = 1),
    plot.title = element_text(hjust = 0.5),
    legend.position = "bottom"
    ) +
  scale_x_date(date_breaks = "1 month", date_labels = "%b %Y")
```



This graph shows the number of human deaths each month caused directly or indirectly by floods. More than 200 deaths were attributed to the flooding events, with the death toll peaking during the summer of 2021.

5. Analysis of death factor

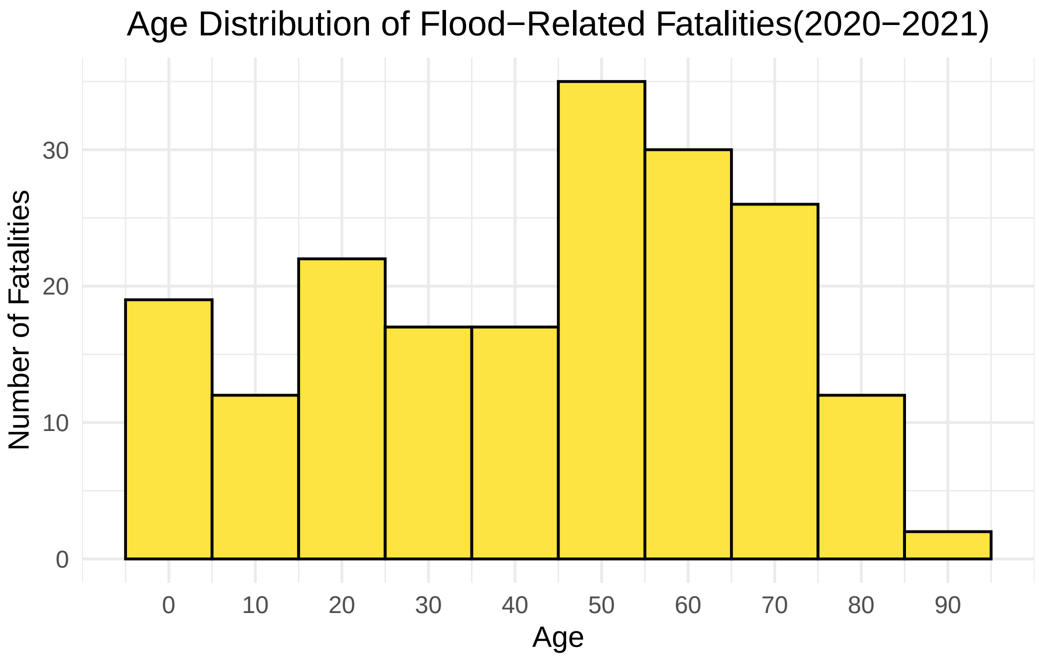
```
# The location distribution of flood-related deaths was analyzed
location_distribution <- table(flood_fatalities_combined$FATALITY_LOCATION)

# Convert table objects to data boxes for use in ggplot
location_distribution_df <- as.data.frame(location_distribution)

# Name the data frame column
names(location_distribution_df) <- c('FATALITY_LOCATION', 'Number_of_Fatalities')

# Draw the plot
ggplot(location_distribution_df, aes(x=reorder(FATALITY_LOCATION, Number_of_Fatalities),
                                     y = Number_of_Fatalities, fill=FATALITY_LOCATION)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values=c('#8338EC','#FDF0D5','#FF006E','#FFBE0B','#3A86FF',
                             '#780000','#669BBC','#FB5607','#003049','#C1121F')) +
  theme_minimal() +
  labs(title = 'Distribution of Fatality Locations for Flood-Related Fatalities',
       x = 'Fatality Location', y = 'Number of Fatalities') +
  theme(plot.title = element_text(hjust = 0.5)) +
  coord_flip()
```



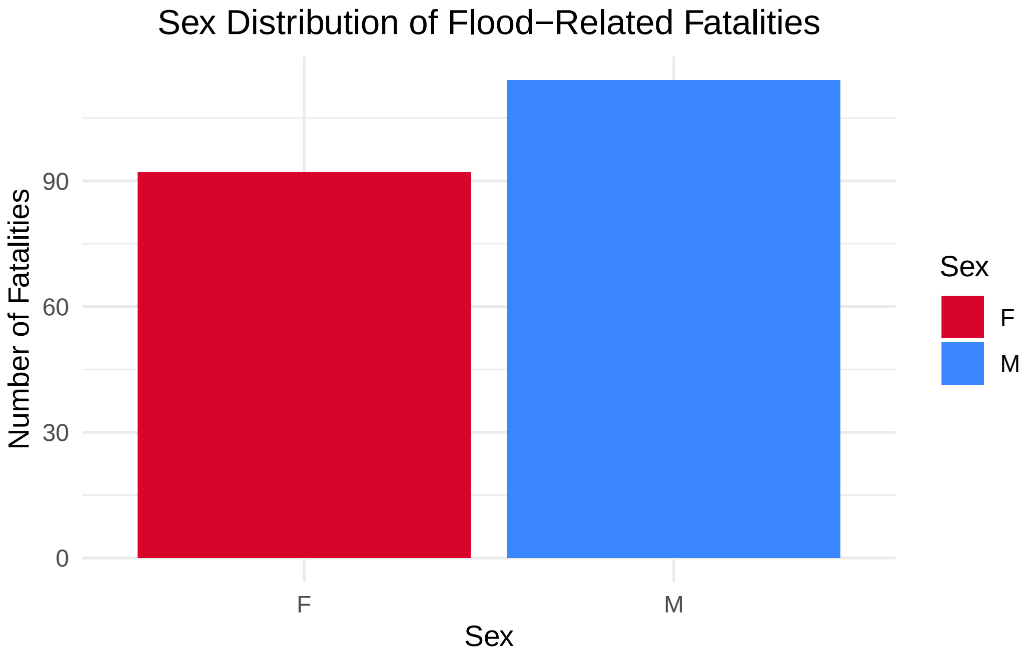
The distribution of flood-related deaths shows that “Vehicle/Towed Trailer” was the most common place of death or injury, followed by “In Water” and “Permanent Home”. We can also see that there are a considerable number of casualties in open areas.

Next, we will analyze statistics related to age and gender, which may help us further understand which populations are most vulnerable during flood events.

```r
# The age distribution of the deceased was analyzed
age_distribution <- na.omit(flood_fatalities_combined$FATALITY_AGE)
age_distribution <- as.integer(age_distribution)

# Plot the age distribution histogram
ggplot(data.frame(Age = age_distribution), aes(x = Age)) +
  geom_histogram(binwidth = 10, fill = "#FEE440", color = "black") +
  scale_fill_viridis_d() +
  labs(title = 'Age Distribution of Flood-Related Fatalities(2020-2021)',
       x = 'Age',
       y = 'Number of Fatalities') +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_x_continuous(breaks = seq(0, 90, by = 10))
```



```
# Keep records of known gender
known_sex <- flood_fatalities_combined |>
  filter(FATALITY_SEX %in% c('M', 'F'))

# The distribution of deaths by known sex was analyzed
sex_distribution <- table(known_sex$FATALITY_SEX)

# Draw a sex bar chart
ggplot(data.frame(Sex = names(sex_distribution), Count = as.integer(sex_distribution)),
       aes(x = Sex, y = Count, fill = Sex)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("M" = "#3A86FF", "F" = "#D90429")) +
  labs(title = 'Sex Distribution of Flood-Related Fatalities',
       x = 'Sex',
       y = 'Number of Fatalities') +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```



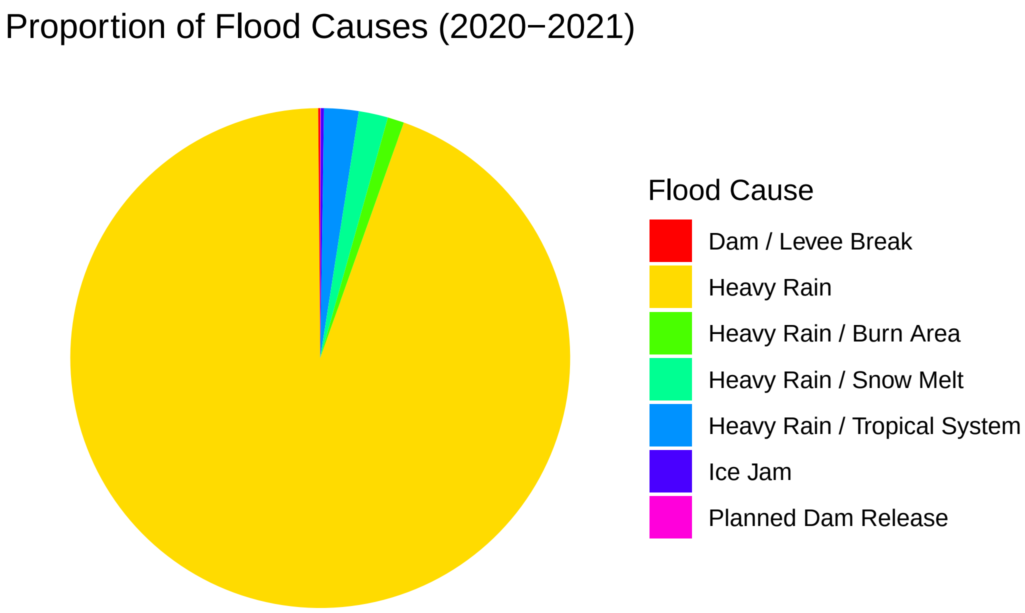
The distribution of flood-related age of death shows that: the average age was about 44 years. Deaths occur at all ages, but are particularly common among middle-aged and elderly people. The oldest victim was 86 years old, while the youngest was 1 year old.

Gender distribution shows: the male death toll was 114. The number of female deaths was 92. From this, we can see that slightly more men than women died in flood-related deaths.

Combining statistics on place of death, age and sex, it can be inferred that middle-aged and older men and people in vehicles or in water are at higher risk of injury or death during flood events.

6. Analysis of flood cause

```r
# Analyze the flood cause in the 'FLOOD_CAUSE' column
flood_causes <- flood_details |>
  filter(FLOOD_CAUSE != "", !is.na(FLOOD_CAUSE)) |>
  count(FLOOD_CAUSE) |>
  arrange(desc(n))

# Create a pie chart of flood causes
ggplot(flood_causes, aes(x = "", y = n, fill = FLOOD_CAUSE)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar(theta = "y") +
  scale_fill_manual(values = rainbow(nrow(flood_causes))) + # Use rainbow colors
  labs(title = 'Proportion of Flood Causes (2020-2021)',
       x = "", y = "", fill = "Flood Cause") +
  theme_void() +
  theme(legend.position = "right") +
  theme(plot.title = element_text(hjust = 0.5))
```



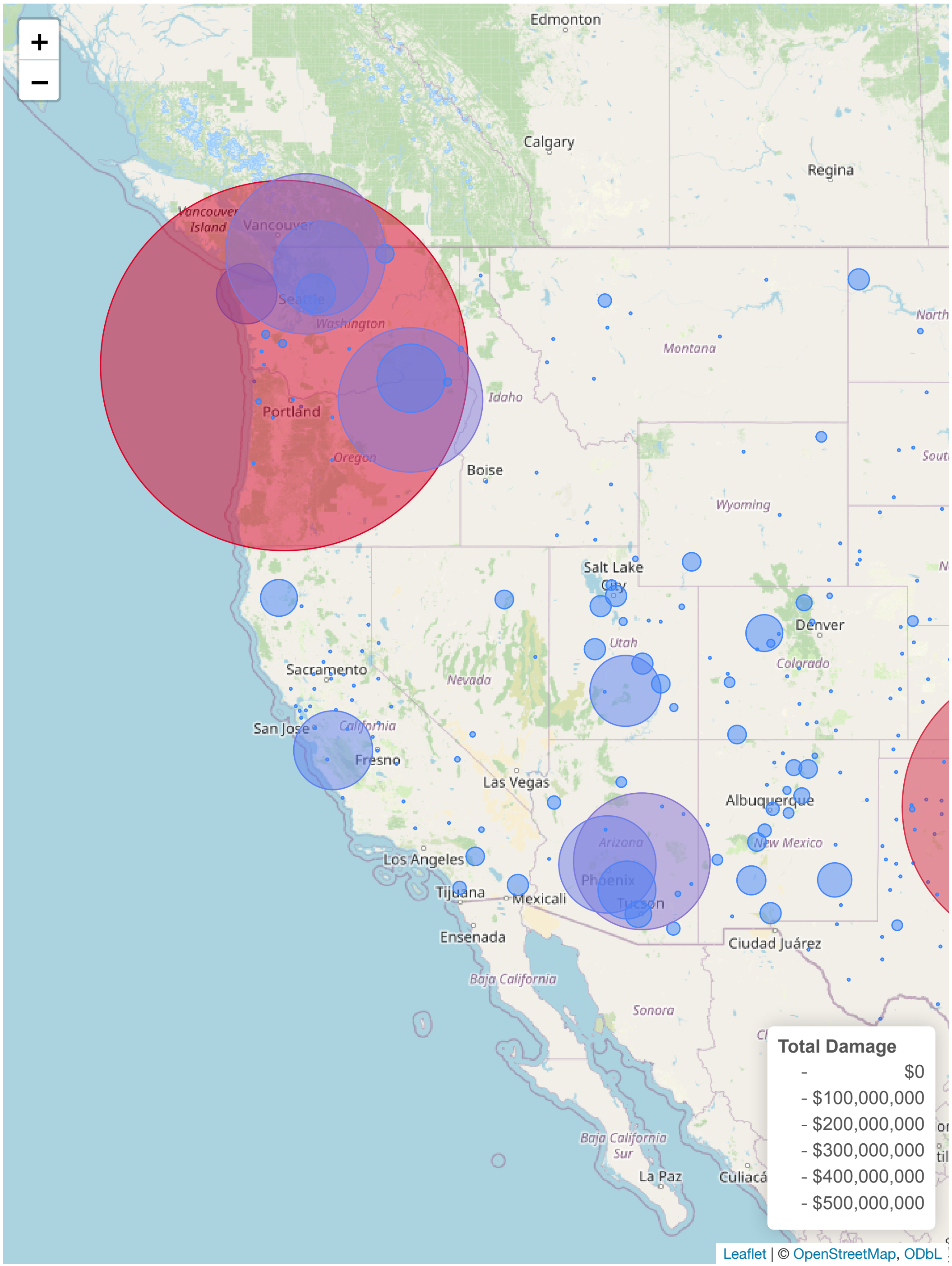
The analysis results of flood causes are as follows:

- Heavy Rain: 12373 times
- Heavy Rain/Tropical System: 293 times
- Heavy Rain/Snow Melt: 248 times
- Heavy Rain/Burn Area: 139 times
- Ice Jam: 27 times
- Dam/Levee Break: 17 times
- Planned Dam Release: 4 times

The most common cause of flooding is heavy rain, followed by heavy rain combined with tropical systems.

7. Community impact analysis

```r
# Aggregate the total property damage by county or NWS Public Forecast Zone
damage_data <- flood_details_us |>
  group_by(CZ_NAME) |>
  summarize(Total_Damage = sum(DAMAGE_PROPERTY, na.rm = TRUE),
            Avg_Lat = mean(BEGIN_LAT, na.rm = TRUE),
            Avg_Lon = mean(BEGIN_LON, na.rm = TRUE)) |>
  ungroup()

# Create a color palette for the damage scale
pal <- colorNumeric(palette = c("#3A86FF", "#D90429","#FFBE0B"),
                    domain = damage_data$Total_Damage)

# Create the leaflet map
map <- leaflet(damage_data) |>
  addTiles() |>   # Add default OpenStreetMap map tiles
  addCircles(
    lng = ~Avg_Lon, lat = ~Avg_Lat,
    weight = 1, opacity = 1,
    color = ~pal(Total_Damage),
    radius = ~sqrt(Total_Damage) * 30,  # Adjust this value for better visualization
    popup = ~paste(CZ_NAME, ": $", Total_Damage),
    fillOpacity = 0.5
  ) |>
  addLegend("bottomright", pal = pal, values = ~Total_Damage,
            title = "Total Damage",
            labFormat = labelFormat(prefix = "$"))

# Print the map
map
```


+
−
Edmonton
Calgary
Regina
Vancouver Island
Vancouver
Seattle
Washington
North
Montana
Idaho
Portland
Oregon
Boise
Sout
Wyoming
N
Salt Lake City
Denver
Utah
Colorado
Sacramento
Nevada
San Jose
California
Fresno
Las Vegas
Albuquerque
Arizona
New Mexico
Phoenix
Los Angeles
Tucson
Tijuana
Mexicali
Ensenada
Ciudad Juárez
Baja California
Sonora
Baja California Sur
La Paz
Culiacá
Total Damage
- $0
- $100,000,000
- $200,000,000
- $300,000,000
- $400,000,000
- $500,000,000
Leaflet | © OpenStreetMap, ODbL

This interactive map shows the economic damage caused by flood events in various U.S. counties. The larger the circle, the greater the economic loss.

It is clear from the graph that the eastern, southern and central counties have suffered huge economic losses caused by floods and these areas need more financial support to withstand flood disasters.

Conclusion

Based on the multi-angle analysis of flood data, we can draw the following comprehensive conclusions:

1. **Flood frequency and seasonality:**

   Flood events are not evenly distributed throughout the year, showing a clear seasonal pattern. This suggests that in some seasons, such as summer, flood events are more frequent.

2. **Flood duration:**

   The duration of flood events varies greatly, from a few hours to a few days, but most events end within a day. This means that although some floods recede quickly, significant impacts and damage are still possible.

3. **Geographical distribution:**

   Flood events are widely distributed throughout the United States, but are particularly concentrated in the eastern and central regions. This may be related to more frequent rainfall events and specific topographic conditions in those areas.

4. **State-level analysis:**

   Flooding events are unevenly distributed across states, with some states such as Virginia, Missouri, and Texas experiencing particularly frequent flooding events. This may reflect regional variations in climate and topographical conditions.

5. **Economic loss:**

   The floods have caused huge economic damage to property and crops, almost running into billions of dollars. This highlights the importance of implementing effective flood management measures and providing adequate financial support.

6. **Casualties:**

   The number of direct and indirect deaths caused by flooding shows that flooding is a serious public safety problem. This data can help policymakers prioritize where to implement more effective early warning systems and emergency response measures.

7. **Types and causes of floods:**

   Different types of flood events occur at different frequencies in different areas. Heavy rain is the main cause of flooding, but other factors such as heavy rain/tropical system and heavy rain/snow melttropical storms are also important.

8. **Community impact:**

   By analyzing the economic losses and casualties, we can infer that the floods have had a profound impact on some communities, especially in areas that are poor or lack adequate resources for effective flood protection. As a result, these areas are in greater need of federal funding.

Reference

1. NOAA Storm Events Database
2. Disaster Declarations Summaries - v2
3. FEMA Web Disaster Summaries - v1
4. FEMA Datasets
5. FEMA Flood Maps
6. Storm Data Bulk Data Format
7. NOAA Help
8. rfema