

# DC Witness: Mass Shootings Since 2014

*August 20, 2021*

**by Mark Lu**

—

Hey, Trina! I'm going to start attaching these markdown PDFs to research-related things, since it's just good for you (or the programmers) to know exactly what I'm doing when I'm pitching or sending updates about things. I know you're interested in putting interactive maps on the website, so these descriptions might help.

The purpose of this project is to map out every mass shooting in DC since 2014.

Since the data I am using comes from the Gun Violence Archive (GVA) project, we are operating on GAV's definition of a mass shooting: a shooting involving at least four fatal or nonfatal casualties in a public or private area.

Language: R

```
# Importing 2014-2021 GVA dataframes on mass shootings
```

```
incidents_2014_GVA <-  
read_csv("C:/Users/mark3/Desktop/dc_witness/mass_shootings/incidents-2014-  
GVA.csv")  
incidents_2015_GVA <-  
read_csv("C:/Users/mark3/Desktop/dc_witness/mass_shootings/incidents-2015-  
GVA.csv")  
incidents_2016_GVA <-  
read_csv("C:/Users/mark3/Desktop/dc_witness/mass_shootings/incidents-2016-  
GVA.csv")  
incidents_2017_GVA <-  
read_csv("C:/Users/mark3/Desktop/dc_witness/mass_shootings/incidents-2017-  
GVA.csv")  
incidents_2018_GVA <-  
read_csv("C:/Users/mark3/Desktop/dc_witness/mass_shootings/incidents-2018-  
GVA.csv")  
incidents_2019_GVA <-  
read_csv("C:/Users/mark3/Desktop/dc_witness/mass_shootings/incidents-2019-  
GVA.csv")  
incidents_2020_GVA <-  
read_csv("C:/Users/mark3/Desktop/dc_witness/mass_shootings/incidents-2020-  
GVA.csv")  
incidents_2021_GVA <-  
read_csv("C:/Users/mark3/Desktop/dc_witness/mass_shootings/incidents-2021-  
GVA.csv")
```

```
# Taking only DC mass shootings (when "State" = "District of Columbia")
```

```
MS_2014 <- incidents_2014_GVA[which(incidents_2014_GVA$State == "District of  
Columbia"), ]
```

```

MS_2015 <- incidents_2015_GVA[which(incidents_2015_GVA$State == "District of
Columbia"), ]
MS_2016 <- incidents_2016_GVA[which(incidents_2016_GVA$State == "District of
Columbia"), ]
MS_2017 <- incidents_2017_GVA[which(incidents_2017_GVA$State == "District of
Columbia"), ]
MS_2018 <- incidents_2018_GVA[which(incidents_2018_GVA$State == "District of
Columbia"), ]
MS_2019 <- incidents_2019_GVA[which(incidents_2019_GVA$State == "District of
Columbia"), ]
MS_2020 <- incidents_2020_GVA[which(incidents_2020_GVA$State == "District of
Columbia"), ]
MS_2021 <- incidents_2021_GVA[which(incidents_2021_GVA$State == "District of
Columbia"), ]

# Combining all years into one dataframe

MS_1421 <- rbind(MS_2014,MS_2015,MS_2016,MS_2017,MS_2018,MS_2019,MS_2020,MS_2021)

# Writing dataframe to new CSV file

write.csv(MS_1421, "mass_shootings_2014-2021.csv")

```

—

The resulting .csv file was Geocoded with Latitude and Longitude values using Geocode by Awesome Table in Google Sheets, then converted into a GeoJSON file using geojson.io to satisfy compatability with JavaScript and MapBox's map-building API.

The following is the code for the HTML file. It applies the data points against a MapBox map as well as a pop-up box listing each mass shooting's location, date, number of persons killed, and number of persons injured. The HTML output file is included in the zip file.

Language: HTML/CSS/JS

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Display a popup on click</title>
<meta name="viewport" content="initial-scale=1,maximum-scale=1,user-scalable=no">
<link href="https://api.mapbox.com/mapbox-gl-js/v2.3.1/mapbox-gl.css"
rel="stylesheet">
<script src="https://api.mapbox.com/mapbox-gl-js/v2.3.1/mapbox-gl.js"></script>
<script src="https://gist.github.com/ogtaste/dfc879ceb85bb711a24a02735981c2e1.js">
</script>
<style>
body { margin: 0; padding: 0; }
#map { position: absolute; top: 0; bottom: 0; width: 100%; }
</style>
</head>

```

```
<body>
<style>
  .mapboxgl-popup {
    max-width: 400px;
    font: 12px/20px 'Montserrat', Arial, Helvetica, sans-serif;
  }
</style>
<div id="map"></div>
<script>
  mapboxgl.accessToken =
'pk.eyJ1IjoibWFya3dpdG5lc3MiLCJhIjoiy2tybzdmbTg4MDZ4eTJubXduazJ3NGlvaYJ9.LKFYG34Hs
jA8zYuTHFqouQ';
  var map = new mapboxgl.Map({
    container: 'map',
    style: 'mapbox://styles/mapbox/streets-v11',
    center: [-77.04, 38.907],
    zoom: 11.15
  });

  map.on('load', function () {
    map.addSource('shootings', {
      type: 'geojson',
      // Use a URL for the value for the `data` property.
      data: 'weird.geojson'
    });
    // Add a layer showing the places.
    map.addLayer({
      'id': 'shootings',
      'type': 'circle',
      'source': 'shootings',
      'paint': {
        'circle-stroke-color': '#000',
        'circle-stroke-width': 1,
        'circle-color': '#ff0000'
      }
    });

    // When a click event occurs on a feature in the places layer, open a
    popup at the
    // location of the feature, with description HTML from its properties.
    map.on('click', 'shootings', function (e) {
      var coordinates = e.features[0].geometry.coordinates.slice();
      var location = e.features[0].properties.address;
      var date = e.features[0].properties.date;
      var killed = e.features[0].properties.killed;
      var injured = e.features[0].properties.injured;
      var totalcasualties = e.features[0].properties.total_casualties;

      // Ensure that if the map is zoomed out such that multiple
      // copies of the feature are visible, the popup appears
      // over the copy being pointed to.
      while (Math.abs(e.lngLat.lng - coordinates[0]) > 180) {
        coordinates[0] += e.lngLat.lng > coordinates[0] ? 360 : -360;
      }
    });
  });
</script>
```

```

        new mapboxgl.Popup()
            .setLngLat(coordinates)
            .setHTML("Date: " + date + "<br>" + "Location: " + location + "
<br>" + "Killed: " + killed + "</br>" + "Injured: " + injured + "<br>" + "Total
Casualties: " + totalcasualties)
            .addTo(map);
    });

    // Change the cursor to a pointer when the mouse is over the places layer.
    map.on('mouseenter', 'shootings', function () {
        map.getCanvas().style.cursor = 'pointer';
    });

    // Change it back to a pointer when it leaves.
    map.on('mouseleave', 'shootings', function () {
        map.getCanvas().style.cursor = '';
    });
});
</script>

<div class='shootingINFO'>
    <div><strong>Date:</strong> <span id='Incident Date'></span></div>
    <div><strong>Location:</strong> <span id='Address'></span></div>
    <div><strong>Killed:</strong> <span id='Killed'></span></div>
    <div><strong>Injured:</strong> <span id='Injured'></span></div>
    <div><strong>Total Casualties:</strong> <span id='Total Casualties'></span>
</div>
</div>

</body>
</html>

```

—

The above code is able to be put into a box in WordPress. I'm sure you're already familiar with this since you've shown me a map before but with homicides instead of mass shootings. So, the code is there if you or the programmers want to use or modify it.