Intense Rainfall and Flash Flood Reports – Webpage

Michael Erickson (WPC/NOAA, CIRES) – James Nelson (WPC/NOAA) – Mark Klein (WPC/NOAA) – William Diment (CIRES)

Abstract

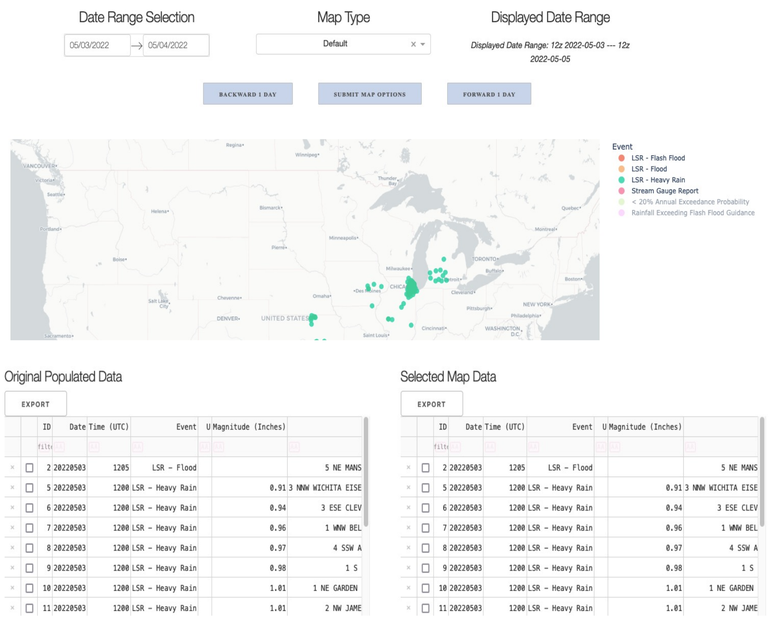
There has been a need for an updated and interactive location where Local Storm Reports (LSRs) for heavy rainfall and flash flood/regular flood can be collected and coagulated for ease of access. This website provides such an access point, providing researchers with numerous tools to be able to look up recent and archived LSRs, displaying information on a map and using data tables to display the information in a common CSV format.

The user is able to select information from the map using a number of selection tools, and is able to parse the information in the data table selection itself using a number of filtering tools available in the data table. The user is then able to export data to download using these filtered tools. This enables for a more precise selection of LSRs, and allows for users to more easily pinpoint weather patterns within the data that may be occurring.

Introduction

By default, the website automatically loads flood reports for the current day over a 48-hour period between 12 UTC and 12 UTC. A text box of metadata (e.g. flood type, latitude, longitude, etc.) associated with the flood observation will appear with a mouse hover over the observation point on the map. The page is updated with new flood reports every hour. The user can change the date range of the flood observations by using the "Date Range Selection" Calendar option, either typing in a date range in the format of DD/MM/YYYY or using the forward/back buttons to select a date range. For example, if a user would like to select a date range from 05/03/2022 to 05/04/2022, type those numbers as presented here into the boxes, or click on the backward/forward button until reaching those months, then click on those date selections

The website generates intense rainfall and flood data tables below the display map according to the date range selected by the user and the spatially subsetted region using the Lasso and Box tools. The data is organized into two tables – the original generated data, and the user selected data on the right



	ID	Date	Time (UTC)	Event	U Magnitude (Inches)	
2	20220503	1205	LSR - Flood			5 NE MANS
5	20220503	1200	LSR - Heavy Rain		0.91	3 NNW WICHITA EISE
6	20220503	1200	LSR - Heavy Rain		0.94	3 ESE CLEV
7	20220503	1200	LSR - Heavy Rain		0.96	1 WNW BEL
8	20220503	1200	LSR - Heavy Rain		0.97	4 SSW A
9	20220503	1200	LSR - Heavy Rain		0.98	1 S
10	20220503	1200	LSR - Heavy Rain		1.01	1 NE GARDEN
11	20220503	1200	LSR - Heavy Rain		1.01	2 NW JAME

Observation Data Types

The observations include:
National Weather Service (NWS) Local Storm Reports (LSRs) for Flash Flood, Flood, and Heavy Rain.
Stream Gauge Reports - Streamflow measurements from the United States Geological Survey exceeding minor flood stage and having an upstream drainage basin area less than 300 km^2

In addition, two proxies are included.
< 20% Annual Exceedance Probability - Stage IV rainfall analysis exceeding the 5-year Average Recurrence Interval from the NOAA Atlas 14 data (except Atlas 2 data in the Pacific Northwest)'
Rainfall Exceeding Flash Flood Guidance - Stage IV rainfall analysis exceeding 1, 3, and 6 hour Flash Flood Guidance'

Programming Methodology

The rainfall reports page was designed in Python, using Dash as the web framework. Dash is designed to enable the user to write HTML code using Python, and includes a large number of common design components that interact seamlessly with both HTML and custom Python code. Dash also enables the use of 'callbacks' to dynamically interact with the page using Python.

Following that, Dash is also tightly integrated with Plotly, a premier Python plotting tool. With all of this combined, Dash is a very powerful, robust web framework combining common web design tools that can then access the Python programming environment, enabling complex plotting and deep interaction between constituent parts of a given webpage.

The code interacts with the server architecture (Apache) via the use of a WSGI program; Gunicorn, which acts as the gateway to the Python code itself

A list of primary package versions follows:

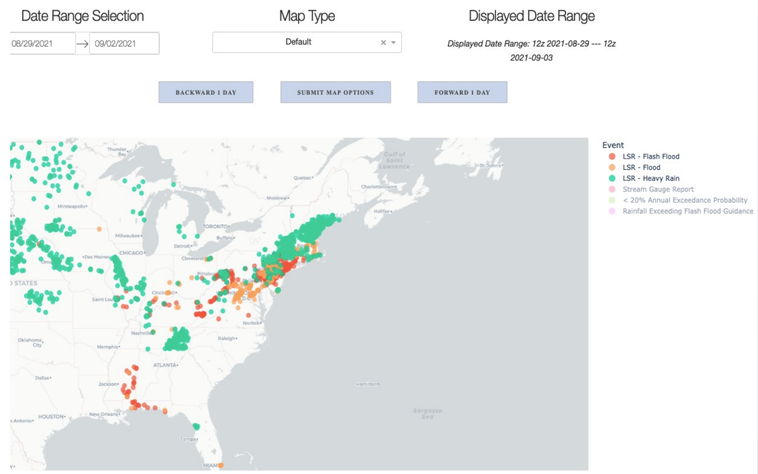
- Dash – 2.0.0
- Plotly – 5.3.1
- Pandas – 1.3.4
- Numpy – 1.17.4
- Gunicorn – 20.0.4

Following is a code example of a callback in Dash to generate a given data table

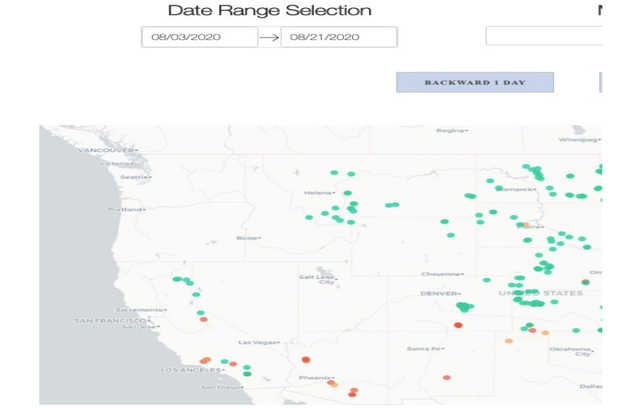
```
@app.callback(
  Output(component_id='csv_table_with_selected_data', component_property='children'),
  [Input(component_id='noaaMapID', component_property='selectedData'),
  Input(component_id='noaaMapID', component_property='figure'),
  Input(component_id='csv_table', component_property='children'),
  Input(component_id='noaaMapID', component_property='restyleData')])
def test_output(selectedData, fig, original_csv, restyleData):
  csvOutput = ''
  origCSVList = []
  csvList = []
  EventList = ['LSR - Flash Flood', 'LSR - Flood', 'LSR - Heavy Rain', 'Stream Gauge Report', '< 20% Annual Exceedance Probability'
  columnNames = ["ID","Date","Time","UTC","Event","U","Magnitude","Loc","City","State","Lat","Lon","Issuance","Remark","Source"]
  stylebit = 0

  csvLength = 0


  if(selectedData):
   for x in range(0, len(selectedData['points'])):
    csvList.append(selectedData['points'][x]['customdata'])

   columnNames = ["ID","Date","Time","Event","U","Magnitude","Loc","City","State","Lat","Lon","Issuance","Remark","Source"]
   noaaDF=pd.DataFrame(columns=columnNames)

   for dataPoint in csvList:
     noaaDF.loc[len(noaaDF)] = dataPoint
   csvOutput = noaaDF.to_csv(index=False, encoding='utf-8')

   csvOutput ="data:text/csv;charset=utf-8,%EF%BB%BF" + urllib.parse.quote(csvOutput)
   noaaDataTable = dash_table.DataTable(
    id='noaa_csv_table',
    columns=[{"name":i,"id": i, "deletable": True, "selectable": True} for i in noaaDF.columns],
    data=noaaDF.to_dict('records'),
    export_columns='visible',
    export_format='csv',
    export_headers='names',
    page_action='native',
```

Display Examples

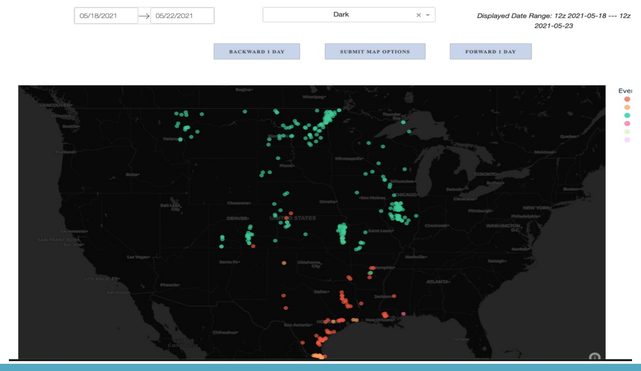
Pictured below: Hurricane Ida and the Local Storm Reports it generated from 8/29/2021 – 9/03/2021



Pictured Below: Western United States from 8/03/2021 – 8/21/2021



Pictured Below – CONUS from 5/18/21 – 5/23/21 in Dark Mode



Acknowledgements

- CIRES
- NOAA/WPC
- CU Boulder
- WOC (Web Operation Center, NOAA)
- NCO (NCEP Central Operations, NOAA)