Tools for cracking the snowpack's mysteries

MoSnoPro-US 🏔️❄️
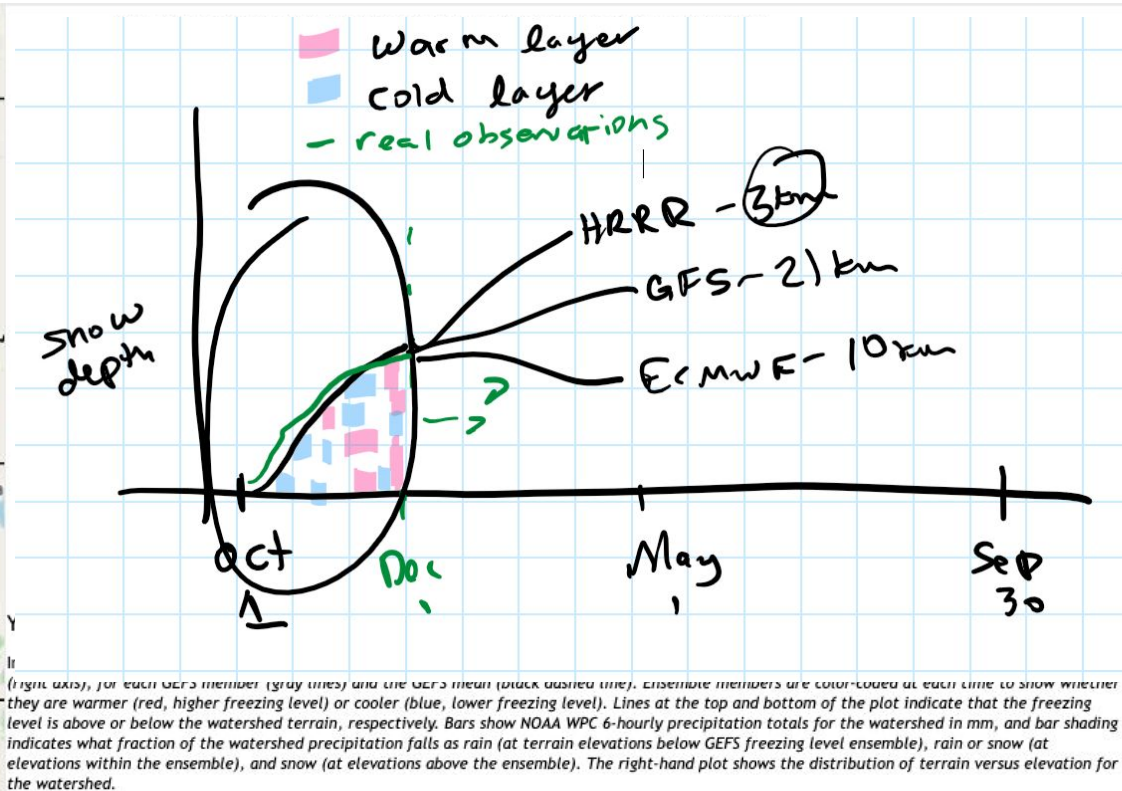
Danny Hogan, Bow Sruangsung, Xinya Tang, Jane Dai

Upper Columbia-Priest Rapids

GFS
ECMWF
Activate Both Models To Compare

☑ Map

Point Obs:
☐ Snow-Lev Radar ▲

Precip % as Rain/Snow Based on Wshed Z0C

- No Precip next 7 d
- 100% Snow
- 80-99% Snow
- 60-79% Snow
- 40-59% Rain/Snow
- 60-79% Rain
- 80-99% Rain
- 100% Rain
- Missing Data

Hover for more information
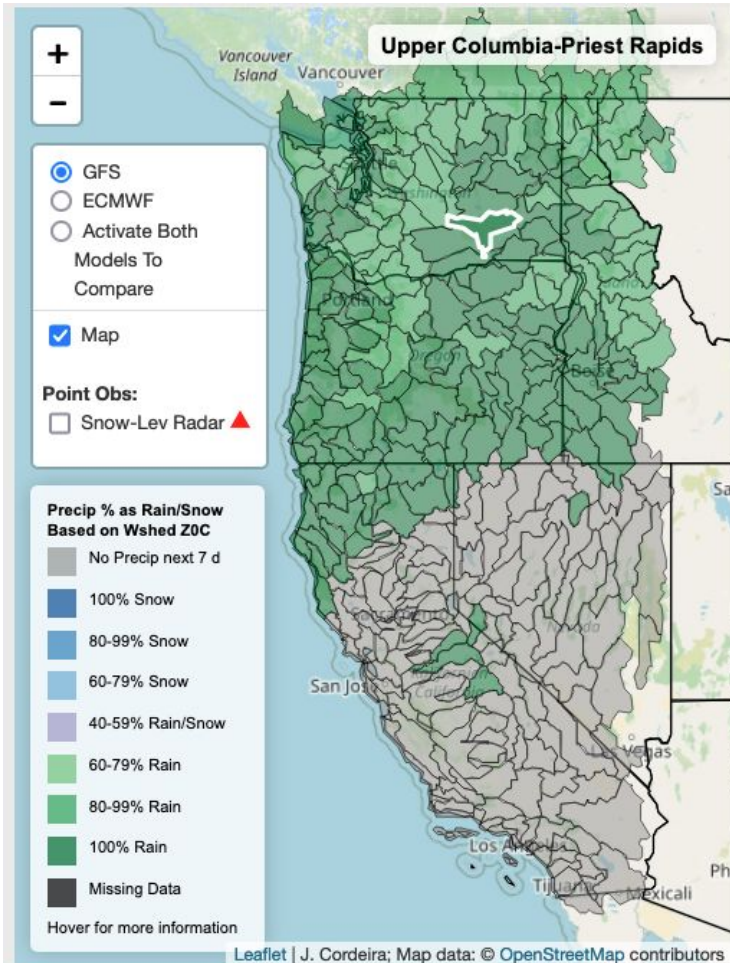
Leaflet | J. Cordeira; Map data: © OpenStreetMap contributors

Warm layer
Cold layer
— real observations

HRRR – 3km
GFS – 21 km
ECMWF – 10 km

snow depth

Oct
Dec
May
Sep 30

(right axis), for each GEFS member (gray lines) and the GEFS mean (black dashed line). Ensemble members are color-coded at each time to show whether they are warmer (red, higher freezing level) or cooler (blue, lower freezing level). Lines at the top and bottom of the plot indicate that the freezing level is above or below the watershed terrain, respectively. Bars show NOAA WPC 6-hourly precipitation totals for the watershed in mm, and bar shading indicates what fraction of the watershed precipitation falls as rain (at terrain elevations below GEFS freezing level ensemble), rain or snow (at elevations within the ensemble), and snow (at elevations above the ensemble). The right-hand plot shows the distribution of terrain versus elevation for the watershed.

Streamlit   plotly   anvil   🧐???

Streamlit

Strengths

- Provides open-source app framework, ease of deploy
- Quick to learn, minimum front-end experience required
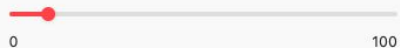- Allows dynamic updates

Limitations

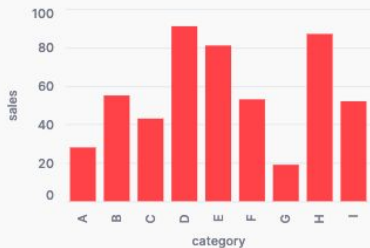- Limited customization in UI/UX
- State management limitations

Takeaway: Our project uses an interactive UI, but Streamlit may struggle with some interactive features.

Streamlit

Pick a number


```
number = st.slider("Pick a number", 0, 100)
```

Pick a file

Drag and drop files here
Limit 200MB per file • TXT

Browse files

```
file = st.file_uploader("Pick a file")
```

Pick a color


```
color = st.color_pi
```



```
st.altair_chart(my_chart)
```

Pick a pet

- ● Dog
- ○ Cat
- ○ Bird

```
pet = st.radio("Pic
```

Pick a date

‹ November ⌄ 2024 ⌄ ›

Su	Mo	Tu	We	Th	Fr	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

```
date = st.date_input("Pick a date")
```

plotly DASH

Strengths

- Rich interactive visualizations using Plotly—**highly customizable**
- Robust for **complex data & scalable applications**

Limitations

- Slightly **steeper learning curve** compared to Streamit
- UI customization requires more coding—**making quick changes challenging**
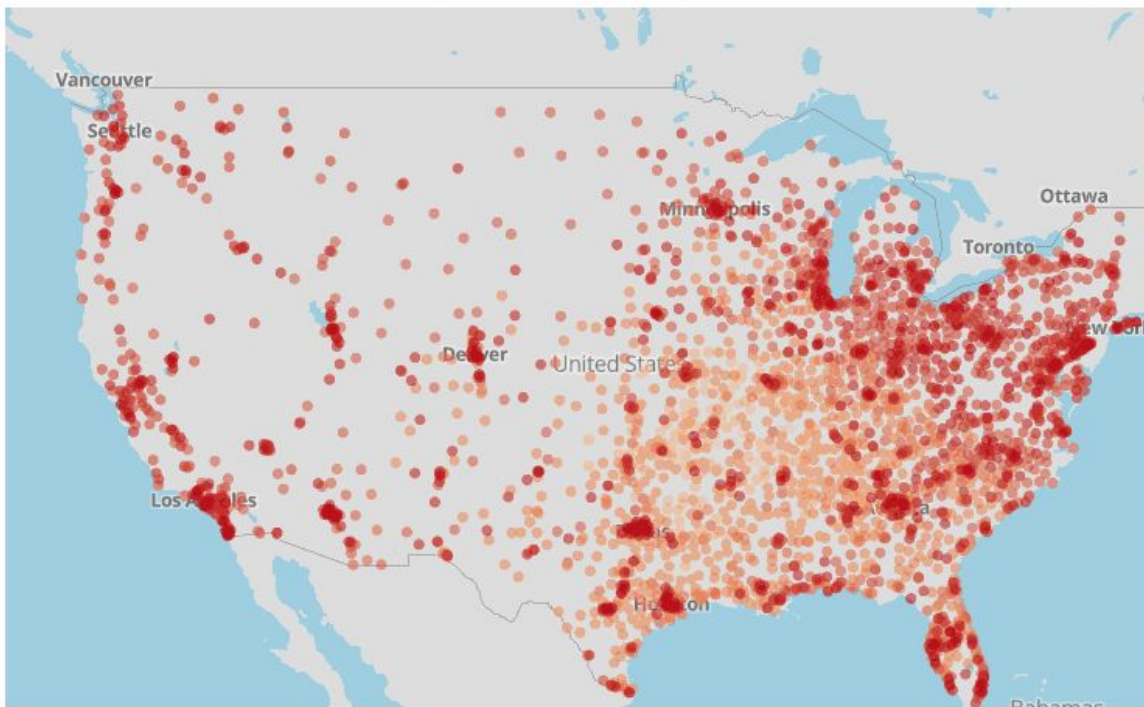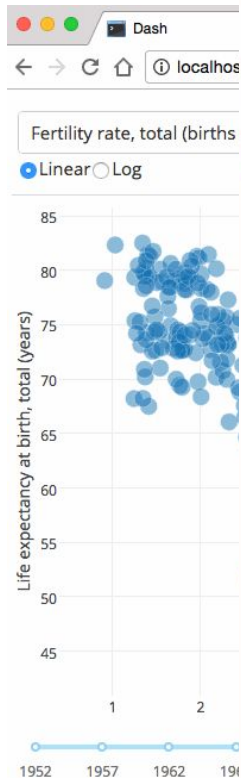
Issues on Github   Open 456   Closed 1,360

Integration challenges or performance tweaks

Strengths

- Work front-end to back-end in python
- Free cloud hosting
- Simple deployment
- Interactive GUI
- Good examples and tutorials

Limitations

- GUI features may be limiting or difficult to learn
- Finicky interface
- Not open source, proprietary driven
- Collaboration is limited in free tier

Takeaway: Anvil would be good for a personal project, but likely not for a collaborative group project

Example of features

Summary and Selection

Option	Strengths	Limitations
Streamlit	Simple and Fast setup (Prototype, Data Centric application)	Limited customization Struggle with complex layouts
Plotly Dash	High interactivity with rich and customizable visualization Scalable and Robust (Detailed data-driven application)	Slightly steeper learning curve
Anvil	Low-code Drag-and-Drop UI (simpler applications)	Limited customization Some features: paid tier