BWare

Background

Objective:

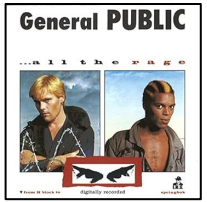
Create a dashboard to analyze historical trends in buoy weather data

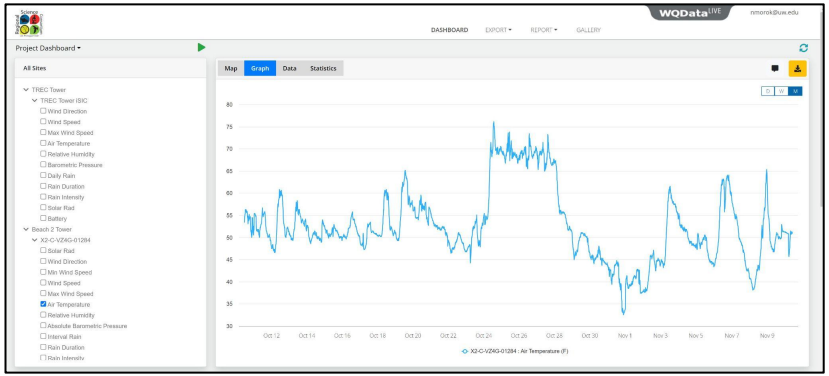
Users:

The dashboard will be used by our stakeholder, the Regional Science Consortium, and general members of the public

Product:

The dashboard is meant to augment the experience of their current data interface





Tkinter (Tk interface)

How it works: Tkinter is the standard Python interface for building GUIs using the Tk toolkit. GUIs are built in Windows using Widgets that are organized using Frames.

Pros:

- Lightweight + Easy for simple tasks
- Built-in themes + Variety of templates
- Many online resources + well supported

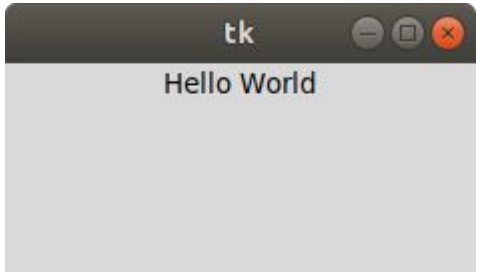
Cons:

- Challenging to build, complex
- Themes look old/out of date
- Lacks advanced widgets (e.g. data-driven views)

Hello World

```
#!/usr/bin/env python3
from tkinter import *
root = Tk()                                 # Create the root (base) window
w = Label(root, text="Hello, world!")       # Create a label with words
w.pack()                                    # Put the label into the window
root.mainloop()                             # Start the event loop
```

https://en.wikipedia.org/wiki/Tkinter



https://www.geeksforgeeks.org/hello-world-in-tkinter/#

https://iot4beginners.com/look-and-feel-customization-on-tkinter-python-gui/

Tkinter (Tk interface)

How it works: Tkinter is the standard Python interface for building GUIs using the Tk toolkit. GUIs are built in Windows using Widgets that are organized using Frames.

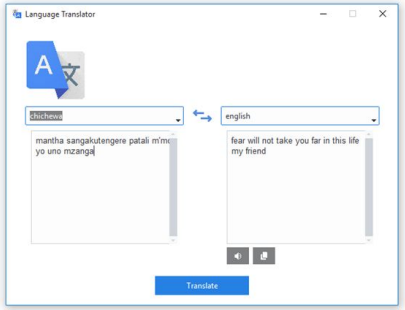
Pros:

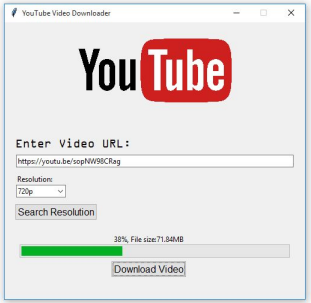
- Lightweight + Easy for simple tasks
- Built-in themes + Variety of templates
- Many online resources + well supported

Cons:

- Challenging to build, complex
- Themes look old/out of date
- Lacks advanced widgets (e.g. data-driven views)

More Advanced Examples





https://dev.to/khumbolamulungu/top-11-advanced-tkinter-projects-you-should-try-this-year-115l

https://iot4beginners.com/look-and-feel-customization-on-tkinter-python-gui/

Shiny for Python

Shiny for Python

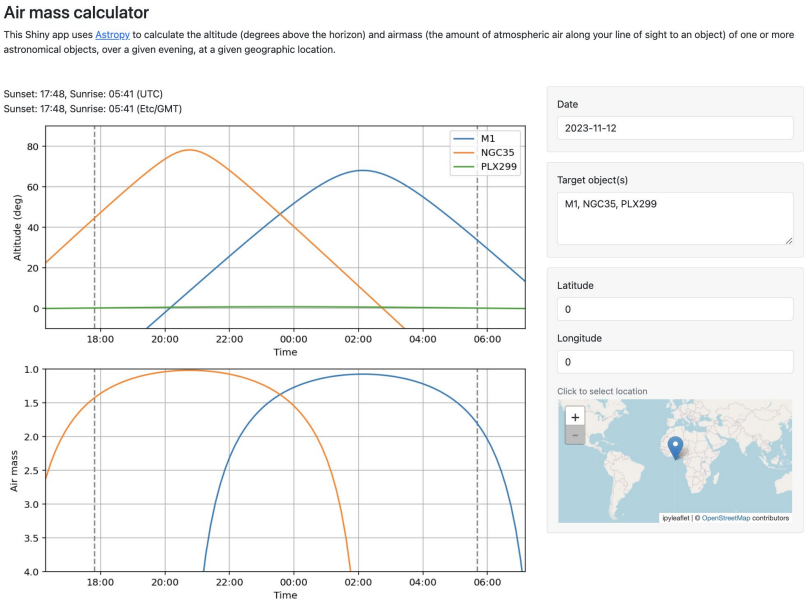
How it works: Shiny for Python can be used to build web applications with Python code. It is similar to Shiny for R. Designed around reactive programming: “When a (reactive) input is modified, it causes a (reactive) output function to re-execute”

Pros:

- Specifically designed for interactive applications
- Easy to learn if already familiar with R
- Templates are moderns + good with geospatial data

Cons:

- Slow (rendering and interactions can be very slow) because they are primarily single-threaded
- Advanced styling requires a lot of css



https://gallery.shinyapps.io/airmass/

https://shiny.posit.co/py/gallery/

Shiny for Python

Shiny for Python

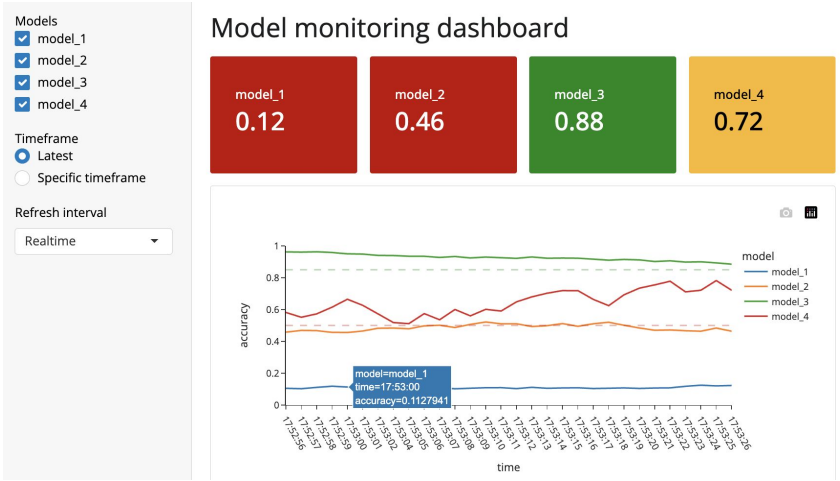
How it works: Shiny for Python can be used to build web applications with Python code. It is similar to Shiny for R. Designed around reactive programming: “When a (reactive) input is modified, it causes a (reactive) output function to re-execute”

Pros:

- Specifically designed for interactive applications
- Easy to learn if already familiar with R
- Templates are moderns + good with geospatial data

Cons:

- Slow (rendering and interactions can be very slow) because they are primarily single-threaded
- Advanced styling requires a lot of css



https://gallery.shinyapps.io/model-score/

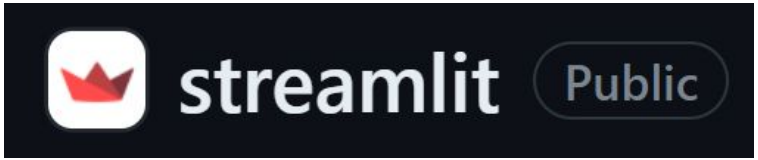
https://shiny.posit.co/py/gallery/

Streamlit Overview

Originally designed with machine learning in mind, Streamlit is an **open source platform** designed to **quickly** design interactive web apps through python

- **Use of python code:**
  - Streamlit can be coded using python.
- **Ease of use/ speed:**
  - “Streamlit lets you turn data scripts into shareable web apps in minutes, not weeks.” - README.md
- **Ability to use widgets:**
  - Web interface will be enhanced by user interactions via widgets available via streamlet
- **Community Cloud Platform**
  - Streamlit provides free deployment services via their Community Cloud Platform
  - 1GB free space for testing

Streamlit: Github Analysis



- Community
  - 28.4 thousand stars
  - 2.5 thousand forks
- Issues
  - 669 open: 3042 closed
  - Issues generally closed within 2-3 days
  - Most issues seem very specific
    - Not problems that would impact our project
- README.md
  - Well documented
  - Hyperlinks, images, concise instructions
- License
  - Apache 2.0