HAS Tools:

Beyond notebooks: scripts and modules

October 28, 2024

Why do something besides notebooks?

https://www.youtube.com/watch?v=7jiPeIFXb6U

https://docs.google.com/presentation/d/1n2RlMdmv1p25Xy5thJUhkKGvjtV-dkAIsUXP-AL4ffI/edit#slide=id.g362da58057_0_1

What's going on here?

```
In [3]: def f(x): return x + 2

In [2]: y = f(2)

In [4]: y == 4
Out[4]: False

In [5]: print(y)
        5
```

This is getting pretty long!

Notebooks should help you do analysis, not get in the way by making you scroll forever.

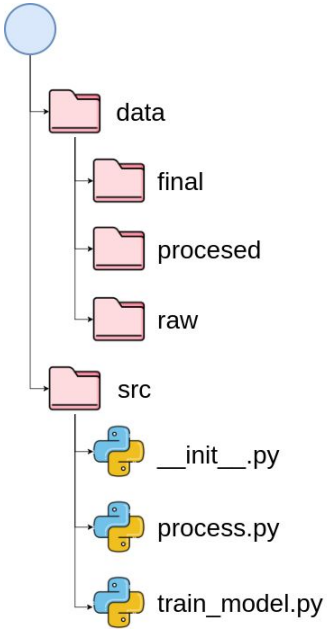
- What if you wanted to run this code for many locations?
- Just copy and paste the cell and change the lat/lons?
- Make a function and store it up top?
- What if you have a bunch more other functions?

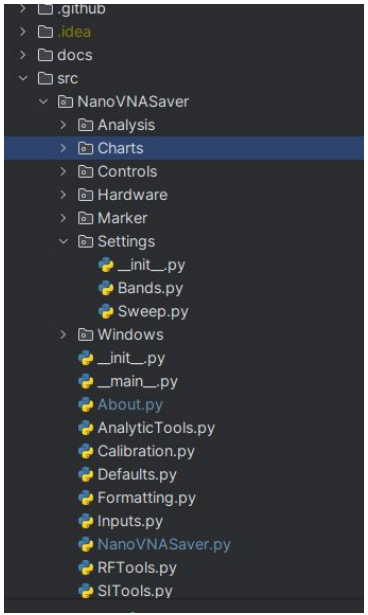
```
# Select temperature data for Tucson and Buffalo
tucson_lat, tucson_lon = 32.25, -110.97  # Tucson coordinates
buffalo_lat, buffalo_lon = 42.88, -78.87  # Buffalo coordinates

# Convert lon from 0 to 360 format to -180 to 180
tucson_lon = (tucson_lon + 360) % 360
buffalo_lon = (buffalo_lon + 360) % 360

# Extract temperature time series for Tucson and Buffalo
tucson_temp = ds['air'].sel(lat=tucson_lat, lon=tucson_lon, method='nearest')
buffalo_temp = ds['air'].sel(lat=buffalo_lat, lon=buffalo_lon, method='nearest')

# Group data by month for both locations
tucson_grouped = tucson_temp.groupby('time.month')
buffalo_grouped = buffalo_temp.groupby('time.month')

# Convert temperatures to Celsius for plotting
tucson_data = [(month, temps.values - 273.15) for month, temps in tucson_grouped]
buffalo_data = [(month, temps.values - 273.15) for month, temps in buffalo_grouped]

# Prepare the figure
fig, ax = plt.subplots(figsize=(12, 6))

# Plot Tucson boxplots
tucson_positions = np.arange(1, 25, 2) +0.25  # Positions for Tucson boxplots (odd numbers)
bp_tucson = ax.boxplot(
    [temps for _, temps in tucson_data],
    positions=tucson_positions,
    patch_artist=True,
    medianprops={'color': 'black'},
    flierprops={'marker': 'o', 'markerfacecolor': 'gray', 'alpha': 0.5},
    boxprops={'facecolor': 'salmon'}
)

# Plot Buffalo boxplots in the next step
buffalo_positions = tucson_positions + 0.75  # Positions for Buffalo boxplots (even numbers)
bp_buffalo = ax.boxplot(
    [temps for _, temps in buffalo_data],
    positions=buffalo_positions,
    patch_artist=True,
    medianprops={'color': 'black'},
    flierprops={'marker': 'o', 'markerfacecolor': 'gray', 'alpha': 0.5},
    boxprops={'facecolor': 'lightblue',}
)

# Customize the plot
ax.set_xlabel('Month')
ax.set_ylabel('Temperature (°C)')
ax.set_title('Monthly Temperature Distributions in Tucson, AZ and Buffalo, NY')
ax.set_xticks(np.arange(1.5, 25, 2))  # Center the labels between the two boxplots
ax.set_xticklabels(['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
                    'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'])

# Clean up axes
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)

# Add subtle horizontal lines from y-axis
ax.yaxis.grid(True, linestyle='--', alpha=0.25)
ax.axhline(0, color='black', linewidth=0.5, zorder=-1)  # Add horizontal line at 0

# Add legend to differentiate Tucson and Buffalo
plt.legend(
    [bp_tucson["boxes"][0], bp_buffalo["boxes"][0]],
    ['Tucson', 'Buffalo'],
    loc='upper right',
    fontsize='large'
)

plt.tight_layout()
plt.show()
```

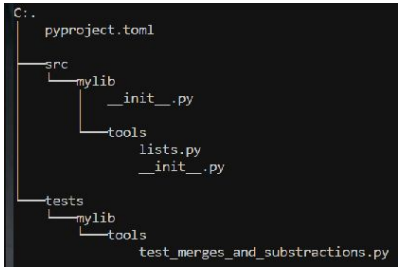
Scripts & modules can help you automate and organize your code

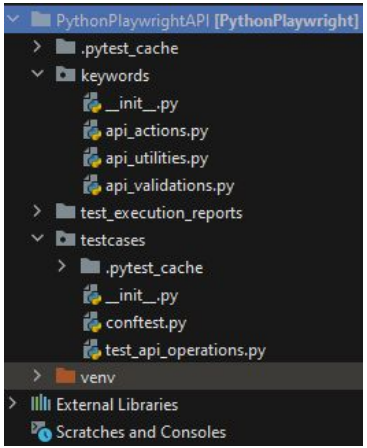
Code is organized into “modules” which are just files with python code inside

If you run these directly, they are called “scripts”









Let’s go to codespaces for some demo time

For more info about project structuring see:
https://realpython.com/python-application-layouts/