**Design Pattern:** Façade (Structural)

**Module:** matplotlib/lib/matplotlib/pyplot.py

pyplot implements the façade design pattern, acting as an interface for a more complex subsystem involving classes like Figure, Axes, Subplot, Artist, and others.

By offering clients the option to interact with a single interface rather than multiple classes, the system becomes much easier to use. Instead of using methods from the façade’s various dependencies, clients can just use pyplot’s methods, which have the classes of the subsystem plot points, draw bars, label axes, save figures, etc. This also weakens coupling between the clients and the subsystems.

**Justification:**

pyplot, like all façades, is an alternative interface, not a replacement – subsystem objects are shielded, but clients are not restricted from accessing them. Clients can still directly interact with the subsystem to manually create Figures, Axes, and plots if they wish.

The façade design pattern may be confused with the mediator design pattern because they both abstract the functionality of other classes. However, mediator facilitates communication between peers (and it is therefore known to the peer classes), while façade provides interface to a subsystem, and isn’t known to the subsystem classes. Although there are some helper methods in the subsystem made to help pyplot act as a façade, like Figure’s \_gci(), the functionalities of the subsystems are independent from pyplot, and so pyplot is firmly an implementation of the façade design pattern.