

	Zeppelin	Jupyter	RStudio	Plotly Dash
Core Experience				
Notebook Functionality	★ chart creator, interactive tables, dynamic forms.	Not much interactivity, but better UI than rmd. Obscure metadata features.	.rmd feels forced because it's in a text editor and you have to handwrite the cell markdown	Not for notebooks. Chart Studio is hard to use regardless.
Browser-based	Zepl or Apache-Zeppelin	Tree, Lab, or Hub	★ RStudio, Server Pro (same as desktop app)	Plot.ly Chart Studio
Desktop App	None	nteract.io	★ RStudio (native browser)	None
File Explorer	Just a list on main page	JupyterLab's left pane or /tree	★ Files pane	plot.ly/my_files
Config Backend in UI	partial solution, not reliable (also, conda UI not reliable)	UI settings not reliable (also, conda UI not reliable)	★ Packrat package mgmt (but people want renv)	N/A, not for running code
Publish to a Portal	Zepl (upload only, no push/ publish) (only note.json format, can't import .ipynb, but converter)	★ Google Colab... interesting (upload ipynb to google drive) ^ MSFT next? JupyterHub (no commercial provider) (upload only, no push/ publish)	RStudio Connect (Shiny & .rmd are dynamic/ interactive, but .ipynb is static)	Plot.ly Dash (charts only, not notebooks. push to portal) ^ <i>UPDATE: notebook apps!</i>
Dashboard Mode	Resizable (cell width=1-12)	jupyter/dashboards	★ via R Shiny Dashboards	Primarily a dashboard tool
Pluggable Themes	Requires editing source code	dunovank/jupyter-themes	★ In preferences menu	Only for dashboards
Extensibility				
Language Agnostic	★ Interpreters (reliable out of the box)	Kernels	R, Python via Reticulate	R, Python, Javascript
Plugins	★ Helium (slides)	nbextensions + UI Extensions Manager	Tidyverse community releases.	Some React.js web component support
Deployment				
Install via Package Manager	many maven packages	\$ pip install jupyter jupyterlab \$ conda install jupyterlab	N/A	\$ pip install plotly Or \$ pip install dash
Install via Homebrew	\$ brew install apache-zeppelin	\$ brew install jupyter	\$ brew cask install rstudio	No mention of brew dash
Install via Docker	0.8.1 (needs work)	★ all-spark-jupyter	rocker/rstudio	/plotly is their internal tools
Deploy via Kubernetes	\$ helm install stable/zeppelin Only `Shiro` auth? Helm chart w Spark + Docs	★ JupyterHub (helm chart)	RStudio Server Pro (no helm chart)	“Dash has a react/ flask app at its core,” which can be deployed as a web service.
Dependencies	Conda manages NPM. No bridge to Zepl.	Conda manages NPM. Lab/ tree caused problems with node extensions and never got it fully working.	Just works as a standalone app out of the box.	Separate packages for plotly ad chart studio (of which there are multiple versions).
Rendering Notebooks				
Paired with Frontend Library	AngularJS interpreter	\$ npm install jupyter-react-js	★ R Shiny framework	React.js components
Renders in Github	note.json doesn't render But there is a plugin (broken).	★ .ipynb renders	.rmd does not render	Just a .json of chart config
MVC Webapp Frameworks	No integrations. Zeppelinhub viewer became Zepl	`nbviewer` is a standalone webapp on top of tornado, so people just link out to JupyterHub.	Hacky CMS approach w Django	★ Embed in Flask, Django (again, just graphs)
Integrations				
Pandas	★ UI Table	Just stylized tables	reticulate `r` View(py\$DF)`	N/A, relies on ipython
Apache Spark	★ dedicated interpreters for Spark and Beam (Databricks appears to have forked zeppelin for their cloud)	`findspark` and manually managing `spark context`	Sparklyr (improves SparkR)	N/A
Plotting	★ reliable Matplotlib interpreter and automated dataframe plots	Unreliable Matplotlib	ggplot2 equivalent is reliable	Primarily a plotting library
Pluggable Data Sources	Zepl Data Sources	Manual connections	★ Connections tab in App	“Download SQL client”
Workflow tools	Apache Beam interpreter. (Groovy interpreter would be a good fit for nextflow)	pip airflow (Groovy kernel would be a good fit for nextflow)	(wow, just realized there is no async workflow package for R. made an SO post)	N/A

Machine Learning Lifecycle Tools	AWS (aws-sagemaker-spark-sdk), H2O	★ Polyaxon , Kubeflow . AWS (Sagemaker), GCP (AI Platform), H2O	GCP (cloudml package), AWS (Sagemaker... EC2 + reticulate... , pdocs recommend Jupyter)	N/A
Cloud Providers	GCP (Dataproc), AWS (EMR), Azure (HDinsight)	★ GCP (Colab , DataLab , Dataproc), AWS (~ EMR), ★ Azure (Notebooks)	GCP (Dataproc), AWS (marketplace), Azure (marketplace)	Manual install

SUMMARY

- Zeppelin has the most notebook functionality and integrations.
 - RStudio has the best overall experience.
 - Jupyter has the most adoption/ biggest community.
-
- RStudio is introducing native Python scripts.
 - JupyterHub has the best Kubernetes deployment. Whereas Zeppelin only mentions Shiro auth and RStudio is not fully converted yet.
 - Jupyter dominates the data science ecosystem. I'd chalk this up to collaboration/ marketing w PyData community.
 - Zeppelin has the best Spark support out of the box, including Apache Beam interpreters.
 - Zeppelin has awesome graph database query/ visualization (Neo4j Cypher/ Bloom), which isn't even open source... very surprising.
 - Jupyter's extensions are completely unreliable, whereas Zeppelin's work out of the box.
 - Feels like Databricks forked Zeppelin and didn't contribute improvements back to master branch.
 - **Azure** is the only cloud provider that has launched a standalone notebook service. But the cloud providers support both Zeppelin and Jupyter as part of their services.
 - Databricks notebooks have built-in UI features for MLflow.

OPPORTUNITIES

- No notebooks have built-in UI features for workflows/ pipelines. Maybe it would be nice to track where results are stored and the metadata about the runs... similar to ML lifecycle?
- Is there room for working with deployment scripts for Docker/ Kubernetes CI/ CD? Is there a way to nano/vim into files on the local file system?