# Analytical Frameworks

Sayyor Yusupov (sayyor.yusupov@estudiantat.upc.edu)
Arina Gepalova (arina.gepalova@estudiantat.upc.edu)
Barcelona, Spain - eBISS 2023

## 1. Introduction

There are multiple tools that facilitate data scientists' work in different aspects. Analytical frameworks can be used to manage and organise the data, and simplify the analysis and processing of data in various domains. These frameworks provide a structured and efficient way to work with data, enabling users to extract insights, make data-driven decisions, and derive meaningful conclusions. The frameworks can be devided into 3 groups: Computational Notebooks, Frameworks for Interactive Analysis and Frameworks for Data Stream Analysis.

## 3. Interactive Analytics

Tools like Power BI and Tableau process and analyze massive unstructured data and visualize the results in interactive dashboards in real time.

| | Power BI | Tableau |
|---|---|---|
| Common Features | - interactivity<br>- filters<br>- data preparation<br>- Machine Learning support | - dashboards<br>- standard visualization<br>- collaborative editting<br>- embed to webpages |
| Version Control | no built-in solution,<br>manually with OneDrive, Git | support from 9.3 |
| Data Integration | - Databases<br>- Cloud Services | - Files<br>- Web Data Connectors |
| Operating System | only Windows | Windows, Mac |
| Programming Languages | R, M language | R, Python, Java, C, C++ |
| Performance | better data manipulation,<br>performs better for limited data | scales better to large datasets |
| Cost | | typically more expensive |
| Deployment | migrate to Microsfot Azure<br>Data Gateways for on-premises | any infrastructure on-premises |

Fig. 3. Comparison of Tableau and Power BI.

## 4. Stream Analytics

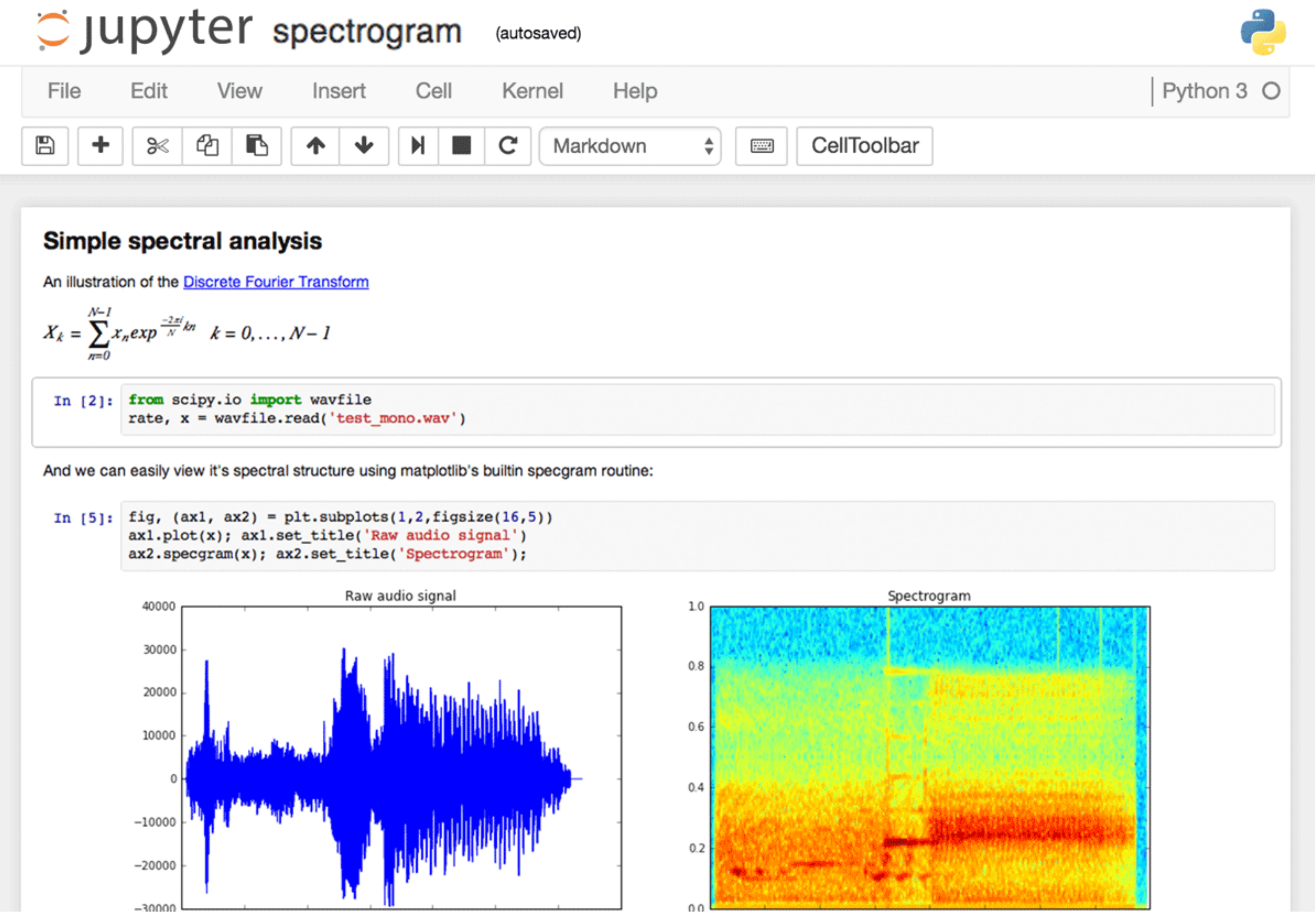
Apache Spark and Apache Flink are popular frameworks that enable the processing of **large volumes of data in real-time** [3].

| | Apache Spark | Apache Flink |
|---|---|---|
| Creator | University of California | Apache Software Foundation |
| Processing | Batch and Stream (micro-batch) | Batch and Stream (native) |
| Latency | Low (seconds) | Low (sub-seconds) |
| Throughput | High | High |
| Scalability | High | High |
| Fault tolerance | Resilient Distributed Dataset | Incremental checkpointing |
| Optimization | Manually by developers | Before execution on the streaming engine |
| Windowing | Timed | Time, Count |
| Iterations | No | Yes |
| SQL Support | Yes | Yes |
| State Backend | HDFS | In-memory, file system, RockDB |
| Language | Python, Java, Scala, R, C#, F# | Python, Java, Scala, SQL |

Fig. 4. Comparison of Data Stream Frameworks

## 2. Computational Notebooks

Notebooks are interactive platforms to **code**, **analyze** data and **visualize** within the same document.



Simple spectral analysis

An illustration of the Discrete Fourier Transform

$$X_k = \sum_{n=0}^{N-1} x_n exp^{\frac{-2\pi i}{N}kn} \quad k = 0, \dots, N-1$$

```
In [2]: from scipy.io import wavfile
        rate, x = wavfile.read('test_mono.wav')
```

And we can easily view it's spectral structure using matplotlib's builtin specgram routine:

```
In [5]: fig, (ax1, ax2) = plt.subplots(1,2,figsize(16,5))
        ax1.plot(x); ax1.set_title('Raw audio signal')
        ax2.specgram(x); ax2.set_title('Spectrogram');
```

Fig. 1. Sample Notebook Layout.

**Reactivity** - automatic updating of outputs based on code changes
**Reproducibility** - another user gets the same notebook state
**Streaming Data** - changes in Data propagated to the notebook (BeakerX, Tempe [2])
**Scheduling** - automatic execution at specified time intervals

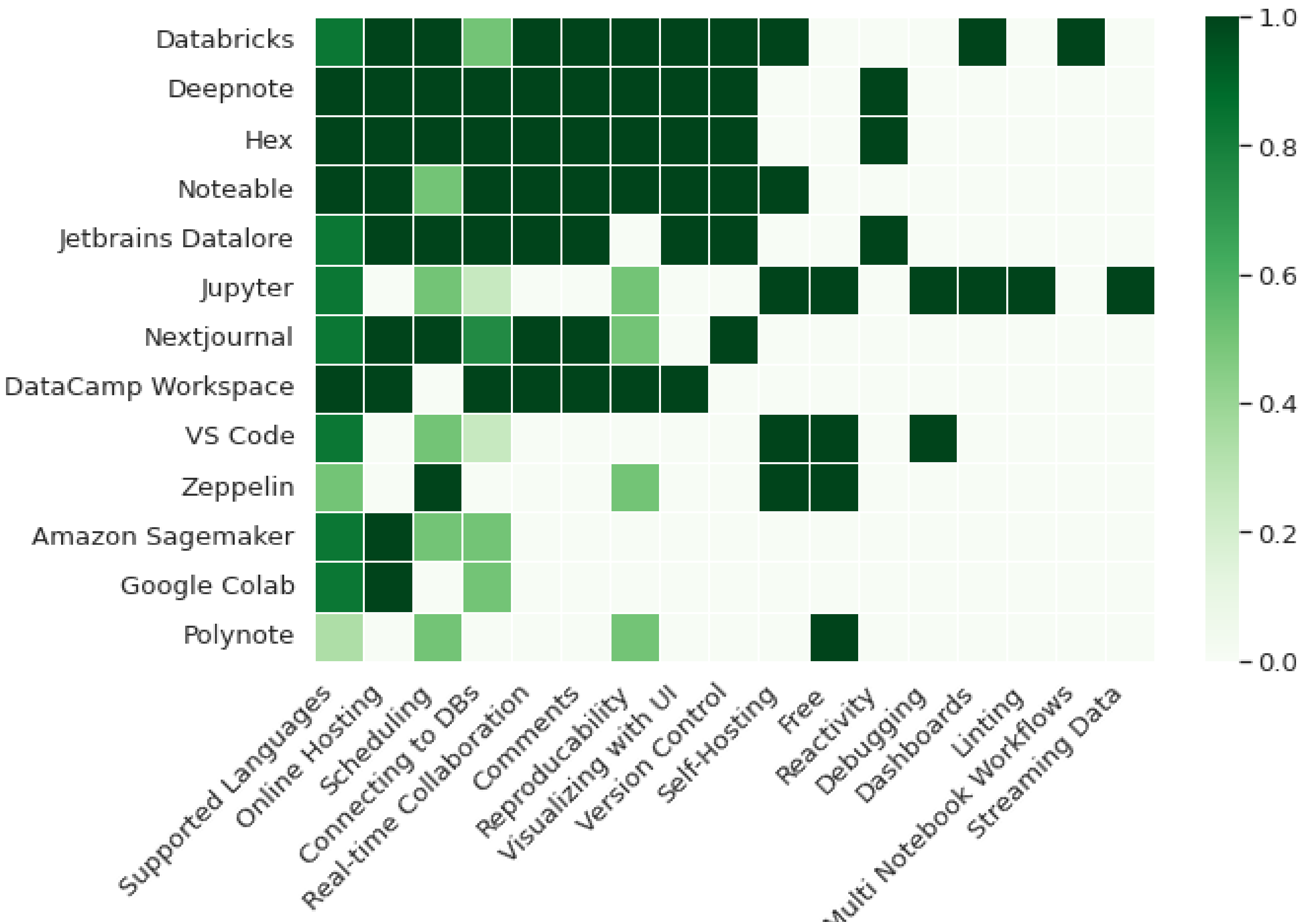


Fig. 2. Different Features of Computational Notebooks, partially based on [1]

Features to Improve [4]:
**Transaction Support** - revert to the previous state
Automatic **Translation** of Languages
**Reliability**: notebook kernels crash due to large input or many cells

## 5. Conclusions

Features that can be improved in Computational Notebooks: sharing, managing code and reliability. The choice of the notebook system depends on the language, setup, the need for reactivity, and additional features required. Furthermore, Tableau and Power BI are market leaders in data visualization. The Flink framework is gaining popularity because it processes data in real-time (in addition to Apache Spark features).

References

[1]. Robert Lacok. Data Science Notebooks. Mar. 2023. url: https://datasciencenotebook. org/.

[2]. Robert DeLine et al. "Tempe: Live scripting for live data". In: 2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC). 2015, pp. 137–141. doi: 10.1109/VLHCC.2015.7357208.

[3]. Ana Almeida et al. "Time series big data: a survey on data stream frameworks, analysis and algorithms". In: Journal of Big Data. SIGMOD '19 10.1 (2023), p. 83. doi: 10.1186/ s40537-023-00760-1. url: https://doi.org/10.1186/s40537-023-00760-1.

[4]. ] Souti Chattopadhyay et al. "What's Wrong with Computational Notebooks? Pain Points, Needs, and Design Opportunities". In: CHI '20. Honolulu, HI, USA: Association for Computing Machinery, 2020, pp. 1–12. isbn: 9781450367080. doi: 10.1145/3313831.3376729. url: https://doi.org/10.1145/3313831.3376729.