

Data Dunkers Teacher's Guide

Care to comment? Get access here: <https://bit.ly/3IVD68Y>

Welcome to the Data Dunkers Teacher's Guide for Incorporating Jupyter Notebooks into Data Analysis Education, sponsored by Pascal Siakam and in partnership with Dell Canada.

This guide is designed to accompany the lessons available at [Data Dunkers GitHub Repository](#), providing educators with a structured approach to teaching data analysis through hands-on learning. Using Jupyter Notebooks, a powerful tool for data science and analysis, our goal is to offer a comprehensive resource that not only teaches technical skills but also fosters critical thinking and problem-solving abilities among students. With a focus on using data from Pascal Siakam and the Toronto Raptors, we bring real-world relevance to lessons, enriching the educational experience.

Through this guide, teachers will navigate a structured series of lessons that span from the basics of Jupyter Notebooks to advanced data visualization techniques. We start with an introduction to Jupyter Notebooks, familiarizing both educators and students with the environment where their data analysis journey will unfold. From there, we delve into the various stages of data analysis:

- **Getting the Data:** Students learn to gather data from a wide array of sources, including direct input within notebooks, as well as external sources such as CSV files, Excel spreadsheets, webpages, and Google Sheets. Special emphasis is placed on the importance of using real-world data, particularly from Pascal Siakam's career statistics and Toronto Raptors game data, to enhance the learning experience.
- **Processing the Data:** This section equips students with essential data manipulation techniques using the pandas library. Students will learn to select, filter, sort, and enrich datasets, preparing them for insightful analysis. These skills are crucial for cleaning and structuring raw data into a format that is ready for exploration. Practical exercises will include working with basketball data to create meaningful analyses that reflect real-world sports analytics.
- **Plotting the Data:** The final stage introduces students to data visualization, teaching them to create meaningful representations of their data. Through detailed lessons on bar graphs, scatter plots, pie charts, and histograms, students will learn to select the appropriate plot types that best represent their analysis findings, using both pandas and Plotly. The data visualization section will focus on showcasing how to convey sports performance and team dynamics through engaging and informative plots.

Each section of this guide is carefully crafted to provide a step-by-step approach to teaching these concepts, integrating Pascal Siakam's and the Toronto Raptors' data to make learning more relatable and engaging. The guide includes objectives, materials needed, pre-lesson preparations, detailed lesson plans, and extension activities. Furthermore, it offers strategies for assessment and feedback, ensuring that students not only learn to perform data analysis but also understand the impact of their findings on real-world scenarios, such as sports analytics.

This guide is more than just a manual; it is a pathway to empowering students with the skills to turn data into insights. By the end of this course, students will be well-versed in the use of Jupyter Notebooks for data analysis, equipped with a solid foundation in data manipulation and visualization techniques, and ready to tackle real-world data challenges, particularly in the sports analytics domain.

Contents

[License](#)

[Overview](#)

[Inviting Your Feedback](#)

[Overview of Sections](#)

[Intro to Jupyter Notebooks](#)

[Getting the Data \(Input Stage\)](#)

[Processing the Data \(Process Stage\)](#)

[Plotting the Data \(Output Stage\)](#)

License

This resource is licensed under the terms of “Creative Commons Attribution-ShareAlike 4.0 International”.

You are free to:

- Share — copy and redistribute the material in any medium or format for any purpose, even commercially.
- Adapt — remix, transform, and build upon the material for any purpose, even commercially.
- The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

- Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
- ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.
- No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

[[Back to Contents](#)]

Overview

The "Data-Dunkers-Teacher's Guide" document is structured to assist educators in [teaching data analysis through Jupyter Notebooks](#), focusing on real-world data from basketball, specifically Pascal Siakam and the Toronto Raptors. It covers the following main areas:

1. **License and Feedback Mechanism:** The guide begins by outlining the Creative Commons license under which it is shared, inviting feedback from educators and students to improve the resource. This section encourages community participation to refine and enhance the guide's quality.
2. **Overview:** This section provides a structured format for the guide, detailing the objectives, materials needed, pre-lesson preparation, detailed lesson plans, assessment and feedback methods, extension activities, reflection, and additional resources for each segment of the lesson. It aims to offer a comprehensive approach to teaching Jupyter Notebooks and data analysis concepts.
3. **Introduction to Jupyter Notebooks:** The guide introduces educators and students to Jupyter Notebooks, covering basics like Markdown, code cells vs. Markdown cells, executing a simple "Hello World" program, and basic Python syntax. It includes objectives, materials needed, pre-lesson preparations, lesson plans, assessment and feedback, extension activities, reflection, and additional resources.
4. **Getting the Data (Input Stage):** This segment focuses on introducing students to various methods of data importation into Jupyter Notebooks, including from within the notebook, CSV files, Excel files, webpages, and Google Sheets. It emphasizes the hands-on experience with data importation, setting a foundation for advanced analysis tasks.
5. **Processing the Data (Process Stage):** Here, the focus shifts to data processing, where students learn essential techniques using the pandas library for data manipulation. The aim is to prepare raw data for analysis by cleaning, organizing, and transforming it.
6. **Plotting the Data (Output Stage):** This final stage introduces students to data visualization, teaching them to create meaningful representations of their data with various types of plots. The section aims to enable students to select appropriate visualization techniques to convey their analysis findings effectively.

Each section of the guide is crafted to provide a step-by-step approach to teaching concepts, integrating real-world data to make learning more relatable and engaging. The guide concludes with detailed instructions for creating visualizations, assessing understanding, and suggesting further exploration through extension activities and additional resources.

[[Back to Contents](#)]

Inviting Your Feedback

Here's the revised section with the addition of instructions on how to request permission to comment on the Google Doc:

"We highly value the insights and feedback of our community. Your contributions help us refine and enhance the quality of this educational resource, ensuring it remains relevant, accurate, and engaging for all users. If you have suggestions, comments, or would like to propose additions to the content, we encourage you to actively participate by making comments directly in the original Google Doc.

To provide your feedback:

1. Navigate to the [Google Doc](#) version of this resource.
2. If you do not have immediate permission to comment, click on the **"Request edit access"** button typically located at the top-right corner of the document.
3. Fill in the short form with your request. Please specify that you are seeking permission to comment.
4. Once access is granted, you will receive a notification. You can then add your comments directly on the text where you think improvements can be made.

Your comments are crucial to the ongoing improvement of this guide. Whether you're an educator with insights on how the material could be better integrated into the lessons, a student with ideas on making the guide more accessible, or simply someone passionate about data analysis education, we're looking forward to hearing from you.

[[Back to Contents](#)]

Overview of Sections

The sections in the Teacher's Guide follow this structured format:

1. **Introduction**
 - Brief overview of the section's purpose and goals.
 - Highlighting the significance of the technology or concept being introduced.
2. **Objectives**
 - A clear list of learning outcomes detailing what students will understand or be able to do by the end of the section.
3. **Materials Needed**
 - Specifies the resources and tools required for the lesson, ensuring both teacher and students are prepared.
4. **Pre-Lesson Preparation**
 - Steps for teachers to follow before the lesson, including ensuring access to necessary technology and preparing teaching materials.
5. **Lesson Plan**
 - Detailed breakdown of the lesson into timed segments, each focusing on a specific topic or activity.
 - Includes explanations, demonstrations, and guided practice activities to facilitate learning.
 - Sections are clearly defined, with specific objectives for each, allowing for structured progression through the material.
6. **Assessment and Feedback**
 - Methods for evaluating student understanding and providing feedback, including quizzes and review of practice activities.
7. **Extension Activities**
 - Suggestions for additional projects or exercises that extend learning beyond the basic lesson plan.
8. **Reflection**
 - Encourages students to think about what they've learned and how it applies to future projects or learning.
9. **Additional Resources**
 - Links and references to further reading, tutorials, and documentation to deepen understanding of the subject matter.

This structured format ensures that each section of the Teacher's Guide is comprehensive, providing a roadmap for educators to deliver effective and engaging lessons on Jupyter Notebooks and data analysis concepts.

Note that this document is a work-in-progress. Educators and students are encouraged to [provide feedback](#) to help make this document better for others.

[[Back to Contents](#)]

Intro to Jupyter Notebooks

This guide aims to provide teachers with a comprehensive framework to introduce Jupyter Notebooks to their students, ensuring a smooth and engaging learning experience.

Objectives:

- Understand what Jupyter Notebooks are and their importance in data science and analysis.
- Learn the basics of Markdown and how to use it for documentation within notebooks.
- Distinguish between code cells and Markdown cells.
- Execute a simple "Hello World" program.
- Familiarize students with basic Python syntax and running code in Jupyter Notebooks.

Materials Needed:

- Access to a computer with internet connectivity.
- Jupyter Notebook installed or access to an online platform like Google Colab or Callysto.
- Sample Jupyter Notebook file for demonstration.

Pre-Lesson Preparation:

- Ensure all students have access to Jupyter Notebook, either through installation or online platforms.
- Prepare a sample notebook that includes Markdown examples, a "Hello World" program, and basic Python syntax examples.
- Review the basics of Python programming if not already familiar.

Lesson Plan:

- 1. Introduction to Jupyter Notebooks (15 minutes)**
 - Discuss the history and development of Jupyter Notebooks.
 - Explain the acronym Jupyter and its significance.
 - Showcase the interface and main features of Jupyter Notebooks.
- 2. Markdown Basics (20 minutes)**
 - Demonstrate how to create Markdown cells.
 - Show examples of formatting text in Markdown (bold, italics, lists, links, and images).
 - Practice activity: Have students create a Markdown cell with formatted text including a link and an image.
- 3. Code Cells vs. Markdown Cells (15 minutes)**
 - Explain the difference between code cells and Markdown cells.
 - Demonstrate how to run Python code in a code cell.
 - Practice activity: Students write and execute a simple Python code snippet.
- 4. Hello World Program (10 minutes)**
 - Guide students through writing and executing a "Hello World" program in a code cell.

- Discuss the significance of the "Hello World" program as a programming tradition.
- 5. **Basic Python Syntax** (20 minutes)
 - Briefly explain basic Python syntax that will be commonly used in the course.
 - Demonstrate arithmetic operations, variables, and basic data types in Python.
 - Practice activity: Students perform simple arithmetic operations and print the results.
- 6. **Running Code in Jupyter Notebook** (15 minutes)
 - Show how to run code in Jupyter Notebook and interpret the output.
 - Explain error messages and basic troubleshooting steps.
 - Practice activity: Students modify their "Hello World" program to include basic Python syntax and run it.

Assessment and Feedback:

- Conduct a short quiz to assess understanding of Jupyter Notebooks, Markdown, and basic Python syntax.
- Review the practice activities with the class, providing feedback and clarifications as needed.

Extension Activities:

- Assign students to create a Jupyter Notebook that includes a brief bio about themselves using Markdown formatting and basic Python code to calculate their birth year based on their age.

Reflection:

- Ask students to reflect on what they learned about Jupyter Notebooks and how they envision using them in future projects.

Additional Resources:

- Links to online tutorials and documentation for deeper exploration of Jupyter Notebooks and Markdown.

[[Back to Contents](#)]

Getting the Data (Input Stage)

This guide provides teachers with a structured approach to introduce students to the crucial skill of gathering and analyzing data from a diverse range of sources using Jupyter Notebooks. Through hands-on practice, students will learn to comfortably import and manipulate data from within the notebook itself, CSV files, Excel spreadsheets, webpages, and Google Sheets. By working with these varied data sources, students will lay a solid foundation for advanced analysis tasks. This comprehensive exposure ensures that students gain practical experience and confidence in handling data from multiple origins, equipping them with the necessary skills for real-world data analysis scenarios.

Objectives:

- Learn various methods to import data into Jupyter Notebooks.
- Understand how to import data from within the notebook, CSV files, Excel files, webpages, and Google Sheets.
- Practice loading data into pandas DataFrames for analysis.
- Familiarize with the structure and manipulation of imported data.

Materials Needed:

- Computers with internet access and Jupyter Notebook installed or access to an online platform like Google Colab.
- Sample datasets in CSV, Excel format, and links to webpages and Google Sheets with accessible data.
- Access to pandas documentation and resources.

Pre-Lesson Preparation:

- Ensure all datasets are accessible and permissions for Google Sheets are correctly set for public access.
- Prepare examples of datasets in various formats (CSV, Excel, webpage table, Google Sheet).
- Review pandas library functions for importing data.

Lesson Plan:

1. **Introduction to Data in Jupyter Notebooks** (10 minutes)
 - Discuss the significance of data analysis and how data can be sourced from various formats.
 - Overview of pandas library for data manipulation and analysis.
2. **Getting Data From Within the Notebook** (20 minutes)
 - Demonstrate defining data directly within a notebook using Python lists and dictionaries.
 - Practice activity: Students define their own simple dataset within a notebook.
3. **Getting Data from a CSV File** (20 minutes)

- Explain what CSV files are and their common use.
- Demonstrate how to load a CSV file into a pandas DataFrame.
- Practice activity: Students load a provided CSV file and display its contents.
- 4. **Getting Data from an Excel File** (20 minutes)
 - Discuss the use of Excel files for data storage.
 - Show how to read Excel files into pandas DataFrames.
 - Practice activity: Students import data from an Excel file and explore its structure.
- 5. **Getting Data from a Webpage** (20 minutes)
 - Explain how data can be scraped from webpages.
 - Demonstrate using pandas to read tables directly from a webpage.
 - Practice activity: Students extract a table from a provided webpage URL.
- 6. **Getting Data from a Google Sheet** (20 minutes)
 - Cover how to share Google Sheets for public access.
 - Show how to import data from Google Sheets into pandas DataFrames.
 - Practice activity: Students import data from a shared Google Sheet.

Assessment and Feedback:

- Create an assignment where students must gather data from a source not covered in class, import it into a Jupyter Notebook, and perform basic analysis (e.g., count, sum, average).
- Provide feedback on students' practice activities and assignments, highlighting efficient data loading techniques and common pitfalls.

Extension Activities:

- Challenge students to find their own external data source, import it into a Jupyter Notebook, and prepare a short presentation on their findings.
- Introduce basic data cleaning techniques using pandas to prepare the imported data for analysis.

Reflection:

- Ask students to discuss the challenges they faced when importing data from different sources and how they overcame them.
- Encourage students to reflect on the importance of being able to import and manipulate data from various sources in data analysis projects.

Additional Resources:

- Provide links to advanced tutorials on data import and manipulation with pandas.
- Share resources on data cleaning and preliminary data analysis techniques.

[[Back to Contents](#)]

Processing the Data (Process Stage)

This guide assists teachers in structuring comprehensive lessons on data processing, crucial for enabling students to effectively clean, organize, and transform data for analysis. Focusing on hands-on activities within Jupyter Notebooks using the pandas library, students will gain practical experience in key data processing techniques. The lessons specifically cover manipulating columns, filtering data based on criteria, sorting datasets in various orders, and adding new columns through calculations. By mastering these fundamental processing tasks, students will be well-prepared to undertake deeper analysis and extract meaningful insights from their datasets.

Objectives:

- Master fundamental data processing techniques using pandas in Jupyter Notebooks.
- Learn how to manipulate columns, filter data, sort datasets, and add new columns to DataFrames.
- Develop skills in preparing and transforming raw data into a more analyzable format.

Materials Needed:

- Computers with Jupyter Notebook installed or access to Google Colab.
- Datasets previously imported during the "Getting the Data" stage.
- Access to pandas documentation for reference.

Pre-Lesson Preparation:

- Review pandas methods for manipulating DataFrames, including selecting columns, filtering rows, sorting data, and creating new columns.
- Prepare examples demonstrating each data processing technique.

Lesson Plan:

1. **Review of Imported Data** (10 minutes)
 - Quick review of methods to import data into Jupyter Notebooks.
 - Overview of the pandas DataFrame structure.
2. **Playing with Columns** (20 minutes)
 - Demonstrate how to select specific columns from a DataFrame.
 - Show how to rename columns for clarity or consistency.
 - Practice activity: Students select and rename columns in their DataFrame.
3. **Filtering Our Data** (20 minutes)
 - Teach students how to apply conditions to filter rows in a DataFrame.
 - Explain logical operators and their use in filtering.
 - Practice activity: Students filter their dataset based on specific criteria (e.g., numeric values greater than a threshold).
4. **Sorting Our Data** (20 minutes)
 - Discuss the importance of sorting data for analysis.
 - Demonstrate sorting DataFrames by one or more columns.

- Practice activity: Students sort their datasets in ascending and descending order based on different columns.
- 5. **Adding New Columns** (20 minutes)
 - Show how to perform operations on existing columns to create new ones.
 - Explain the use of lambda functions for more complex operations.
 - Practice activity: Students add new calculated columns to their DataFrame, such as calculating percentages or aggregating values.

Assessment and Feedback:

- Assign a mini-project where students must clean and preprocess a provided dataset using the techniques learned. This project should include selecting relevant columns, filtering data, sorting the dataset, and adding at least one new calculated column.
- Offer personalized feedback on these mini-projects, focusing on the effective use of pandas functions and the logic behind data filtering and sorting choices.

Extension Activities:

- Introduce more advanced data manipulation concepts like groupby for aggregation and pivot tables.
- Encourage students to apply these data processing techniques to their own project or a dataset of interest.

Reflection:

- Facilitate a discussion on how data processing steps can impact the outcome of data analysis projects.
- Encourage students to share any difficulties they encountered during the practice activities and how they resolved them.

Additional Resources:

- Provide links to advanced pandas documentation and tutorials for further learning.
- Share articles and case studies illustrating the impact of data preprocessing on data analysis outcomes.

[[Back to Contents](#)]

Plotting the Data (Output Stage)

This guide equips teachers to provide an in-depth lesson on data visualization, allowing students to effectively transform their processed data into compelling visual narratives. By engaging in practical activities and exploring a variety of plotting techniques, students will become adept at choosing and crafting visualizations that vividly convey their data analysis insights. Specifically, the lessons cover the creation of bar graphs, scatter plots, pie charts, and histograms using pandas and Plotly in Jupyter Notebooks. These plot types are chosen for their broad applicability in representing different data relationships and distributions, providing students with a versatile toolkit for data presentation and storytelling.

Objectives:

- Introduce students to data visualization concepts using pandas and Plotly in Jupyter Notebooks.
- Teach students how to create various types of plots to represent data visually, including bar graphs, scatter plots, pie charts, and histograms.
- Enable students to choose appropriate visualization techniques to convey data insights effectively.

Materials Needed:

- Computers with Jupyter Notebook installed or access to Google Colab.
- Datasets prepared and processed from previous stages.
- Access to Plotly and pandas documentation for reference.

Pre-Lesson Preparation:

- Ensure all students are familiar with the basics of pandas DataFrames from previous lessons.
- Prepare examples of datasets with different types of data suitable for various plots.
- Review the syntax and options available in Plotly for creating different types of charts.

Lesson Plan:

1. **Introduction to Data Visualization** (15 minutes)
 - Discuss the importance of data visualization in data analysis and communication.
 - Introduce Plotly as a powerful tool for creating interactive plots in Jupyter Notebooks.
2. **Creating Bar Graphs** (20 minutes)
 - Explain when to use bar graphs and demonstrate creating both vertical and horizontal bar graphs.
 - Practice activity: Students create bar graphs to visualize data from their datasets, such as sales data or survey responses.
3. **Creating Scatter Plots** (20 minutes)

- Teach the significance of scatter plots in identifying relationships between variables.
- Demonstrate how to create scatter plots and add trend lines.
- Practice activity: Students generate scatter plots to explore correlations in their datasets, such as relationship between age and spending.
- 4. **Creating Pie Charts** (20 minutes)
 - Discuss the use of pie charts to represent parts of a whole.
 - Show how to create pie charts with Plotly, emphasizing customizations like labels and colors.
 - Practice activity: Students create pie charts to display categorical distributions, such as market share or survey results.
- 5. **Creating Histograms** (20 minutes)
 - Introduce histograms for visualizing distributions of data.
 - Demonstrate how to adjust bins and interpret histogram shapes.
 - Practice activity: Students plot histograms to analyze distributions in their datasets, such as product prices or customer age groups.

Assessment and Feedback:

- Assign a project where students must use their processed datasets to create a comprehensive report including at least three different types of plots that highlight key insights from the data.
- Provide feedback on the choice of plots, clarity of visualization, and the insights conveyed through the visualizations.

Extension Activities:

- Challenge students to explore more advanced plotting features in Plotly, such as 3D plots, heatmaps, or interactive time series.
- Encourage students to integrate plots into a coherent narrative or presentation that tells a story with their data.

Reflection:

- Facilitate a discussion on how different plots can convey different insights and how the choice of plot affects the audience's understanding of the data.
- Ask students to share what they found most challenging about creating visualizations and how they overcame those challenges.

Additional Resources:

- Provide links to galleries of Plotly charts for inspiration.
- Share resources on principles of effective data visualization and design.

[[Back to Contents](#)]