



Guide to NumPy, pandas, and Data Visualization

A collection of tutorials, practice problems, cheat sheets, guided projects, and frequently asked questions.

Main Guide Page - **You are here**

[Tutorial 1: Introduction to NumPy and pandas for Data Analysis](#)

[Tutorial 2: Introduction to Data Visualization in Python](#)

[Tutorial 3: Data Storytelling and Information Design](#)

[NumPy Cheat Sheet and PDF Download](#)

[Pandas Cheat Sheet and PDF Download](#)

[Matplotlib Cheat Sheet and PDF Download](#)

This comprehensive guide offers everything you need for an introduction to NumPy, pandas, and data visualization with tutorials that build foundational skills, practice problem exercises to reinforce learning, handy cheat sheets for quick reference, and projects that provide practice in real-world data science scenarios. The included FAQs address common challenges, making it an essential resource for anyone serious about learning Python for data science.



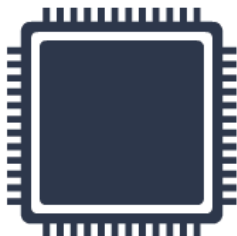
NumPy, pandas, and data visualization for data science. Check out the associated full tutorials for more details. If you're just starting out and want to actively learn these concepts directly in your browser, enroll in Dataquest's [Data Analyst in Python](#) career path for free.

1. Introduction to NumPy and pandas for Data Analysis

 NumPy ndarray

6	5
1	3
5	6
1	4
3	7
5	8
3	5
8	4

CPU



Cycles to
Complete

Output

Have you ever found yourself struggling with slow data processing or complex calculations when working with large datasets in Python? NumPy and pandas are powerful libraries that can help you overcome these challenges. By learning these tools, you can significantly speed up your data analysis tasks, handle large datasets with ease, and perform complex operations efficiently. Whether you're dealing with financial data, scientific measurements,



If you're an aspiring data scientist, analyst, or anyone who works with data in Python, mastering NumPy and pandas is essential for taking your skills to the next level. These libraries form the foundation of many data science and machine learning workflows, enabling you to manipulate, analyze, and visualize data with greater ease and efficiency. By investing time in learning these tools, you'll be better equipped to handle real-world data challenges and make data-driven decisions in your projects or career.

[Introduction to NumPy and pandas for Data Analysis](#) — here's a breakdown of what this NumPy and pandas tutorial teaches:

Lesson 1 – Introduction to NumPy

- Create and manipulate `ndarray` objects for efficient numerical operations
- Utilize vectorized operations to perform calculations on entire arrays at once
- Understand the performance benefits of NumPy over standard Python lists

Lesson 2 – Boolean Indexing with NumPy

- Apply Boolean indexing to filter data based on specific conditions
- Combine multiple conditions for complex data selection
- Use Boolean arrays to perform element-wise operations on datasets

Lesson 3 – Introduction to pandas

- Create and manipulate `DataFrame` and `Series` objects for structured data analysis
- Understand the relationship between pandas and NumPy data structures
- Load data from CSV files into pandas `DataFrame` objects

Lesson 4 – Exploring Data with pandas: Fundamentals

- Use `info()` and `describe()` methods to gain quick insights into datasets



Lesson 5 – Exploring Data with pandas: Intermediate

- Utilize `iloc` for advanced data selection based on integer positions
- Apply index alignment for efficient data manipulation across different datasets
- Use complex filtering and sorting techniques to extract specific insights

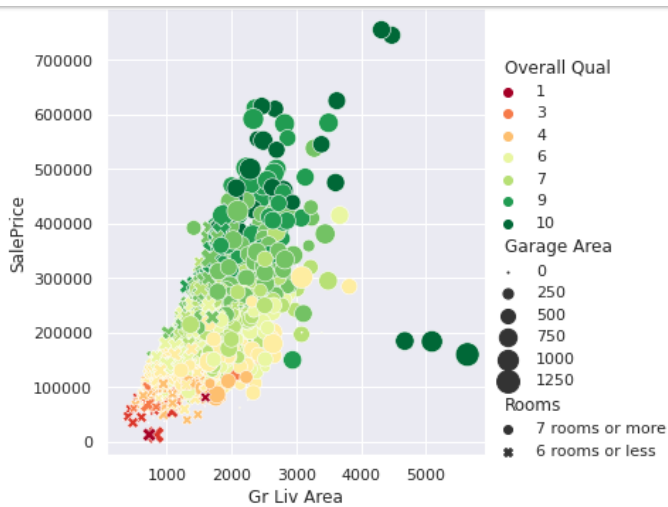
Lesson 6 – Data Cleaning Basics

- Implement techniques for cleaning and standardizing column names
- Convert data types and handle missing values in datasets
- Apply string methods for data cleaning and transformation

Guided Project: Exploring eBay Car Sales Data

- Apply NumPy and pandas skills to analyze a real-world dataset of used car listings
- Clean and preprocess data to ensure accuracy in analysis
- Create visualizations to uncover insights and identify data quality issues

By learning NumPy and pandas for data analysis, you'll be equipped with powerful tools to handle large datasets efficiently, perform complex calculations quickly, and extract meaningful insights from your data. These skills will enable you to tackle real-world data challenges with confidence, whether you're analyzing financial trends, processing scientific data, or developing machine learning models. You'll be able to clean and preprocess data more effectively, create insightful visualizations, and make data-driven decisions in your projects or career. With these foundational skills in place, you'll be well-prepared to advance your data science capabilities and take on more sophisticated analysis tasks.



Raw numbers often conceal valuable insights within datasets. Data visualization transforms these numbers into compelling visual narratives, revealing patterns and trends that might otherwise remain hidden. By creating simple time series graphs, seasonal trends will emerge from the raw data. Data visualization can transform our understanding and improve the accuracy of our analyses.

Learning data visualization is essential for anyone working with data, from analysts and scientists to business professionals. As data continues to grow in volume and complexity, the ability to create clear, insightful visualizations has become a highly sought-after skill. Employers value candidates who can effectively communicate insights through visuals, not just crunch numbers. Whether you're looking to enhance your current role or transition into a data-focused career, mastering data visualization techniques will set you apart in the job market and enable you to make more informed decisions based on your data.



Lesson 1 – Line Graphs and Time Series

- Create line graphs using `matplotlib.pyplot` to visualize trends over time
- Customize plot elements such as titles, labels, and formatting
- Compare multiple time series on a single graph to identify patterns and relationships

Lesson 2 – Scatter Plots and Correlations

- Generate scatter plots to explore relationships between two variables
- Calculate and interpret Pearson's correlation coefficient
- Understand the distinction between correlation and causation in data analysis

Lesson 3 – Bar Plots, Histograms, and Distributions

- Create bar plots to compare values across different categories
- Use histograms to visualize the distribution of numerical data
- Identify and interpret common distribution patterns (e.g., normal, uniform, skewed)

Lesson 4 – pandas Visualizations and Grid Charts

- Utilize pandas' built-in plotting functions for quick data exploration
- Create grid charts (small multiples) to compare multiple related graphs
- Apply data cleaning techniques to prepare datasets for visualization

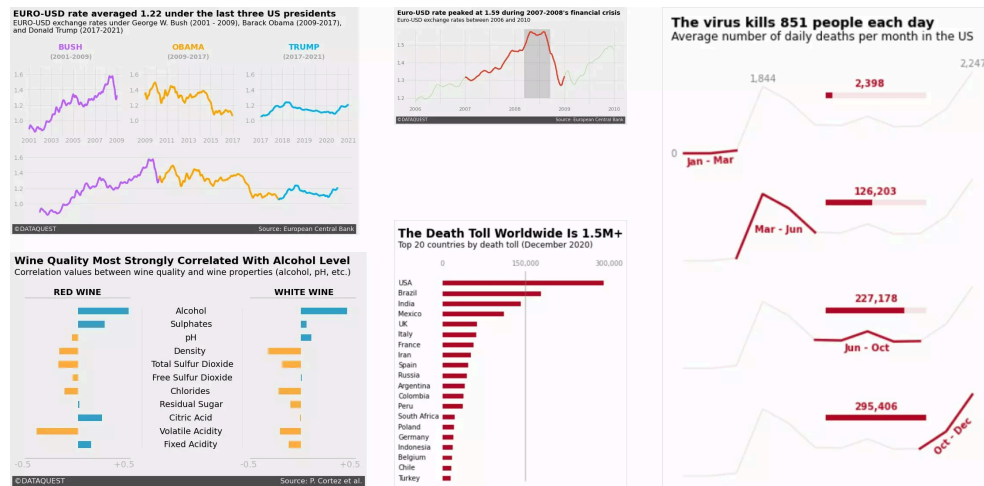
Lesson 5 – Relational Plots and Multiple Variables

- Use seaborn's `relplot()` function to create multi-variable visualizations
- Represent additional variables using color, size, and shape in scatter plots
- Interpret complex relationships between multiple variables in a single plot



using popular Python libraries like matplotlib, seaborn, and pandas to create a wide range of visualizations, from simple line graphs to complex multi-variable plots. These skills will enable you to effectively communicate your findings to both technical and non-technical audiences, making you a valuable asset in any data-driven organization. Whether you're analyzing financial trends, exploring scientific data, or optimizing business processes, the ability to create insightful visualizations will enhance your decision-making capabilities and open up new opportunities in the field of data science.

3. Telling Stories Using Data Visualization and Information Design



Standing in front of a room, watching eyes glaze over as you click through slide after slide of charts, can be an embarrassing experience for any data analyst. You know your numbers make sense, but they're just not connecting with your audience. This is where the art of data storytelling comes in. By focusing on visual design and weaving in a clear narrative, you can



Learning to tell compelling data stories is an essential skill for anyone working with data. Whether you're a data scientist presenting findings to stakeholders, a business analyst explaining trends to executives, or a researcher sharing results with the public, the ability to create clear, engaging visualizations can significantly impact how your insights are received and acted upon. This tutorial will equip you with the techniques to create professional-looking, narrative-driven visualizations using Python, helping you communicate your data more effectively and persuasively.

[Telling Stories Using Data Visualization and Information Design](#) — here's a breakdown of what this data storytelling tutorial teaches:

Lesson 1 – Design for an Audience

- Tailor visualizations to specific audience needs and knowledge levels
- Use Matplotlib's object-oriented interface for greater control over plot elements
- Apply design principles to maximize the data-ink ratio and enhance clarity

Lesson 2 – Storytelling Data Visualization

- Create multi-panel visualizations to show data progression over time
- Add context through annotations, labels, and narrative elements
- Guide viewers through data insights using visual cues and formatting

Lesson 3 – Gestalt Principles and Pre-Attentive Attributes

- Apply Gestalt principles (proximity, similarity, enclosure, connection) to organize visual information
- Utilize pre-attentive attributes like color and size to guide attention effectively
- Understand and implement visual hierarchy in data visualizations



- Customize plots to achieve a clean, professional look
- Create visually consistent and impactful data stories across multiple visualizations

Guided Project: Storytelling Data Visualization on Exchange Rates

- Clean and prepare real-world financial data for visualization
- Apply rolling means to smooth out short-term fluctuations in time series data
- Create a compelling visual narrative using exchange rate data and economic events

By completing this tutorial, you'll be able to create professional-quality data visualizations that effectively communicate complex insights. You'll learn to design charts and graphs tailored to your audience, incorporate storytelling elements to make your data more engaging, and apply psychological principles to enhance visual comprehension. These skills will enable you to present data in a way that not only informs but also inspires action, whether you're analyzing market trends, presenting research findings, or explaining complex phenomena to a general audience.

Practice Problems

Test your knowledge with the NumPy, pandas, and data visualization exercises below. For additional practice problems and real-time feedback, try our [interactive coding environment](#), great for Python practice online.



Creating a ndarray of ones

In this NumPy practice problem, you'll create a 2-dimensional array in which every entry is equal to one.

NumPy provides the `np.ones()` function that we can use to do this. It takes as input a shape and produces a ndarray with the given shape in which each entry is equal to one.

Example:

```
x = np.ones((7,))  
print(x)
```

```
[1.  1.  1.  1.  1.  1.  1.]
```

Instructions

1. Create a 2-dimensional ndarray with seven rows and five columns in which each entry is equal to one. Assign it to a variable `x`.

Hint



Answer



Practice solving this exercise using our [interactive coding environment](#) designed for Python practice online with real-time feedback.



Extracting the Even Values

In this NumPy practice problem, we want you to extract all even values from a 2-dimensional array. The result should be a 1-dimensional array that contains all even values of the provided 2-dimensional array.

Recall that the even numbers are 0, 2, 4, 6, and so on.

For example, if the 2-dimensional array was:

```
[[1 2 4]
 [3 4 6]]
```

The result would be:

```
[2 4 4 6]
```

Instructions

1. Create a 1-dimensional array named `even_values` that contains all even values from the provided 2-dimensional array `x`.



```
[ 8, 1, 5, 3, 10],  
[ 2, 6, 7, 6, 6],  
[ 7, 8, 3, 6, 7],  
[10, 7, 7, 5, 2]  
)
```

Hint

Answer

Practice solving this exercise using our [interactive coding environment](#) designed for NumPy practice online with real-time feedback.

3. Stacking Vertically 1

In this NumPy practice problem, you'll be given a list with 1-dimensional arrays. Your goal is to create a 2-dimensional array whose rows are the 1-dimensional arrays in that list.

For example, if the list was:



```
np.array([1, 2, 3, 4]),  
np.array([5, 6, 7, 8]),  
np.array([9, 10, 11, 12]),  
np.array([13, 14, 15, 16])  
]
```

We want to transform it into the following 2-dimensional array:

```
[[ 1  2  3  4]  
 [ 5  6  7  8]  
 [ 9 10 11 12]  
 [13 14 15 16]]
```

You might want to take a look at the [numpy.vstack\(\)](#) function.

Instructions

1. Use the provided list of ndarrays `lst` to create a 2-dimensional ndarray whose rows are the ndarrays in `lst`. Assign it to variable `x`.

Hint



Answer



Practice solving this exercise using our [interactive coding environment](#) designed for Python practice online with real-time feedback.



Calculating the maximum

In this NumPy practice problem, you'll calculate the maximum value of a given 2-dimensional ndarray.

Instructions

1. Calculate the maximum value of the provided ndarray `x`. Assign the answer to variable `maximum`.

```
x = np.array([
    [56, 75, 82, 18, 47, 27, 53],
    [85, 97, 48, 21, 57, 44, 20],
    [24, 40, 15, 57, 17, 93, 55],
    [68, 61, 24, 77, 90, 73, 34],
    [22, 14, 31, 34, 68, 19, 55],
    [37, 75, 64, 42, 14, 16, 98]
])
```

Hint



Answer



Practice solving this exercise using our [interactive coding environment](#) designed for Python practice online with real-time feedback.



Creating DataFrame from Dictionaries

In this pandas practice problem, you'll practice creating a DataFrame from two dictionaries having the same set of keys.

We provide you with two dictionaries. These dictionaries contain data on the life expectancy of the top 10 countries with the highest life expectancies.

The **women** dictionary contains the women's life expectancy. The **men** dictionary contains the men's life expectancy. The data was gathered from [here](#).

Instructions

1. Create a DataFrame named **life_exp_top10** with two columns **Women** and **Men** (in that order). The values in these columns should be the ones in respective dictionaries.

Hint



Answer



Practice solving this exercise using our [interactive coding environment](#) designed for Python practice online with real-time feedback.

2. Selecting the Age Column

In this pandas practice problem, you'll practice selecting a column from a DataFrame.



ages .

Hint



Answer



Practice solving this exercise using our [interactive coding environment](#) designed for Python practice online with real-time feedback.

3. List to Series Object

In this pandas practice problem, we give you a list and your goal is to create a Series object with the same values as that list.

Remember that you can create a Series object using the `pandas.Series()` constructor.

Instructions

1. Create a Series object whose values are the same as the `values` list. Assign it to a variable named `series` .

Hint



Answer



Practice solving this exercise using our [interactive coding environment](#) designed for Python practice online with real-time feedback.



Instructions to Complete

In this pandas practice problem, you'll practice selecting data using boolean indexing.

Instructions

1. Select all countries from the `countries` DataFrame such that:
 - The `Birthrate` is smaller than the `Deathrate`.
2. Assign the result to a variable named `birth_vs_death`.

Hint



Answer



Practice solving this exercise using our [interactive coding environment](#) designed for Python practice online with real-time feedback.

Data Visualization Practice Problems

1. Types of Line Graphs

In this data visualization practice problem, you'll plot a line graph for Argentina using time series data from the COVID-19 pandemic. The [dataset you're working with](#) contains daily reported cases and deaths for various countries, recorded in columns like `Date_reported`, `Country`, `Cumulative_cases`, `New_cases`, and more.

To handle the time series data properly, you'll need to convert the `Date_reported` column to datetime format before plotting the graph. This will ensure the x-axis of the plot accurately



1. Load the dataset from the provided link and assign it to a variable named `who_time_series`.
2. Convert the `Date_reported` column to datetime format using `pd.to_datetime()`.
3. Isolate the data for Argentina in a variable named `argentina`.
4. Plot a line graph of Argentina's `Cumulative_cases`.
5. Include the title "`Argentina: Cumulative Reported Cases`", an x-axis label "`Date`", and a y-axis label "`Number of Cases`".

Hint



Answer



Practice solving this exercise using our [interactive coding environment](#) designed for Python practice online with real-time feedback.

2. Scatter Plots

In this data visualization practice problem, you'll plot a scatter plot to visualize the relationship between wind speed and the number of bikes rented. The [dataset you're working with](#) contains daily data for a bike-sharing system, with columns like `dteday` (the date), `temp` (air temperature), `cnt` (number of bikes rented), and `windspeed` (wind speed).

Scatter plots are great for visualizing the relationship between two numerical variables. In this exercise, you'll create a scatter plot to see how wind speed impacts the number of bikes



Instructions

1. Load the dataset from the provided link and assign it to a variable named `bike_sharing`.
2. Convert the `dteday` column to datetime format using `pd.to_datetime()`.
3. Create a scatter plot with the `windspeed` column on the x-axis and the `cnt` column on the y-axis.
4. Label the x-axis as `'Wind Speed'` and the y-axis as `'Bikes Rented'`.
5. Display the plot using `plt.show()`.

Hint



Answer



Practice solving this exercise using our [interactive coding environment](#) designed for Python practice online with real-time feedback.

3. Histograms

In this data visualization practice problem, you'll generate histograms to explore the distribution of the `casual` and `cnt` columns from a bike-sharing dataset. The [dataset you're working with](#) contains daily data for a bike-sharing system, with columns like `dteday` (the date), `cnt` (total number of bikes rented), and `casual` (number of casual users, i.e., non-subscribers).

Histograms are useful for visualizing the distribution of numerical data by grouping values into intervals (or bins). In this exercise, you'll create histograms to observe how the number



Instructions

1. Load the dataset from the provided link and assign it to a variable named `bike_sharing`.
2. Convert the `dteday` column to datetime format using `pd.to_datetime()`.
3. Generate a histogram for the `casual` column using `plt.hist()`.
4. Display the plot using `plt.show()`.
5. Generate another histogram for the `cnt` column.
6. Compare the shape of the `casual` histogram with the `cnt` histogram. What differences do you notice?

Hint

Answer

Practice solving this exercise using our [interactive coding environment](#) designed for Python practice online with real-time feedback.

4. Pandas Visualization Methods

In this data visualization practice problem, you'll generate a histogram to visualize traffic slowness data using Pandas' built-in plotting capabilities. The [dataset you're working with](#) contains traffic data for São Paulo, including a column `Slowness in traffic (%)` that measures the percentage of slowness in traffic at different times.

Before you can create the histogram, you need to clean the data by adjusting the format of the `Slowness in traffic (%)` column. After that, you'll generate the histogram and add customizations such as a title and an x-axis label.



formatting issues that need to be cleaned up.

2. Clean the **Slowness in traffic (%)** column so that it can be used for numerical analysis.
3. Create a histogram for the **Slowness in traffic (%)** column.
4. Add a title that describes the distribution of the data and label the x-axis appropriately.
5. Display the plot.

Hint

Answer

Practice solving this exercise using our [interactive coding environment](#) designed for Python practice online with real-time feedback.

Cheat Sheets

NumPy Cheat Sheet

Check out our comprehensive [NumPy Cheat Sheet](#) that provides a quick reference for essential data manipulation commands. You can also [download](#) the NumPy Cheat Sheet as a PDF.

Pandas Cheat Sheet

Check out our comprehensive [Pandas Cheat Sheet](#) that provides a quick reference for essential data manipulation commands. You can also [download](#) the Pandas Cheat Sheet as a PDF.



essential essential plotting functions in matplotlib. You can also [download](#) the Matplotlib Cheat Sheet as a PDF.

Practice

The best way to get NumPy, pandas, and data visualization practice is to work on a real world challenge in the form of projects. Use these Dataquest guided projects to test your skills and show off your knowledge to potential employers by including them in your portfolio.

1. Exploring eBay Car Sales Data

Difficulty Level: **Beginner**

Overview

In this [beginner-level guided project](#), you'll analyze a dataset of used car listings from eBay Kleinanzeigen, a classifieds section of the German eBay website. Using Python and pandas, you'll clean the data, explore the included listings, and uncover insights about used car prices, popular brands, and the relationships between various car attributes. This project will strengthen your data cleaning and exploratory data analysis skills, providing valuable experience in working with real-world, messy datasets.

Tools and Technologies

- Python
- Jupyter Notebook
- NumPy
- pandas



and have experience with:

- Loading and inspecting data using pandas
- Cleaning column names and handling missing data
- Using pandas to filter, sort, and aggregate data
- Creating basic visualizations with pandas
- Handling data type conversions in pandas

Step-by-Step Instructions

1. Load the dataset and perform initial data exploration
2. Clean column names and convert data types as necessary
3. Analyze the distribution of car prices and registration years
4. Explore relationships between brand, price, and vehicle type
5. Investigate the impact of car age on pricing

Expected Outcomes

Upon completing this project, you'll have gained valuable skills and experience, including:

- Cleaning and preparing a real-world dataset using pandas
- Performing exploratory data analysis on a large dataset
- Creating data visualizations to communicate findings effectively
- Deriving actionable insights from used car market data

Relevant Links and Resources

- [Example Solution Code](#)
- [Original eBay Kleinanzeigen Dataset on Kaggle](#)



Overview

In this [beginner-level guided project](#), you'll analyze a dataset of westbound traffic on the I-94 Interstate highway between Minneapolis and St. Paul, Minnesota. Using Python and popular data visualization libraries, you'll explore traffic volume patterns to identify indicators of heavy traffic. You'll investigate how factors such as time of day, day of the week, weather conditions, and holidays impact traffic volume. This project will enhance your skills in exploratory data analysis and data visualization, providing valuable experience in deriving actionable insights from real-world time series data.

Tools and Technologies

- Python
- Jupyter Notebook
- pandas
- Matplotlib
- seaborn

Prerequisites

To successfully complete this project, you should be comfortable with [data visualization in Python](#) techniques and have experience with:

- Data manipulation and analysis using pandas
- Creating various plot types (line, bar, scatter) with Matplotlib
- Enhancing visualizations using seaborn
- Interpreting time series data and identifying patterns
- Basic statistical concepts like correlation and distribution



2. Visualize traffic volume patterns over time using line plots
3. Analyze traffic volume distribution by day of the week and time of day
4. Investigate the relationship between weather conditions and traffic volume
5. Identify and visualize other factors correlated with heavy traffic

Expected Outcomes

Upon completing this project, you'll have gained valuable skills and experience, including:

- Creating and interpreting complex data visualizations using Matplotlib and seaborn
- Analyzing time series data to uncover temporal patterns and trends
- Using visual exploration techniques to identify correlations in multivariate data
- Communicating data insights effectively through clear, informative plots

Relevant Links and Resources

- [Example Solution Code](#)
- [Original Metro Interstate Traffic Volume Data Set](#)

3. Storytelling Data Visualization on Exchange Rates

Difficulty Level: **Beginner**

Overview

In this [beginner-level guided project](#), you'll create a storytelling data visualization about Euro exchange rates against the US Dollar. Using Python and Matplotlib, you'll analyze historical exchange rate data from 1999 to 2021, identifying key trends and events that have shaped the Euro-Dollar relationship. You'll apply data visualization principles to clean data, develop a narrative around exchange rate fluctuations, and create an engaging and informative visual



Tools and Technologies

- Python
- Jupyter Notebook
- pandas
- Matplotlib

Prerequisites

To successfully complete this project, you should be familiar with [storytelling through data visualization](#) techniques and have experience with:

- Data manipulation and analysis using pandas
- Creating and customizing plots with Matplotlib
- Applying design principles to enhance data visualizations
- Working with time series data in Python
- Basic understanding of exchange rates and economic indicators

Step-by-Step Instructions

1. Load and explore the Euro-Dollar exchange rate dataset
2. Clean the data and calculate rolling averages to smooth out fluctuations
3. Identify significant trends and events in the exchange rate history
4. Develop a narrative that explains key patterns in the data
5. Create a polished line plot that tells your exchange rate story

Expected Outcomes

Upon completing this project, you'll have gained valuable skills and experience, including:

- Crafting a compelling narrative around complex financial data



- Applying color theory and typography to enhance visual communication

Relevant Links and Resources

- [Example Solution Code](#)
- [ECB Euro reference exchange rate: US dollar](#)

Frequently Asked Questions

What is NumPy?	▼
How do I round each number in a NumPy array?	▼
What does <code>as</code> mean in <code>import numpy as np</code> ?	▼
How can I check if each number is even in NumPy?	▼
How do I create a NumPy random number generator?	▼
What is pandas and how does it complement NumPy?	▼
What is a pandas DataFrame?	▼
What is a pandas Series and how does it relate to a DataFrame?	▼



What is data visualization and why is it important in data analysis?



What are common data visualization tools in Python?



What is data storytelling and how does it improve data presentations?



What are Gestalt principles in data visualization?



What are pre-attentive attributes in data visualizations?



How can I improve my skills in NumPy, pandas, and data visualization?



About the author

Mike Levy

Mike is a life-long learner who is passionate about mathematics, coding, and teaching. When he's not sitting at the keyboard, he can be found in his garden or at a natural hot spring.



[Terms of Use](#)
[Privacy Policy](#)



About

[For Business](#)
[For Educators](#)
[About Dataquest](#)
[Learner Stories](#)
[Contact Us](#)
[Partnership Programs](#)
[Sitemap](#)

Career Paths

[Data Scientist](#)
[Data Engineer](#)
[Data Analyst Python](#)
[Data Analyst R](#)
[Business Analyst Power BI](#)
[Business Analyst Tableau](#)
[Junior Data Analyst](#)



[AI Courses](#)

[Machine Learning Courses](#)

[Deep Learning Courses](#)

[Excel Courses](#)

[Statistics Courses](#)

Explore

[Course Catalog](#)

[Projects](#)

[Teaching Method](#)

[Project-first Learning](#)

[How to Learn Python](#)

[The Dataquest Download](#)