**AI Documentation for my mini-Project 3:**

**Entry#1**

In your final project report and portfolio, we will be expecting to see the following:

● Well-documented scripts (with comments throughout) to filter the data, analyse it and produce visualisations

● Evidence that you understand the methods that were used to produce your chosen datasets (measuring article lengths, n-gram frequencies, TF-IDF and topic modelling)

● Understanding of the appropriate use of a visualisation to assert a point

● Appropriate formatting of a visualisation to assert the point (lack of clutter, clear labels, and clear annotation where needed)

● Use of the datasets holistically to make assertions about the article corpus (you will need to work together as a group to make these observations) Python scripts The python scripts that you write should demonstrate a thorough understanding of pandas and plotly and the principles of data exploration and visualisation. Remember that there are two stages to working with data like this: exploration and presentation. You will likely begin by creating a script that explores the data (what are the column names? how many rows are there? what happens if I apply this filter?) and that tests different visualisations for the data. This is the exploration stage. Using the knowledge you gained during the exploration stage, you will write a script that produces the final visualisation. We expect to see evidence of this in the final scripts that you produce (it is likely that for each task, you will have two scripts - one for exploration and one for presentation). Every team member must work on a script and produce a visualisation and each team member must work with a different data set / csv file (you should not produce all visualisations from just one data set). If a team chooses to produce a new data frame using the articles (and to go for those bonus points, see above), then it is highly recommended that they work on the task of producing the new data set together, rather than leaving it to one team member; the team will still need to produce one visualization per team member. Some concrete examples of what we expect to see in the final portfolio:

● Team of two: two scripts to analyse two different data frames and produce at least two visualizations

Team four: four scripts to analyse four different data frames and produce at least four visualisations If one team member fails to produce a script, then they will fail the assignment (as there is no evidence that they have participated in the project). The project tasks It is entirely your choice what data sets you use and how you use and visualise them, but you will be assessed on your understanding of the methods used to produce the data and of the appropriate kind of visualisation to use for the chosen data set The data folder contains a series of csv files that can be loaded in Python using pandas. Each data frame has been produced using different text analysis methods:

● counting the lengths of the articles, with separate tables grouping lengths by year and month. Documentation of the data frames, the way they have been produced and some of their shared field names is provided in the readme file for the data folder. Remember that this documentation is not exhaustive. It is your job to use what you have learned in pandas to understand the nature of each dataset (df.columns will be useful for you to explore the columns of each data frame) Data The portfolio repository contains all of the data that you will need for the project, including the original articles, should you wish to perform further analysis on the articles themselves. Do not generate script but guide me in a simple way how should i start my project work considering the above text as guide?

**ChatGPT said:**

Great — you're about to begin a **team-based final project** that involves analyzing and visualizing textual data using Python (especially with pandas and plotly). Based on your instructions, here's a **simple, step-by-step guide** to help you **start and structure your project**:

**🧭 STEP 1: Understand the Overall Goal**

You need to:

1. Explore and understand different **CSV datasets** in your project folder.
2. Choose one dataset **per team member**.
3. For each dataset:
   * Write **two Python scripts**: one for *exploration*, one for *presentation*.
   * Create **one final visualisation** (per person).
4. Show clear understanding of the **text analysis method** used (e.g., n-gram, TF-IDF).
5. Use **clear, well-formatted graphs** with labels and annotations.
6. Collaborate to make **holistic observations** about the whole article corpus.

**🗃 STEP 2: Explore the Data Folder**

Open the data folder and look for CSVs. You’ll likely find datasets based on:

* Article lengths (by month or year)
* N-gram frequencies (1-grams, 2-grams, etc.)
* TF-IDF scores
* Topic model results

Also look for a README file — this will explain how some of the data frames were made.

🛠️ **Your task now:**

* Pick one CSV to explore first.
* Load it in a Jupyter notebook or Python script using pd.read\_csv().
* Run df.head() and df.columns to understand what’s inside.
* Think: "What kind of question could I ask using this dataset?"

**🔍 STEP 3: Do the Exploration (Script 1)**

This is your **trial-and-error phase**, not the polished final product. You should:

* Print summaries: df.shape, df.describe(), df.value\_counts()
* Test filters: df[df['year'] == 2023]
* Try basic plots: bar charts, line graphs, histograms using plotly.express

💡 **Goal**: Learn what’s possible and interesting in the data.

📝 Document everything with **comments** so your reasoning is clear:

python

CopyEdit

# Filter the data to show article length distribution in 2023

**🎯 STEP 4: Design the Final Visualisation (Script 2)**

Based on what you found in exploration, build your **clean final output**:

* Create a clear, clutter-free graph
* Add **title, axis labels, annotations**
* Choose the right type of graph:
  + Line graph: trends over time
  + Bar chart: comparing values
  + Heatmap: topic distributions
* Make the visualisation **support a point** (don’t just plot randomly!)

✍️ Script 2 should be:

* Clean and well-commented
* Focused on **presenting one idea clearly**

**🧑‍🤝‍🧑 STEP 5: Coordinate with Your Teammates**

* Each of you must choose a **different CSV**
* You all should produce **separate visualisations**
* But discuss together to find **common themes or insights** about the article corpus

For example:

* One person analyzes article lengths
* One analyzes most common n-grams
* One tracks TF-IDF scores of keywords
* One explores topic modeling results

Then you combine your findings to say:  
🧠 “Over time, articles became shorter, and topics shifted from conflict to diplomacy.”

**📁 STEP 6: Build Your Portfolio Folder**

Your team submission should include:

* One exploration and one presentation script **per member**
* One final visualisation **per member**
* Comments in each script explaining the logic
* A **summary of insights** combining everyone’s observations

**✅ Checklist for Each Team Member**

| **Task** | **Done?** |
| --- | --- |
| Chose a unique dataset | ☐ |
| Wrote an exploration script with comments | ☐ |
| Designed a final visualisation (presentation script) | ☐ |
| Used pandas and plotly | ☐ |
| Explained the data method (e.g., what TF-IDF means) | ☐ |
| Formatted the visualisation well | ☐ |

Would you like help breaking down how to explore and present a specific dataset like **n-gram frequency** or **article lengths** next?

**Entry #2**

**Prompt:**  
I'm a student doing a Python project in a DH course. I’ve already cleaned and merged my data. Now, I want to visualize average article length over time for the top 5 topics. Can you help me understand how to group the data monthly by topic and calculate average lengths using pandas?

**Response:**  
Sure! You're on the right track. To group by month and topic, make sure your dataframe has a proper datetime column. You can use:

python

CopyEdit

df["date"] = pd.to\_datetime(df["date\_column"])

Then group like this:

python

CopyEdit

monthly\_avg = df.groupby(["date", "Topic\_Label"])["length"].mean().reset\_index()

This groups the data by each date-topic combination and calculates the mean article length. You can then use a line plot (like with Plotly Express) to visualize how average article lengths evolve over time for each topic.

**Entry#3**

I need to merge two data frames to get my required merge .csv is it a right approach to combine two data frames for my project explain

ChatGPT said:

Yes, merging two dataframes is absolutely the right approach for your project, especially in the context of Digital Humanities and text analysis. Here's why — along with a clear explanation tailored to your project:

✅ Why Merging is the Right Approach

In your project, you're working with two different dataframes:

1. length\_df: contains article lengths and publication dates.
2. topic\_df: contains topic modeling results for the same articles.

These datasets are related — they both describe the same articles — but they contain different types of information. You need both types of information together to analyze article length by topic, or to filter for the top 5 topics and study their distribution.

🧠 The Goal of Merging

You want to:

* Combine the topic label (from topic\_df) with the article length and date (from length\_df),
* So you can group and analyze articles by topic, length, and time.

This kind of analysis is not possible if the data stays split.

🧪 How You’re Doing It

In your code, you're using:

python

CopyEdit

merged = pd.merge(length\_df, topic\_df, on=["year", "month", "day"], how="inner")

This is a correct and appropriate method, assuming:

* Both dataframes include year, month, and day columns that uniquely identify each article,
* And that you’ve cleaned out any unmatched or incomplete records beforehand (e.g., removing topic = -1).

The how="inner" ensures that only rows with a match in both dataframes are included — this is important to avoid junk rows.

📦 Why You Save the Merged Dataset

You later save the merged result as:

python

CopyEdit

merged\_top.to\_csv("outputs/merged\_top\_5.csv", index=False)

This is a smart move:

* It creates a clean, ready-to-analyze version of your data,
* Reduces repetition of cleaning steps,
* And keeps your project modular and organized.

✅ Conclusion (Academic Style)

Yes, combining the length\_df and topic\_df using a merge operation is the correct and necessary approach in this project. It allows you to consolidate different dimensions of your data — topical and temporal — enabling deeper analysis such as calculating average lengths by topic over time. Without merging, such multi-variable analysis would not be possible. This strategy aligns with the goal of integrating Natural Language Processing outputs with structural metadata to uncover meaningful patterns in news coverage.

**Entry#4**

Is this a good approach: to create a line chart that shows the average article length over time for the top 5 topics using plotly?

" topic\_monthly\_avg = merged\_top.groupby(["date", "Topic\_Label"])["length"].mean().reset\_index() fig1 = px.line(topic\_monthly\_avg, x="date", y="length", color="Topic\_Label", title="Average Article Length Over Time by Top 5 Topics", labels={"length": "Average Length", "date": "Date", "Topic\_Label": "Topic"}, markers=True)

ChatGPT said:

Yes, your approach is very good and appropriate for your project. Here's why:

✅ Why This Is a Good Approach

1. Clear Research Question Match

You're analyzing how article lengths vary over time across the top 5 topics, which is a valuable insight. A line chart is the best way to show temporal (over time) trends.

2. Correct Use of GroupBy

You're using:

python

CopyEdit

topic\_monthly\_avg = merged\_top.groupby(["date", "Topic\_Label"])["length"].mean().reset\_index()

This groups your data by both topic and date, and calculates the average article length per topic per month — exactly what you want.

3. Good Use of Plotly for Interactivity

Using plotly.express.line() is excellent:

* It creates interactive charts.
* markers=True makes the data points stand out.
* You’ve labeled axes clearly with:

python

CopyEdit

labels={"length": "Average Length", "date": "Date", "Topic\_Label": "Topic"}

4. Good for Presentation

This graph can:

* Show if some topics tend to have longer articles.
* Highlight seasonal spikes or trends.
* Be easily understood by non-technical audiences.

🛠️ Minor Suggestions (Optional)

If your dataset is large or monthly is too noisy, you might:

* Resample to quarterly: merged\_top['date'].dt.to\_period('Q')
* Add smoothing (e.g., rolling average) to reveal long-term trends

✅ Summary

Yes — this is a good, thoughtful, and academically honest approach using code techniques taught in class. You’ve chosen a visualization type that fits the research question and used readable, well-commented code.

Would you like help writing a short paragraph to include this graph in your project report or presentation script?