Workflow of functions (in order)

1. **Load** the raw file with `pd.read_csv`.
2. **Drop** any blank rows using `Series.dropna()`.
3. **Cast** the column to strings just in case: `Series.astype(str)`.
4. **Trim** edge spaces via `Series.str.strip()`.
5. **Lower-case** everything with `Series.str.lower()`.
6. **Remove punctuation** using
   `Series.str.replace( ... string.punctuation ... )`
   (advanced: wrap punctuation list in `re.escape` for safety).
7. **(Optional) Filter** rows, e.g. `series[series != ""]`.
8. **Measure** uniqueness with `Series.nunique()`.
9. **Count** frequencies via `Series.value_counts()`.
10. **Write** the clean data back out with `DataFrame.to_csv()`.
11. **Visualise** top counts: `Series.plot.bar()` then `plt.show()`.

Quick-Reference Table

Helper	Why / When	Mini Example
`pd.read_csv`	Load a CSV into a DataFrame	`df = pd.read_csv("data/messy_strings.csv")`
`Series.dropna()`	Remove missing cells early so stats aren't skewed	`df["raw"] = df["raw"].dropna()`
`Series.astype(str)`	Guarantee values are strings before string ops	`df["raw"] = df["raw"].astype(str)`
`Series.str.strip()`	Clip leading / trailing whitespace	`df["raw"] = df["raw"].str.strip()`
`Series.str.lower()`	Standardise casing for matching/deduping	`df["raw"] = df["raw"].str.lower()`
`Series[cond]`	Select rows meeting a condition	`letters = df["raw"][df["raw"].str.len() == 1]`
`Series.nunique()`	Count distinct cleaned strings	`uniq = df["raw"].nunique()`
`Series.value_counts()`	Frequency table (descending)	`counts = df["raw"].value_counts()`
`DataFrame.to_csv()`	Save cleaned output for grading	`df.to_csv("clean.csv", index=False)`
`Series.plot.bar()`	Quick bar plot of counts	`counts.head(5).plot.bar()`
`plt.show()`	Render current matplotlib figure	`import matplotlib.pyplot as plt; plt.show()`