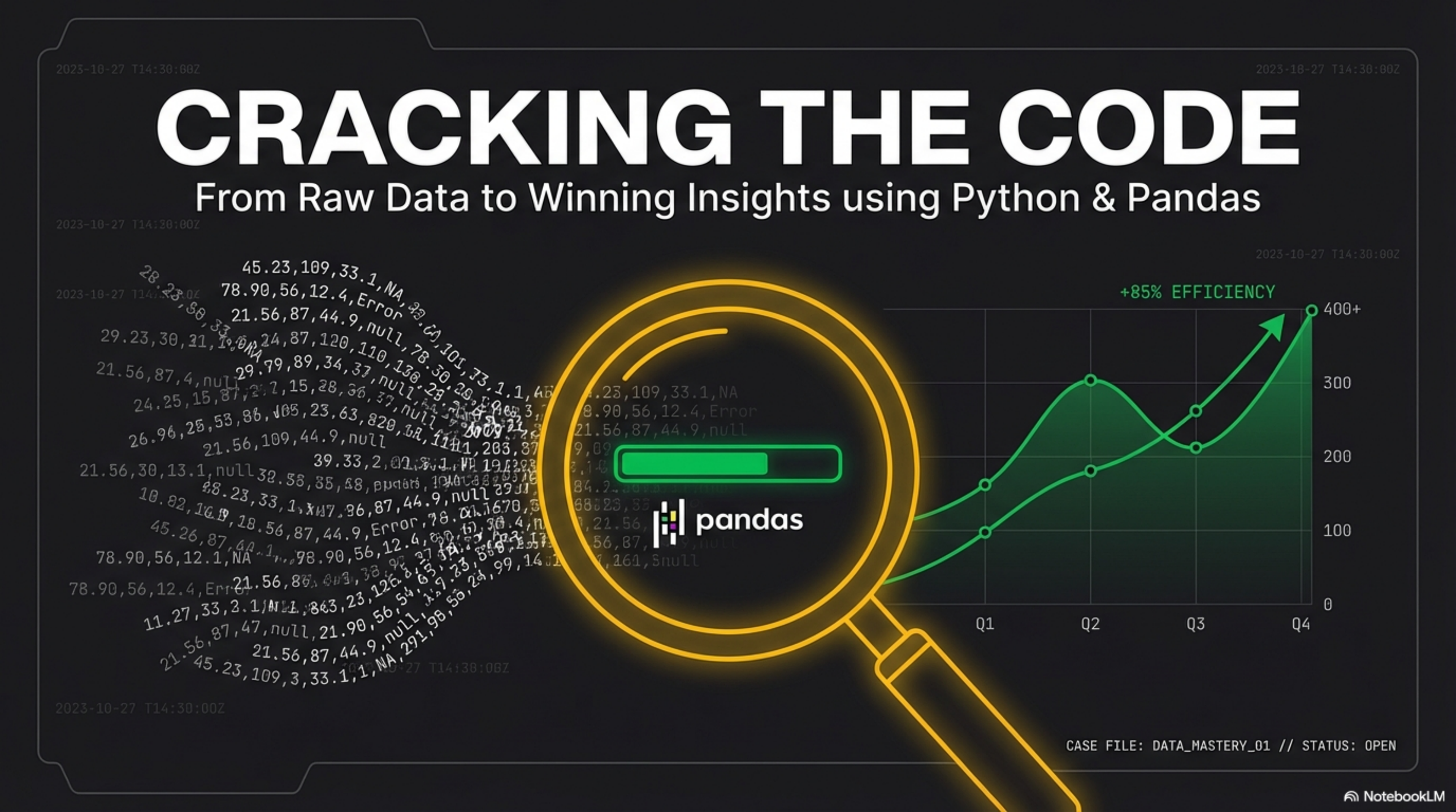
CRACKING THE CODE

From Raw Data to Winning Insights using Python & Pandas



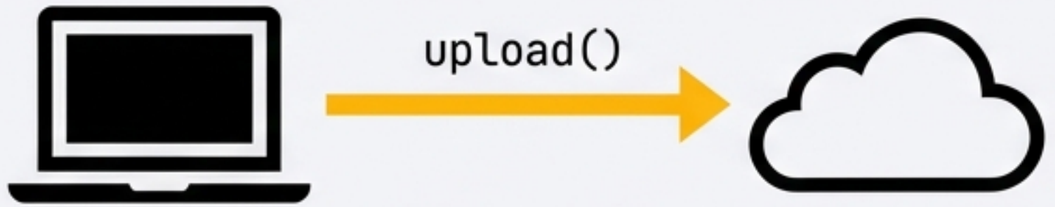
```python
import pandas as pd
from google.colab import files

uploaded = files.upload()
```

The Foundation

Every investigation needs a kit. We begin by importing Pandas, the industry-standard toolbox for data manipulation, and bridging the gap between local storage and the cloud.

Strategic Insight

You can't build a house without a solid foundation.



Python

```
df = pd.read_csv('your_filename_here.csv')
df.head()
```

Table Preview

ID	Date	Category	Value	Status
001	2023-10-27	Evidence	150.0	PENDING
002	2023-10-26	Witness	75.5	VERIFIED
003	2023-10-27	Suspect	220.0	UNDER_REVIEW
004	2023-10-25	Location	90.0	CONFIRMED
005	2023-10-27	Timeline	110.0	ANALYZING
001	2023-10-27	Evidence	150.0	ANALYZING
002	2023-10-26	Witness	75.5	VERIFIED
004	2023-10-25	Location	90.0	UNDER_REVIEW

Opening the Vault

The first look at the crime scene. We use `read_csv` to transform raw text into a structured DataFrame (`df`) and `head()` to take a sneak peek at the evidence.

Strategic Insight

Always verify your data before you trust it.

2023-10-27 09:00:00 UTC

Removing the Noise

```
Ball 1
Ball 2
GreenBall
DoubleDraw
```

Not all data is signal. We discard irrelevant columns (**red herrings**) to focus purely on the core numbers. `axis=1` ensures we cut vertically (columns), not horizontally.

```python
df.drop(columns=columns_to_remove,
        axis=1,
        inplace=True)
```

Strategic Insight

Simplicity is the ultimate sophistication. Eliminate the fluff.

Python

```python
for col in cols_to_fix:
    df[col] = df[col].astype(int)
```

Standardization

The witnesses aren't speaking the same language. We iterate through the columns and force inconsistent decimals into clean integers.

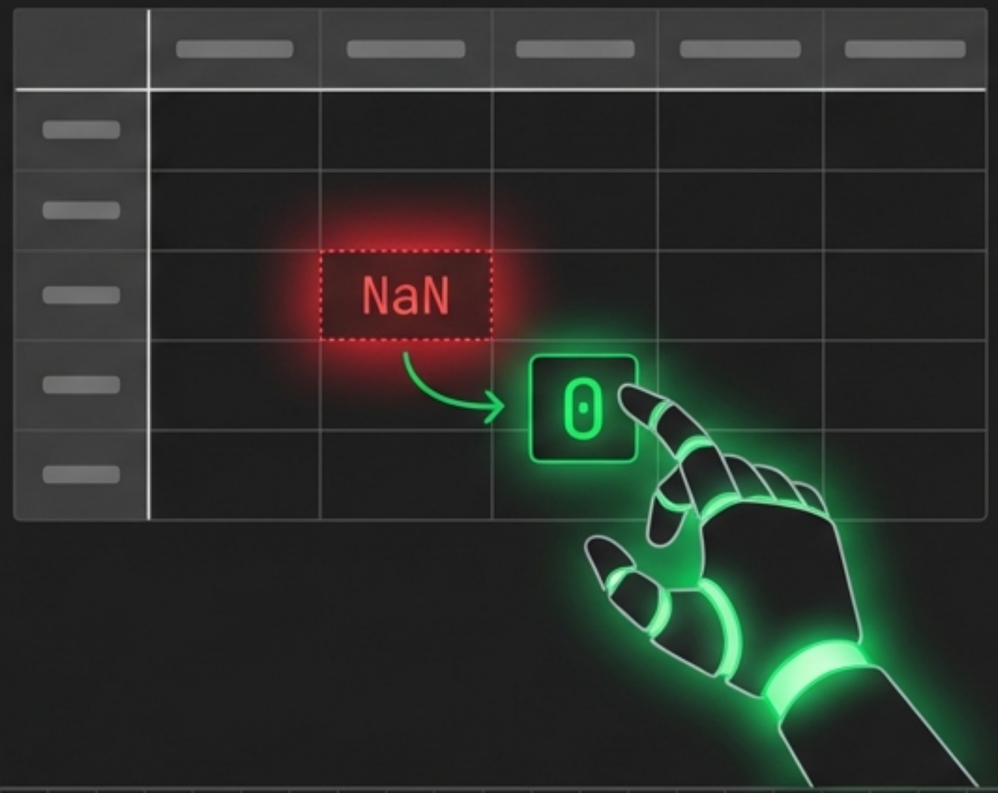
Strategic Insight

Clean data leads to clear results. Numbers must speak the same language.

2023-10-27 09:00:00 UTC

Python

```
df['Fireball'] = df['Fireball'].fillna(0)
```



Patching the Holes

Missing values (NaNs) are potholes that crash calculation scripts. We strategically fill these voids with 0 to maintain structural integrity without skewing the dataset.

Highlighter Amber

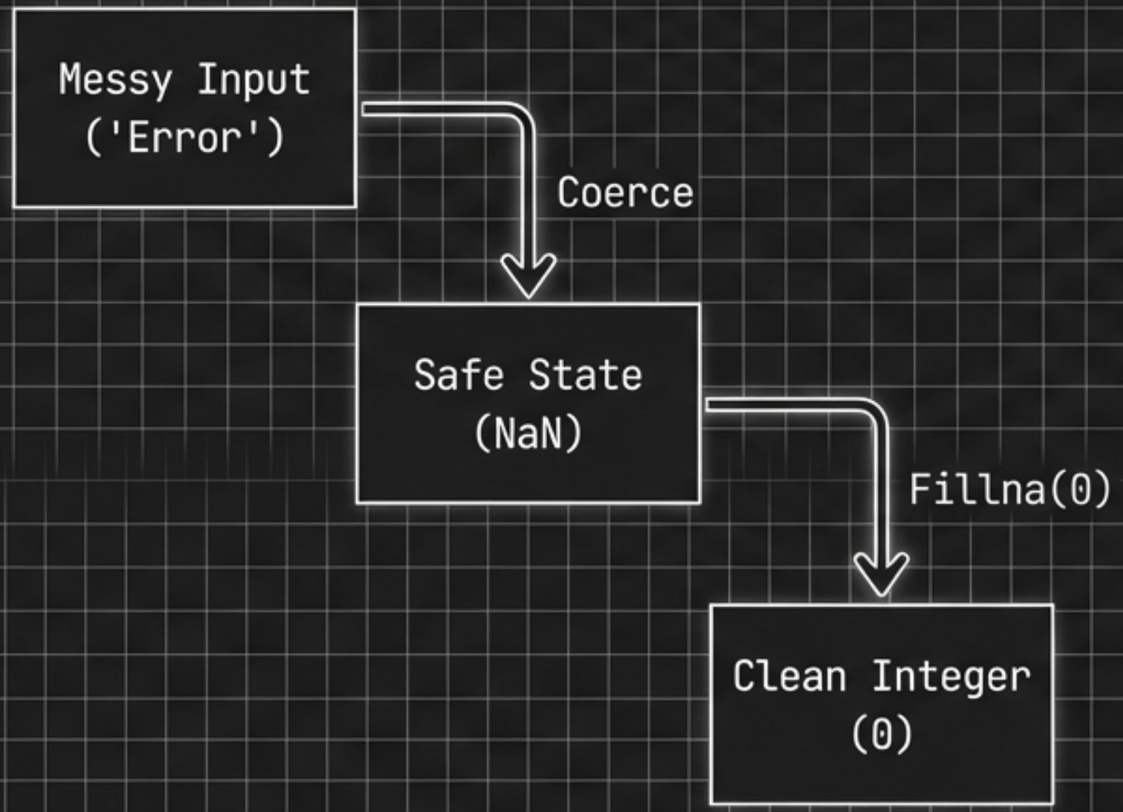
Strategic Insight 💡

Don't let missing info stop your progress. Patch the gaps.

2023-10-27 09:00:00 UTC

Python

```python
pd.to_numeric(df[col], errors='coerce')
```



The Deep Clean

Some data refuses to comply. We use coercion to force non-numeric garbage into a safe state (NaN) rather than letting it break the system.

Strategic Insight

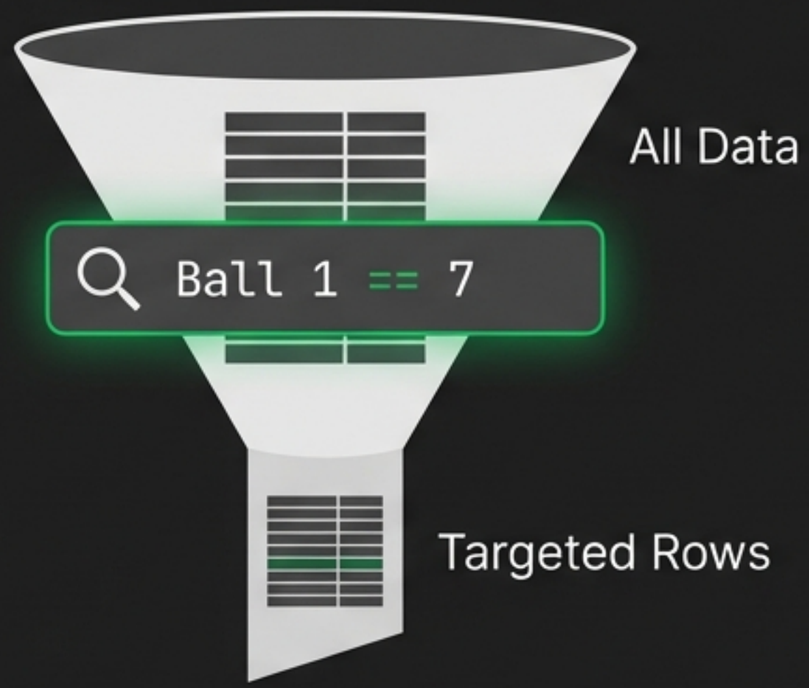
Automation is about making the computer handle the mess.

2023-10-27 09:00:00 UTC

Python

```
sevens = df[df['Ball 1'] == 7]
```

Comparison Operator
(Not Assignment)



The Detective's Filter

We start the interrogation. By asking specific Boolean questions—"Show me every row where 7 is the first suspect"—we turn a mountain of data into a targeted list.

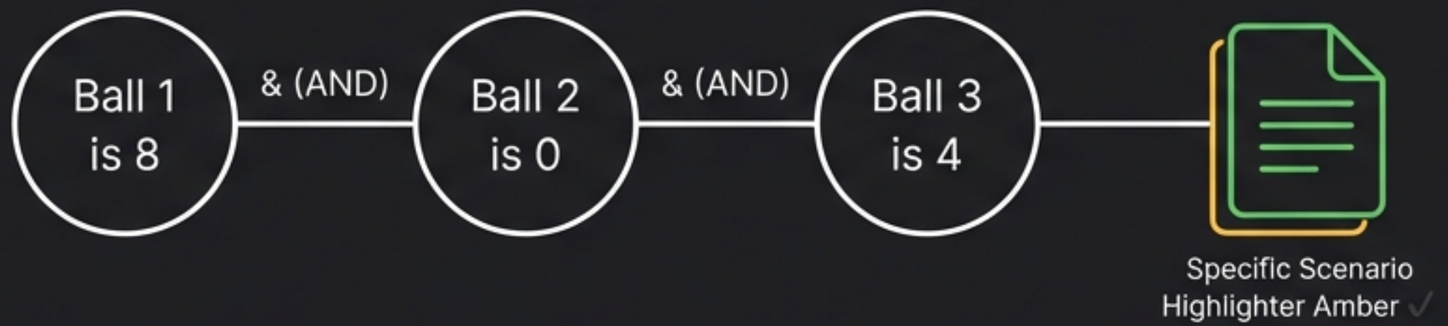
Strategic Insight

Data is only powerful if you know the right questions to ask.

2023-10-27 09:00:00 UTC

Python

```
df[(df['Ball 1'] == 8) &
        (df['Ball 2'] == 0) &
        (df['Ball 3'] == 4)]
```



Advanced Logic Chains

The investigation deepens. We chain multiple conditions using the `&` operator to isolate very rare, specific historical events.

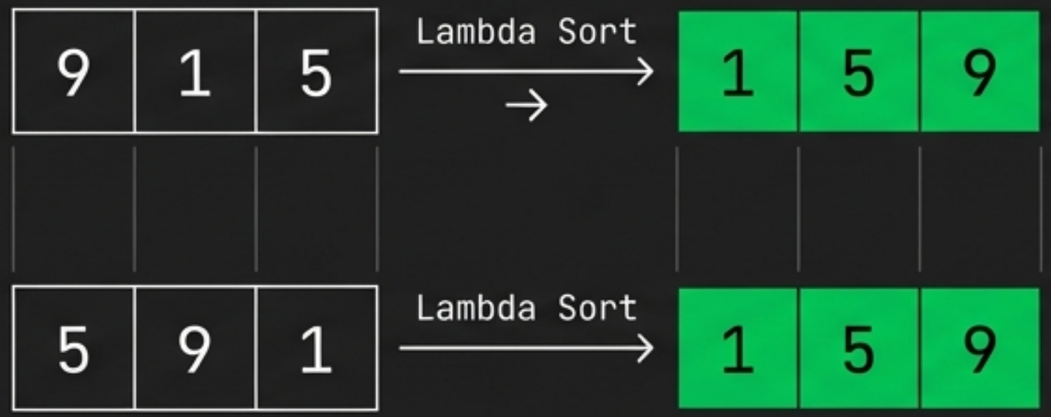
Highlighter Amber

Strategic Insight

Precision querying allows for granular historical analysis.

Python

```
.apply(lambda row:
sorted(row.values))
```



The Magic Line: Row-Wise Sorting

The Problem: 1-5-9 and 9-1-5 are the same result in a “Box” win, but computers see them as different.

The Solution: We apply a Lambda function to sort numbers **across** the row.

Strategic Insight

Python allows us to see patterns the human eye would miss by normalizing chaos.

Python

Frequency Analysis

Who are the usual suspects? We flatten the dataset using `stack()` to treat all positions equally, then count the total frequency of digits 0-9.

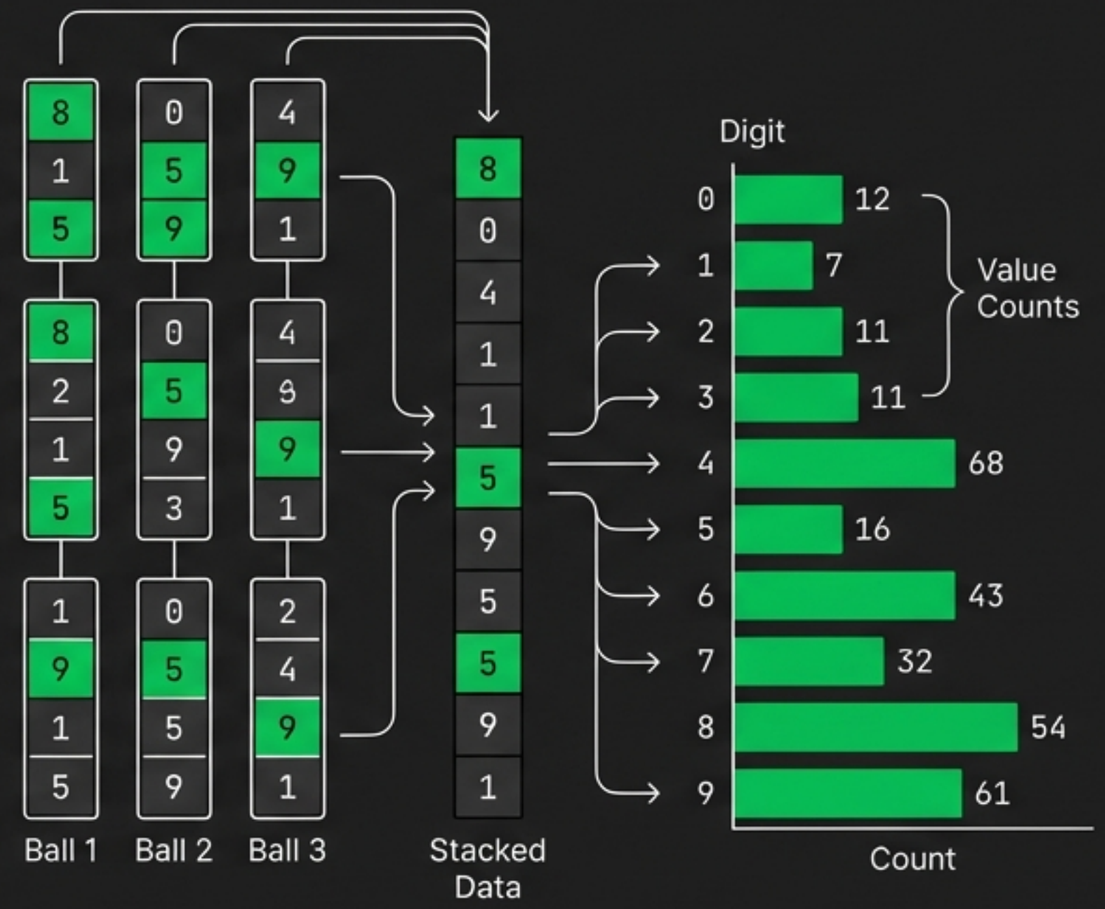
Strategic Insight #FFC107

Moving from microscopic (row-level) to macroscopic (aggregate) trends.

2023-10-27 09:00:00 UTC

JetBrains Mono (#00C853)

```
.stack().value_counts()
```



Python

```python
from itertools import combinations
from collections import Counter

# Create a blank counter
pair_counter = Counter()

# Example row and its combinations
row_values = [1, 5, 9]
ball_list = sorted(row_values)  # [1, 5, 9]
pairs = list(combinations(ball_list, 2))

# Update counter
pair_counter.update(pairs)

# Print the pairs being counted
print(f"Pairs: {pairs}")
# Output: Pairs: [(1, 5), (1, 9), (5, 9)]

# Print counter update
for pair in pairs:
    print(f"Pair {pair}: +1")
# Output example: Pair (1, 5): +1
```

Combinatorics: Finding Pairs

We look for accomplices. By breaking every draw into component pairs, we uncover hidden correlations and identify which numbers appear together most often.

Strategic Insight

Highlighter Amber

Uncovering hidden correlations between independent variables.

Temporal Analysis

Relevance requires context. We apply our pair-counting logic to a specific window of time to distinguish all-time historical trends from current ‘hot’ streaks.

Strategic Insight

Contextualizing data by time is crucial for relevance.

2023-10-27 09:00:00 UTC

```
df[(df['Date'].dt.month == 2) &
   (df['Date'].dt.year == 2026)]
```

JetBrains Mono (A00C8S3)

Python

```python
# Line 1: We tell Pandas to save our cleaned 'df' to a new file.
# 'index=False' prevents Python from adding an extra column of row numbers.
df.to_csv('cleaned_data.csv', index=False)

# Line 2: This is a Colab-specific tool to trigger a download in your browser.
from google.colab import files
files.download('cleaned_data.csv')
```

Securing the Evidence

The case is closed. We package the structured, clean data and export it from the cloud back to local storage for future use.

Strategic Insight

The Cycle is Complete: Raw Data -> Clean -> Insight -> Saved Asset.

2023-10-27 09:00:00 UTC

The Verdict

CASE CLOSED

- [x] DROP (Remove Noise)
- [x] FILLNA (Repair Gaps)
- [x] LAMBDA (Custom Logic)
- [x] COMBINATIONS (Pattern Recognition)

JetBrains Mono
(#00C8S3)

We didn't just read the numbers; we restructured reality to make the answers obvious. That is the power of Python.