Web Server Log Analysis - Python Take-Home Assessment **2**
- Overview 2
- Dataset Information 2
  - Source Details 2
  - Data Structure 2
- Apache Common Log Format Reference 3
  - Field Definitions 3
  - Sample Request Breakdown 3
- Assessment Tasks 3
  - Part 1: Data Loading and Cleaning 3
    - Download and Load Data 3
    - Data Cleaning 3
  - Part 2: Analysis Questions (5 marks each) 4
- Deliverables 5
  - File Requirements 5
  - Video Instructions (Not Part of ZIP) 6
  - Instructions 6

Web Server Log Analysis - Python Take-Home Assessment

Overview

This assessment involves analyzing the Calgary HTTP dataset, which contains approximately one year's worth of HTTP requests to the University of Calgary's Computer Science web server. You'll work with real-world web server log data to extract meaningful insights and demonstrate your Python data analysis skills.

Dataset Information

Source Details

- **Dataset Name**: Calgary HTTP Dataset
- **Dataset Source**: https://ita.ee.lbl.gov/html/contrib/Calgary-HTTP.html
- **Download URL**: ftp://ita.ee.lbl.gov/traces/calgary_access_log.gz
- **Format**: ASCII text file (one request per line)
- **File Size**:
  - **Compressed**: 5.4 MB
  - **Uncompressed**: 52.3 MB

Data Structure

Each log entry contains the following space-separated fields:

Field	Description	Example
host	Request origin: "local" (University of Calgary) or "remote" (external)	remote
timestamp	Request timestamp	Mon Oct 24 14:23:15 1994
filename	Requested file in "num.type" format	1234.html
http_code	HTTP response status code	200, 404, 500
bytes	Response size in bytes (or - if unavailable)	2048

Apache Common Log Format Reference

The dataset follows the standard Apache Common Log Format used by W3C HTTPD servers:

```
remotehost rfc931 authuser [date] "request" status bytes
```

Field Definitions

Field	Meaning	Notes
remotehost	Remote hostname or IP address	IP used if hostname unavailable
rfc931	Remote logname of user	Hyphen (-) if unavailable
authuser	Authenticated username	Hyphen (-) if not authenticated
[date]	Request timestamp	Format: [day/month/year:hour:minute:second zone]
"request"	Full HTTP request line	Example: "GET index.html HTTP/1.0"
status	HTTP status code	200, 404, 500, etc.
bytes	Content-length transferred	Size in bytes

Sample Request Breakdown

```
"GET index.html HTTP/1.0"
├── Method: GET
├── Resource: index.html
└── Protocol: HTTP/1.0
```

Assessment Tasks

Part 1: Data Loading and Cleaning

Download and Load Data

- Download the compressed dataset from the FTP server
- Handle the `.gz` compression format appropriately
- Load the data into a suitable Python data structure
- Implement proper error handling for network/file operations

Data Cleaning

- Parse timestamp strings into `datetime` objects

- Extract file extensions from the filename field
- Handle missing or malformed data entries
- Convert data types appropriately (integers, strings, etc.)
- Remove or flag invalid log entries

Part 2: Analysis Questions (5 marks each)

- Q1: Count of total log records
  - Description: Count the total number of HTTP requests in the log file. Each line represents one log entry.
  - Return Type: int
  - Example: 123456
- Q2: Count of unique hosts
  - Description: Determine the number of distinct hosts (IP addresses or domain names) that accessed the server.
  - Return Type: int
  - Example: 8567
- Q3: Date-wise unique filename counts
  - Description: For each date, count how many unique filenames were requested.
  - Return Type: dict[str, int]
  - Format: { '01-Jul-1995': 123, '02-Jul-1995': 150 }
  - Note: Date format must be 'dd-MMM-yyyy'
- Q4: Number of 404 response codes
  - Description: Count how many HTTP requests resulted in a 404 (Not Found) response.
  - Return Type: int
  - Example: 3490
- Q5: Top 15 filenames with 404 responses
  - Description: Find the 15 most requested URLs that resulted in a 404 error, sorted by frequency.
  - Return Type: list[tuple[str, int]]
  - Format: [('missing.html', 200), ('notfound.gif', 123), ...]
- Q6: Top 15 file extensions with 404 responses
  - Description: Identify the file extensions (like .html, .jpg) that caused the most 404 errors.
  - Return Type: list[tuple[str, int]]
  - Format: [('html', 345), ('gif', 220), ...]
- Q7: Total bandwidth transferred per day for July 1995
  - Description: Sum the bytes transferred per day for July 1995 (exclude missing or '-' byte values).
  - Return Type: dict[str, int]

    - Format: { '01-Jul-1995': 123456789, ... }
- Q8: Hourly request distribution
    - Description: Count how many HTTP requests occurred during each hour (0–23).
    - Return Type: dict[int, int]
    - Format: { 0: 1200, 1: 900, ..., 23: 670 }
- Q9: Top 10 most requested filenames
    - Description: Identify the top 10 most frequently requested filenames, regardless of status code.
    - Return Type: list[tuple[str, int]]
    - Format: [('index.html', 5678), ('home.gif', 4321), ...]
- Q10: HTTP response code distribution
    - Description: Count the occurrences of each HTTP response status code (e.g., 200, 404).
    - Return Type: dict[int, int]
    - Format: { 200: 150000, 404: 3200, 500: 87 }

Deliverables

Candidates are required to submit the following items as part of their assessment. Submissions must follow the folder structure and format guidelines exactly. Failure to comply may result in disqualification.

📁 Folder Structure (Final ZIP)

```
calgary-http-assessment/
├── analysis.ipynb              # Jupyter Notebook with code and all outputs
├── analysis.html               # Exported HTML version of the notebook
├── transcript.txt              # Transcript of your explanation video
├── README.md                   # (Optional) Additional notes or instructions
```

Compress the calgary-http-assessment/ folder into a single ZIP file:

```
calgary-http-assessment.zip
```

File Requirements

File Name	Format	Description
analysis.ipynb	.ipynb	Jupyter Notebook with all output cells included. Notebooks without output will be rejected.

analysis.html	.html	Exported version of the notebook for quick viewing. Use "File > Export Notebook As > HTML".
transcript.txt	.txt	Full transcript of the video. You may upload your video as unlisted on YouTube and use YouTube's transcript feature. Then, paste the transcript in a .txt file.
README.md	.md	(Optional) Notes on approach, assumptions, tools used, or challenges.

Video Instructions (Not Part of ZIP)

You must create a short screen recording in English where you:

- Explain your code and logic
- Execute the notebook from start to finish
- Briefly describe your approach and any challenges

Upload the video as an unlisted video on YouTube and provide the video link in the Google Form. **Do not include the video in the ZIP file**.

Instructions

- Language: All materials (notebook, comments, video, transcript) must be in English.
- Notebook Outputs: Ensure all cells in analysis.ipynb are executed. Notebooks without visible output cells will be rejected.
- Transcript: Must reflect the spoken content in your video. You may use YouTube's auto-generated transcript as a base.
- File Naming: Do not rename the required files. Follow the structure exactly.
- Folder Structure: Use the exact structure above. Do not include extra files or folders.
- Final Submission: Upload a ZIP named calgary-http-assessment.zip and paste your YouTube video link into the Google Form.

Good luck with your analysis! This assessment will demonstrate your ability to work with real-world data and extract meaningful insights from web server logs.