





Python code:

import pandas as pd

import matplotlib.pyplot as plt

import os

import numpy as np

import json

def custom\_plot(df, folder, filename, title, xlabel, ylabel, figsize=(10, 6), fontsize=12, xlabelsize=10, ylabelsize=10):

if not os.path.exists(folder):

os.makedirs(folder)

plt.figure(figsize=figsize)

counts = df['unique\_id'].value\_counts()

plt.bar(counts.index, counts.values)

plt.title(title, fontsize=fontsize)

plt.xlabel(xlabel, fontsize=xlabelsize)

plt.ylabel(ylabel, fontsize=ylabelsize)

plt.yscale('log')

plt.savefig(os.path.join(folder, filename))

plt.show()

plt.close()

def plot\_unique\_id\_distribution(df, folder):

if not os.path.exists(folder):

os.makedirs(folder)

counts = df['unique\_id'].value\_counts()

plt.figure(figsize=(12, 8))

plt.hist(counts, bins=50, color='blue', log=True)

plt.title('Histogram of Unique ID Frequencies', fontsize=20)

plt.xlabel('Frequency of Unique IDs', fontsize=15)

plt.ylabel('Number of Unique IDs (Log Scale)', fontsize=15)

plt.grid(True)

plt.savefig(os.path.join(folder, 'histogram\_unique\_id.png'))

plt.show()

def analyze\_uniqueness(df):

counts = df['unique\_id'].value\_counts()

max\_count = counts.max()

if max\_count > 1:

print(f"The maximum count for a single unique\_id is {max\_count}, which indicates duplicates.")

else:

print("Each 'unique\_id' is unique across the dataset.")

def main():

folder\_path = 'G:/Lysunochka/work and coaching from marielle obells/task 2/marielle correction resume+motivation letter/new/phd tue/Dirk tue/task 3 on real-life data from the project/prepared\_data/prepared\_data'

output\_folder = 'output' # Folder for outputs including plots and JSON files

all\_files = [f for f in os.listdir(folder\_path) if f.endswith('.csv')]

df\_list = []

if not all\_files:

print("No CSV files found in the specified directory.")

return

for file in all\_files:

file\_path = os.path.join(folder\_path, file)

if os.stat(file\_path).st\_size == 0:

print(f"Skipping empty file: {file}")

continue

df\_temp = pd.read\_csv(file\_path)

df\_temp['Código'] = df\_temp['Código'].astype(str).str.strip()

df\_temp['N/S'] = df\_temp['N/S'].astype(str).str.strip()

df\_temp['unique\_id'] = df\_temp['Código'] + '-' + df\_temp['N/S']

df\_list.append(df\_temp)

if not df\_list:

print("All files were empty or missing required columns. No data to process.")

return

df = pd.concat(df\_list, ignore\_index=True)

df.dropna(subset=['unique\_id'], inplace=True)

# Save to JSON

json\_path = os.path.join(output\_folder, 'cleaned\_data.json')

if not os.path.exists(output\_folder):

os.makedirs(output\_folder)

df.to\_json(json\_path, orient='records', lines=True)

analyze\_uniqueness(df)

custom\_plot(df, output\_folder, 'unique\_id\_distribution.png', 'Distribution of unique identifiers', 'Unique ID', 'Number of records', fontsize=20, xlabelsize=20, ylabelsize=20)

plot\_unique\_id\_distribution(df, output\_folder)

if \_\_name\_\_ == "\_\_main\_\_":

main()