

csv_search_generic_gg_polars

November 2, 2023

Jordan Jiosi

The Estate Registry

(Sample)

November 2, 2023

0.1 Import dependencies

```
[101]: import glob, shutil, time, subprocess, sys, os, os.path, re, csv, json, \
        ↪datetime
import pandas as pd
import polars as pl
from art import *
```

0.2 Set environment

```
[102]: # Set styling
Art=text2art("TER",font='block',chr_ignore=True) # Return ASCII text with block
        ↪font
Art2=text2art("The Estate Registry",font='cybermedum',chr_ignore=True) # Return
        ↪ASCII text with block font
ArtLine=text2art("-"*25,font="cybermedum",chr_ignore=True) # Return ASCII text
        ↪with block font
ArtLine=text2art("-"*25,font="cybermedum",chr_ignore=True) # Return ASCII text
        ↪with block font
print(Art)
print(Art2)
print(ArtLine)

# -----
# Set results env
results_file = "search_results.txt"
results_file_csv = "search_results.csv"
```



```
[103]: totalFileCount: int = 0
csvFileNamesReviewed: [str] = []
incFound: [str] = []
searchTerms: [str] = ["estate-registry", "jiosi"]
lnames: [str] = ["jiosi"]
```

1 The Saving, Searching, & pd/terminal Pretty-Printing Algorithms

```
[104]: def save_results_csv(filename, df, directory=destinationDir):
    """
    Save the provided polars DataFrame to a CSV file.
    """
    filepath = os.path.join(directory, filename)

    # Use Polars' to_csv function to save DataFrame to CSV
    df.write_csv(filepath)

    print(f"Results saved to CSV: {filepath}")

def search_csv_files_for_term(sourceDir=sourceDir, search_terms=searchTerms,
    ↪save_file=results_file):
    """
    Search CSV files for specified terms and save the results, displaying the
    ↪data using pandas for easy viewing.
    """
    totalFileCount = 0
    csvFileNamesReviewed = []
    df_c: pl.DataFrame = None
    error_message = None # capture any error messages

    print(f'{'*' * 25 + " Search Parameters " + '*' * 25}')
    print(f'Searching the following directory: {sourceDir}')
    print(f'Searching for the following terms: {search_terms}')
    print(f'{'*' * 25 + "=" * 15 + '*' * 25}')

    try:
        for file in os.listdir(sourceDir):
            full_path = os.path.join(sourceDir, file) # Using full path
            print(f"Searching file: {full_path}")
            if file.endswith(".csv"):
                totalFileCount += 1
                csvFileNamesReviewed.append(full_path)

                # Load CSV data into a pandas DataFrame -- Let's try Polars
                # df_pan = pd.read_csv(full_path, header=1)
```

```

        # df_pls = pl.read_csv(full_path, has_header=True,
↳truncate_ragged_lines=True, skip_rows=1) # In-memory

        df = pl.scan_csv(full_path, skip_rows=1, has_header=True).
↳collect() # Lazily loaded
        cols = df.columns
        print('-'*10)
        print("All emails in file:")
        print(df.select(["Email Address"]).filter(pl.col("Email_
↳Address").is_not_null())
                .group_by('Email Address').count())
        print("All last names in file:")
        print(df.select(["Last Name"]).filter(pl.col("Last Name").
↳is_not_null())
                .group_by('Last Name').count())
        print("Matches and date found:")
        print(df.select(['Created Date', 'Email Address', 'Last Name']).
↳filter(pl.col("Email Address").str.contains(search_terms[0])))

        # Aggregate by "Last Name" and cast to i64
        res = df.select(["Email Address", "Last Name"]).filter(
                (pl.col("Email Address").str.contains(search_terms[0])) &
                (pl.col("Email Address").is_not_null())
        ).group_by(["Email Address"]).agg(
                count_email=pl.col("Email Address").count().cast(pl.Int64)
        )

        # select, filter total count matching search terms & cast to
↳i64
        filtered_df = df.select(["Email Address", "Last Name"]).filter(
                (pl.col("Email Address").str.contains(search_terms[0])) &
                (pl.col("Email Address").is_not_null())
        )
        total_count = filtered_df.shape[0]
        print(f"Matches found: {total_count}")
        print('-'*10)

        filename_column = pl.DataFrame({
                "Filename": [file for _ in range(filtered_df.shape[0])] #
↳Repeats the filename for each row in filtered_df
        })
        filtered_df_with_filename = filtered_df.hstack(filename_column)
        # This is the same DataFrame--CSV save we want like last time
↳with expected quick-peek BAU expected output
        columns_order = ["Filename", "Email Address", "Last Name"]

```

```

        filtered_df_ordered = filtered_df_with_filename.
↪select(columns_order)
        df_c = filtered_df_ordered

        # DF with int type for "count_email" for simplicity
        agg_total = pl.DataFrame({
            "Email Address": ["Total"],
            "count_email": [int(total_count)] # cast to int inferred
↪as i64 by polars
        })

        # Concatenated results for roll-up
        rollup_result = pl.concat([res, agg_total])

        print(f'Roll-up results: {rollup_result}')
        print('-'*25)
        print(f'Final results: {df_c}')
        print('-'*10)
        print('-'*40)

    except Exception as e:
        # Capture the error message
        error_message = f"An error occurred during the file search: {str(e)}"

        print(f'{' '*25 + " Search Results " + ' '*25}')
        print(f"Total files reviewed: {totalFileCount}")
        print(f"Files reviewed: {csvFileNamesReviewed}")
        print(f'Match criteria: {search_terms}')
        print(f"Matches found: {df_c.shape[0]}")
        print(f'{' '*25 + "="*15 + ' '*25}')
        print(f"Saving results to {save_file}")
        print(f'{' '*25 + "="*15 + ' '*25}')
        print(f'{' '*25 + "="*15 + ' '*25}')

        save_results_csv(results_file_csv, filtered_df_with_filename)

        # If there was an error, print the error message
        if error_message:
            print(error_message)

        return totalFileCount, csvFileNamesReviewed, df_c.shape[0]

```

```
[105]: total_files, files_reviewed, inc_found = search_csv_files_for_term()
```

```

***** Search Parameters *****
Searching the following directory:
/Users/jordan/Documents/PCA/local/NN_CSV_search/data
Searching for the following terms: ['estate-registry', 'jiosi']

```

Searching file:

/Users/jordan/Documents/PCA/local/NN_CSV_search/data/22-807_o7jp.pdf

Searching file: /Users/jordan/Documents/PCA/local/NN_CSV_search/data/2023-10-26_d6e0ed93-b667-44b1-9191-d92b179b4163_TEST_EXAMPLE_FILE_TRANSFER.csv

All emails in file:

shape: (2, 2)

Email Address	count
---	---
str	u32
jjiosi@estate-registry.com	1
randomuser@gmail.com	1

All last names in file:

shape: (1, 2)

Last Name	count
---	---
str	u32
Jiosi	1

Matches and date found:

shape: (1, 3)

Created Date	Email Address	Last Name
---	---	---
str	str	str
26/10/2023	jjiosi@estate-registry.com	null

Matches found: 1

Roll-up results: shape: (2, 2)

Email Address	count_email
---	---
str	i64
jjiosi@estate-registry.com	1
Total	1

Final results: shape: (1, 3)

Filename	Email Address	Last Name
---	---	---
str	str	str
2023-10-26_d6e0ed93-b667-44b1-91...	jjiosi@estate-registry.com	null

 Searching file: /Users/jordan/Documents/PCA/local/NN_CSV_search/data/2023-10-26_d6e0ed93-b667-44b1-9191-d92b179b4163.csv

All emails in file:
 shape: (5, 2)

Email Address	count
---	---
str	u32
jjiosi@estate-registry.com	1
user@estate-registry.com	1
randomuser@gmail.com	1
randomuser_2@gmail.com	1
superusersudoroot@gmail.com	1

All last names in file:
 shape: (2, 2)

Last Name	count
---	---
str	u32
Lyson	1
Jiosi	1

Matches and date found:
 shape: (2, 3)

Created Date	Email Address	Last Name
---	---	---
str	str	str
26/10/2023	jjiosi@estate-registry.com	null
26/10/2023	user@estate-registry.com	null

Matches found: 2

 Roll-up results: shape: (3, 2)

Email Address	count_email
---	---
str	i64
user@estate-registry.com	1
jjiosi@estate-registry.com	1
Total	2

Final results: shape: (2, 3)

Filename	Email Address	Last Name
---	---	---
str	str	str
2023-10-26_d6e0ed93-b667-44b1-91...	jjiosi@estate-registry.com	null
2023-10-26_d6e0ed93-b667-44b1-91...	user@estate-registry.com	null

```

-----
***** Search Results *****
Total files reviewed: 2
Files reviewed: ['/Users/jordan/Documents/PCA/local/NN_CSV_search/data/2023-10-26_d6e0ed93-b667-44b1-9191-d92b179b4163_TEST_EXAMPLE_FILE_TRANSFER.csv', '/Users/jordan/Documents/PCA/local/NN_CSV_search/data/2023-10-26_d6e0ed93-b667-44b1-9191-d92b179b4163.csv']
Match criteria: ['estate-registry', 'jiosi']
Matches found: 2
*****
Saving results to search_results.txt
*****
*****
Results saved to CSV:
/Users/jordan/Documents/PCA/local/NN_CSV_search/results/search_results.csv

```

2 Show some basic stats

```

[106]: print(f"Total files reviewed: {total_files}")
       print(f"Files reviewed: {files_reviewed}")
       # Load the CSV data into a pandas DataFrame
       df_res = pd.read_csv(os.path.join(destinationDir, results_file_csv))
       print("\n")
       print(f"Matches found:\n {df_res.head()}\n")
       print(f"Prelims: {df_res.shape[0]} matches found in {len(files_reviewed)}
       ↪file[s] reviewed")

```


Total files reviewed: 2

Files reviewed: ['/Users/jordan/Documents/PCA/local/NN_CSV_search/data/2023-10-26_d6e0ed93-b667-44b1-9191-d92b179b4163_TEST_EXAMPLE_FILE_TRANSFER.csv', '/Users/jordan/Documents/PCA/local/NN_CSV_search/data/2023-10-26_d6e0ed93-b667-44b1-9191-d92b179b4163.csv']

Matches found:

	Email Address	Last Name	\
0	jjios@estate-registry.com		NaN
1	user@estate-registry.com		NaN

	Filename
0	2023-10-26_d6e0ed93-b667-44b1-9191-d92b179b416...
1	2023-10-26_d6e0ed93-b667-44b1-9191-d92b179b416...

Prelims: 2 matches found in 2 file[s] reviewed