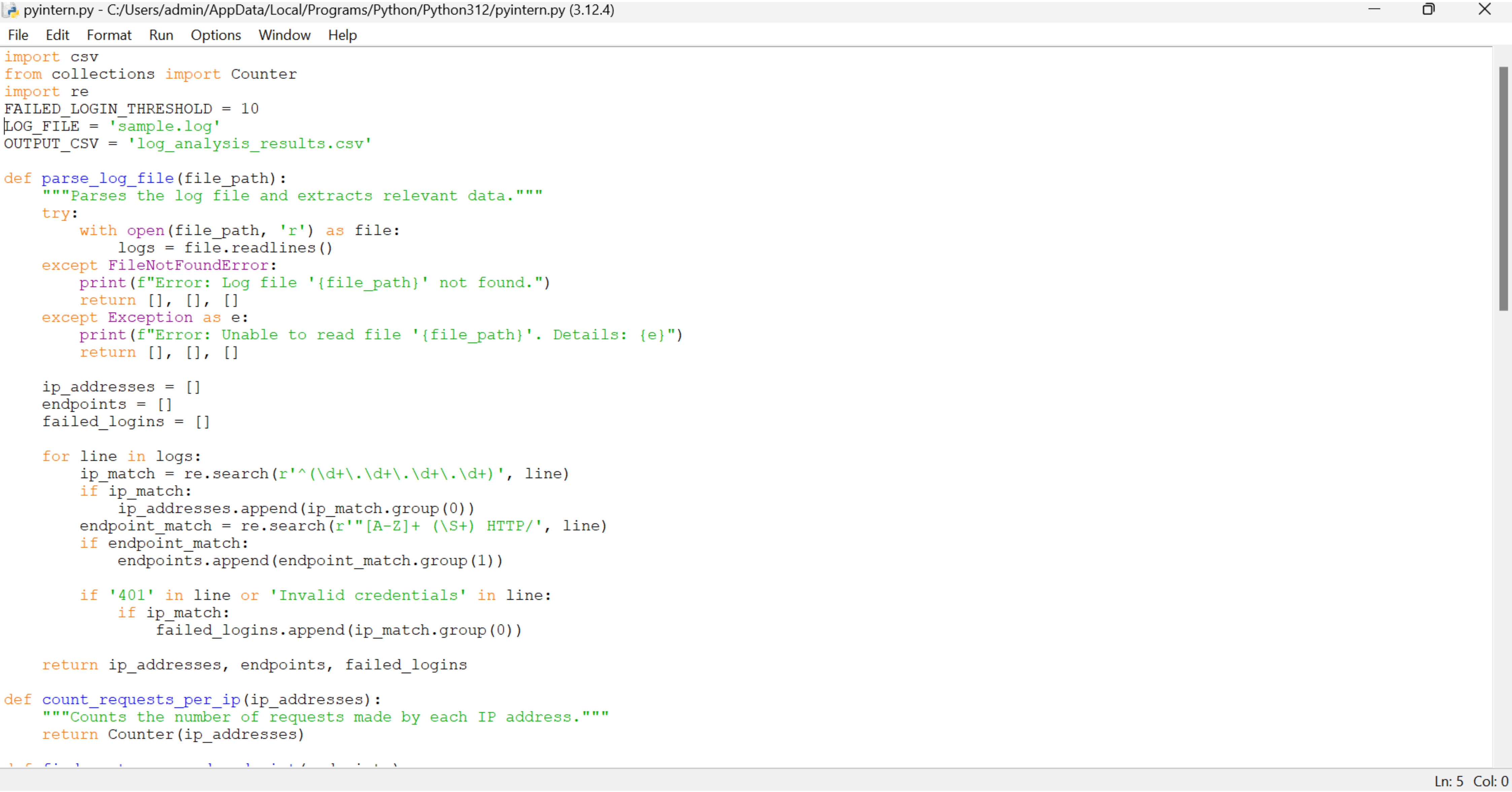
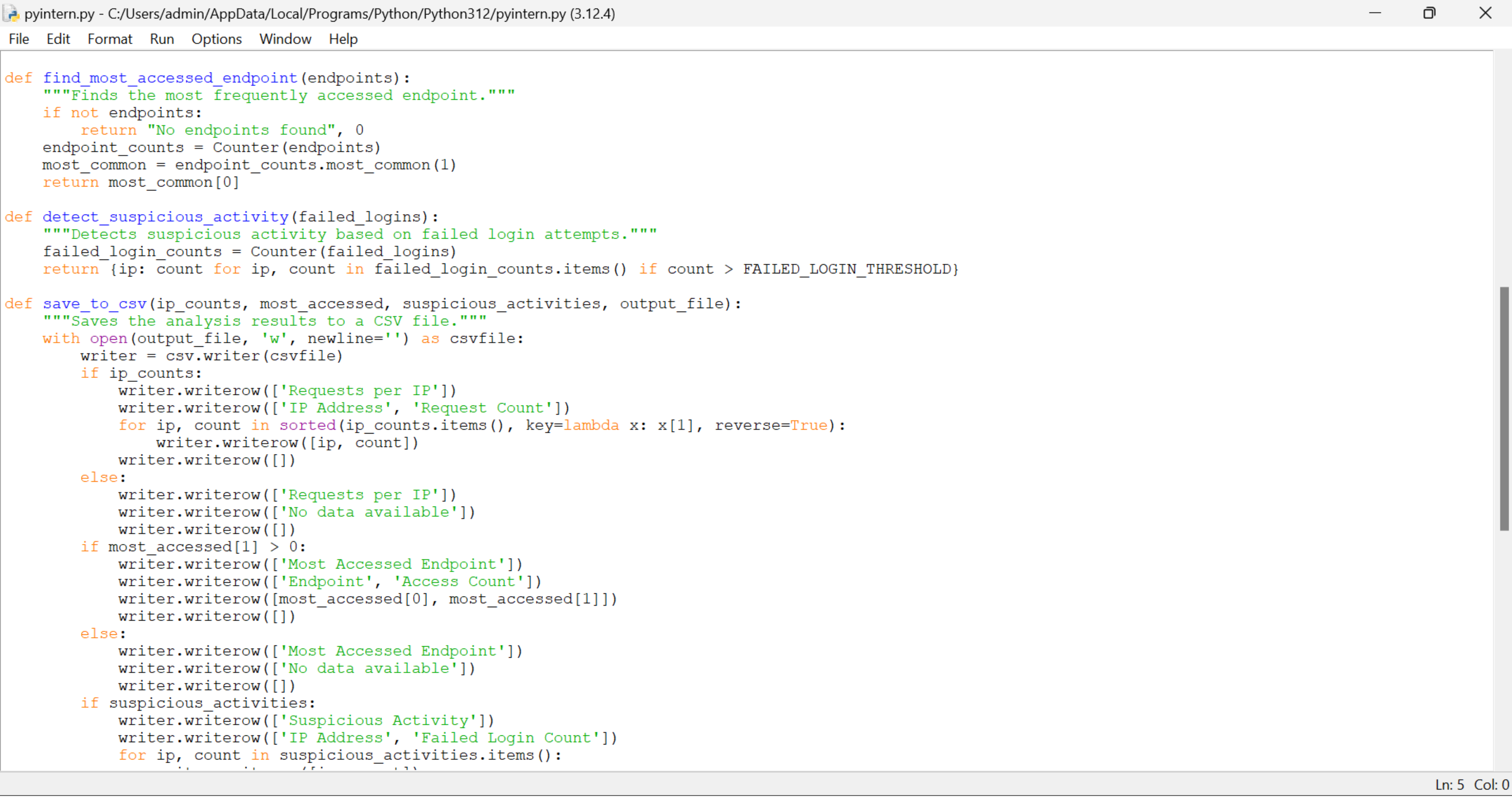
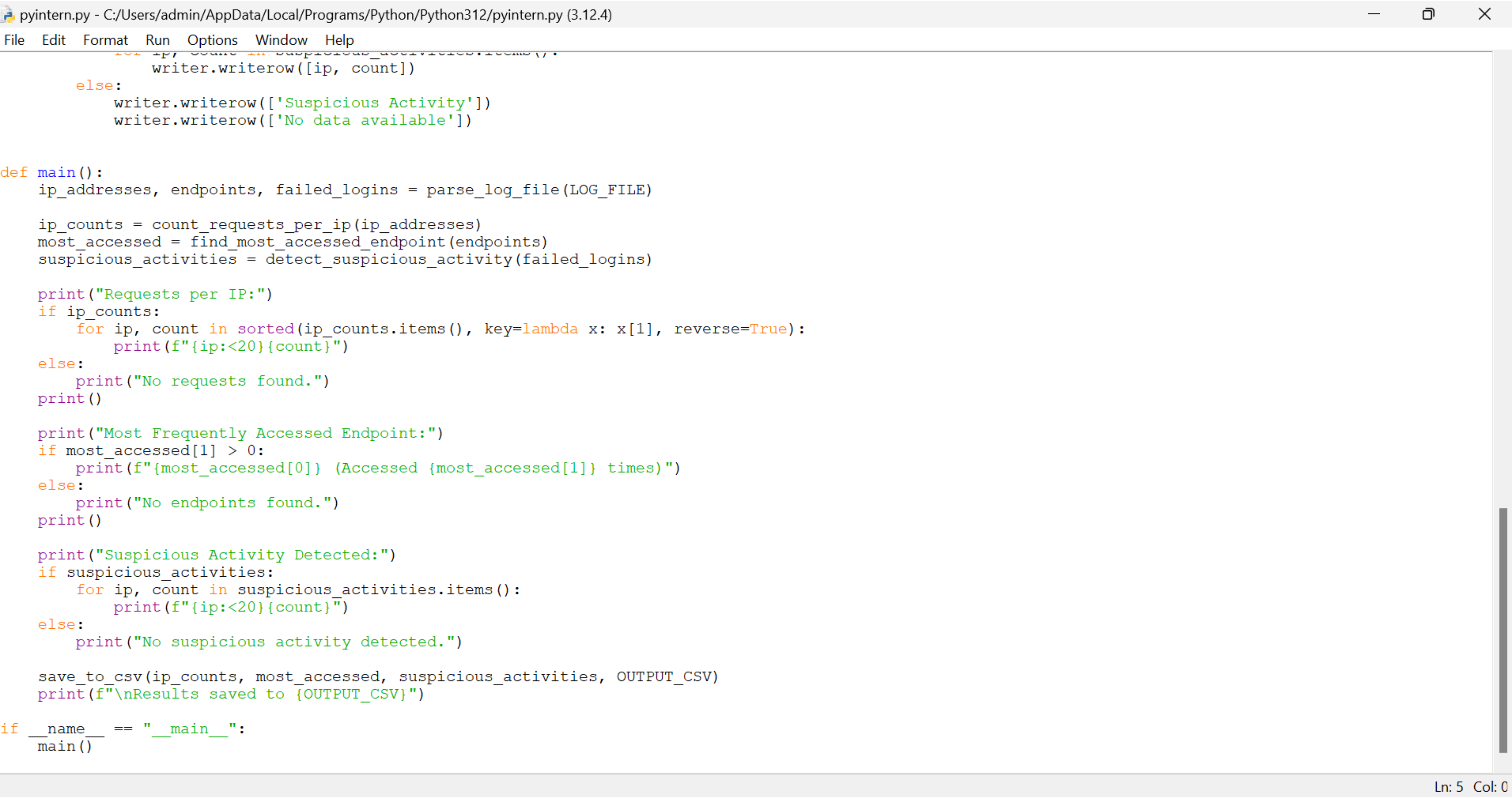

pyintern.py - C:/Users/admin/AppData/Local/Programs/Python/Python312/pyintern.py (3.12.4)
File
Edit
Format
Run
Options
Window
Help
Ln: 5
Col: 0

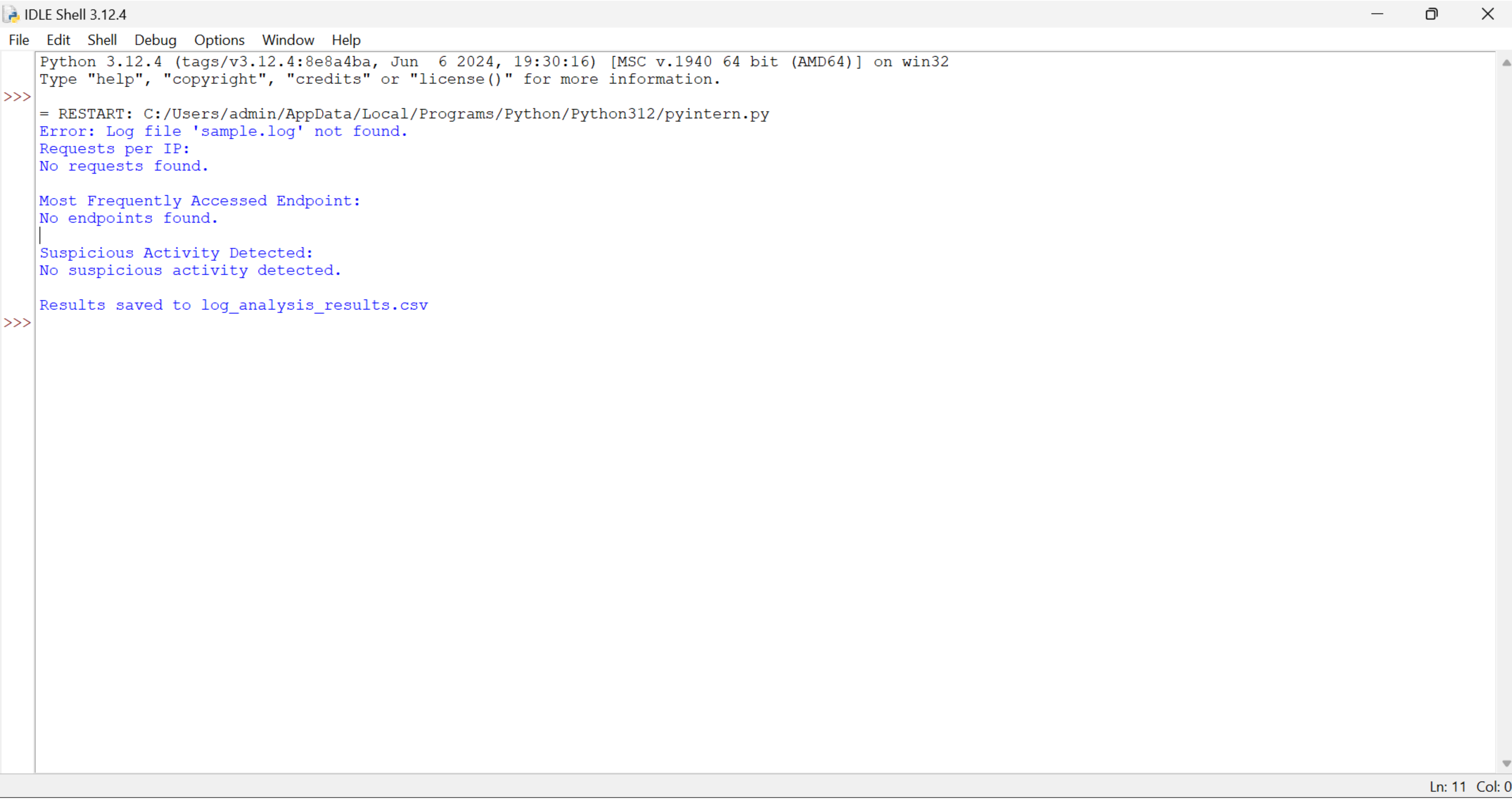
```
import csv
from collections import Counter
import re
FAILED_LOGIN_THRESHOLD = 10
LOG_FILE = 'sample.log'
OUTPUT_CSV = 'log_analysis_results.csv'

def parse_log_file(file_path):
    """Parses the log file and extracts relevant data."""
    try:
        with open(file_path, 'r') as file:
            logs = file.readlines()
    except FileNotFoundError:
        print(f"Error: Log file '{file_path}' not found.")
        return [], [], []
    except Exception as e:
        print(f"Error: Unable to read file '{file_path}'. Details: {e}")
        return [], [], []

    ip_addresses = []
    endpoints = []
    failed_logins = []

    for line in logs:
        ip_match = re.search(r'^(\d+\.\d+\.\d+\.\d+)', line)
        if ip_match:
            ip_addresses.append(ip_match.group(0))
        endpoint_match = re.search(r'"[A-Z]+ (\S+) HTTP/', line)
        if endpoint_match:
            endpoints.append(endpoint_match.group(1))

        if '401' in line or 'Invalid credentials' in line:
            if ip_match:
                failed_logins.append(ip_match.group(0))

    return ip_addresses, endpoints, failed_logins

def count_requests_per_ip(ip_addresses):
    """Counts the number of requests made by each IP address."""
    return Counter(ip_addresses)

def find_most_accessed_endpoint(endpoints):
```


pyintern.py - C:/Users/admin/AppData/Local/Programs/Python/Python312/pyintern.py (3.12.4)
File
Edit
Format
Run
Options
Window
Help
Ln: 5
Col: 0

```python
def find_most_accessed_endpoint(endpoints):
    """Finds the most frequently accessed endpoint."""
    if not endpoints:
        return "No endpoints found", 0
    endpoint_counts = Counter(endpoints)
    most_common = endpoint_counts.most_common(1)
    return most_common[0]

def detect_suspicious_activity(failed_logins):
    """Detects suspicious activity based on failed login attempts."""
    failed_login_counts = Counter(failed_logins)
    return {ip: count for ip, count in failed_login_counts.items() if count > FAILED_LOGIN_THRESHOLD}

def save_to_csv(ip_counts, most_accessed, suspicious_activities, output_file):
    """Saves the analysis results to a CSV file."""
    with open(output_file, 'w', newline='') as csvfile:
        writer = csv.writer(csvfile)
        if ip_counts:
            writer.writerow(['Requests per IP'])
            writer.writerow(['IP Address', 'Request Count'])
            for ip, count in sorted(ip_counts.items(), key=lambda x: x[1], reverse=True):
                writer.writerow([ip, count])
            writer.writerow([])
        else:
            writer.writerow(['Requests per IP'])
            writer.writerow(['No data available'])
            writer.writerow([])
        if most_accessed[1] > 0:
            writer.writerow(['Most Accessed Endpoint'])
            writer.writerow(['Endpoint', 'Access Count'])
            writer.writerow([most_accessed[0], most_accessed[1]])
            writer.writerow([])
        else:
            writer.writerow(['Most Accessed Endpoint'])
            writer.writerow(['No data available'])
            writer.writerow([])
        if suspicious_activities:
            writer.writerow(['Suspicious Activity'])
            writer.writerow(['IP Address', 'Failed Login Count'])
            for ip, count in suspicious_activities.items():
```


pyintern.py - C:/Users/admin/AppData/Local/Programs/Python/Python312/pyintern.py (3.12.4)
File Edit Format Run Options Window Help
Ln: 5 Col: 0

```
            for ip, count in suspicious_activities.items():
                writer.writerow([ip, count])
        else:
            writer.writerow(['Suspicious Activity'])
            writer.writerow(['No data available'])


def main():
    ip_addresses, endpoints, failed_logins = parse_log_file(LOG_FILE)

    ip_counts = count_requests_per_ip(ip_addresses)
    most_accessed = find_most_accessed_endpoint(endpoints)
    suspicious_activities = detect_suspicious_activity(failed_logins)

    print("Requests per IP:")
    if ip_counts:
        for ip, count in sorted(ip_counts.items(), key=lambda x: x[1], reverse=True):
            print(f"{ip:<20}{count}")
    else:
        print("No requests found.")
    print()

    print("Most Frequently Accessed Endpoint:")
    if most_accessed[1] > 0:
        print(f"{most_accessed[0]} (Accessed {most_accessed[1]} times)")
    else:
        print("No endpoints found.")
    print()

    print("Suspicious Activity Detected:")
    if suspicious_activities:
        for ip, count in suspicious_activities.items():
            print(f"{ip:<20}{count}")
    else:
        print("No suspicious activity detected.")

    save_to_csv(ip_counts, most_accessed, suspicious_activities, OUTPUT_CSV)
    print(f"\nResults saved to {OUTPUT_CSV}")

if __name__ == "__main__":
    main()
```



```
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun  6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
= RESTART: C:/Users/admin/AppData/Local/Programs/Python/Python312/pyintern.py
Error: Log file 'sample.log' not found.
Requests per IP:
No requests found.

Most Frequently Accessed Endpoint:
No endpoints found.

Suspicious Activity Detected:
No suspicious activity detected.

Results saved to log_analysis_results.csv
>>> 
```