

ISA 419: Data-Driven Security

08: Visualizing Data with Pandas

Fadel M. Megahed, PhD

Professor
Farmer School of Business
Miami University

 @FadelMegahed

 fmegahed

 fmegahed@miamioh.edu

 Automated Scheduler for Office Hours

Spring 2025

Quick Refresher of Last Week's Class

- ✓ Ensure that your imported data is **technically correct** (rename columns and fix `dtypes`)
- ✓ Understand how to change the unit of analysis by grouping and aggregating data.
- ✓ Use the `agg()` function to do aggregations on grouped data.

Learning Objectives for Today's Class

- Create quick visualizations using the `plot` method from `pandas` (with an understanding of the effect of different backends).
- Utilize `auto-viz` type plots to create a quick EDA of your data.

Plotting with Pandas

Our Data

- We will use the `merged_ips` data set from a previous class to demonstrate how to plot data in pandas.

```
import pandas as pd

toxic_ips = pd.read_csv(
    "https://raw.githubusercontent.com/fmegahed/isa419/main/data/listed_ip_90_all.csv",
    header = None, names = ['ip', 'frequency', 'lastseen']
)

geolocation = pd.read_csv(
    "https://raw.githubusercontent.com/fmegahed/isa419/main/data/ip_geolocation.csv",
    names = ['ip', 'country', 'city', 'latitude', 'longitude']
)

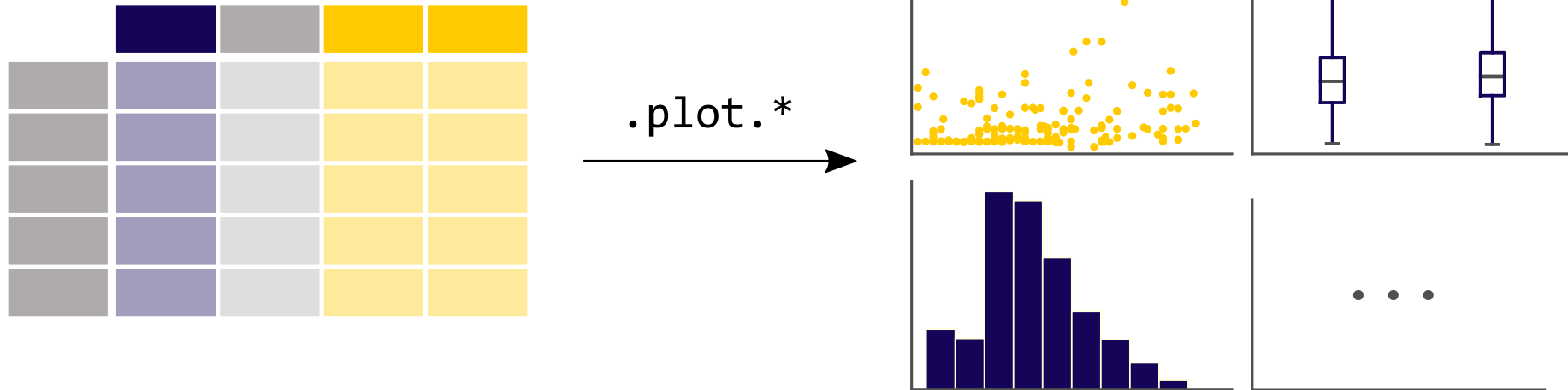
merged_ips = (
    toxic_ips
    .merge(right = geolocation, how = 'left', on = 'ip')
    .dropna()
    .assign( lastseen = lambda df: df['lastseen'].astype('datetime64[ns]') )
)

merged_ips.dtypes[0:3]
```

```
## ip                object
## frequency         int64
## lastseen          datetime64[ns]
```

Plotting with Pandas

- The `plot` method in pandas is a wrapper around `matplotlib` (by default) and is a quick way to visualize data.
- The `plot` method is available on both `Series` and `DataFrame` objects.



Class Activity to Assess your Understanding so Far

05:00


Task

Hints

Solution

- Write Python code to produce a data frame containing the total number of toxic IP frequencies by country.
- Then, identify the top 10 countries with the highest toxic IP frequencies.

Plotting with Pandas (Plot kind)



Getting started

User Guide





API reference

Development

Release notes

Search Ctrl + K

2.2 (stable) ▾



Input/output

General functions

Series

DataFrame ▾

pandas.DataFrame

pandas.DataFrame.index

pandas.DataFrame.columns

pandas.DataFrame.dtypes

pandas.DataFrame.info

pandas.DataFrame.select_dt

🏠 > API reference > DataFrame > pandas.DataF...

pandas.DataFrame.plot

[source]

DataFrame.plot(*args, **kwargs)

Make plots of Series or DataFrame.

Uses the backend specified by the option `plotting.backend`. By default, matplotlib is used.

Parameters:

data : *Series or DataFrame*

The object for which the method is called.

x : *label or position, default None*

Plotting with Pandas (Line Plot)

Data Prep:

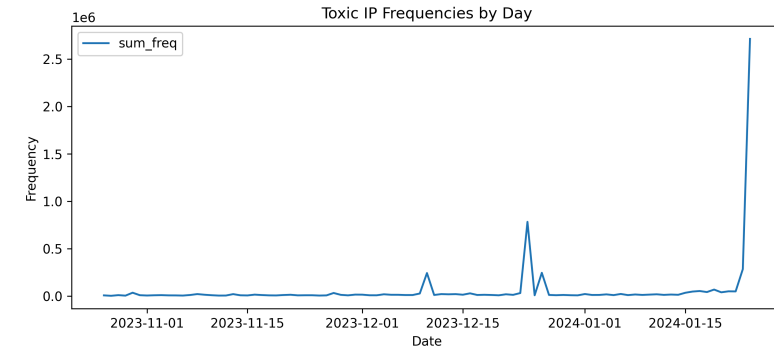
```
# Aggregating the frequencies by day
daily_freq = (
    merged_ips
    .groupby(merged_ips['lastseen'].dt.date)
    .agg(sum_freq = ('frequency', 'sum'))
    .reset_index() # to have last seen as col
    .rename(columns = {'lastseen': 'date'})
)

daily_freq.head(n=2)
```

```
##           date  sum_freq
## 0  2023-10-26      8952
## 1  2023-10-27      4642
```

Plotting:

```
daily_freq.plot(
    x = 'date', y = 'sum_freq', kind = 'line',
    title = 'Toxic IP Frequencies by Day',
    xlabel = 'Date', ylabel = 'Frequency',
    figsize = (10, 4)
)
```



Plotting with Pandas (bar Plot)

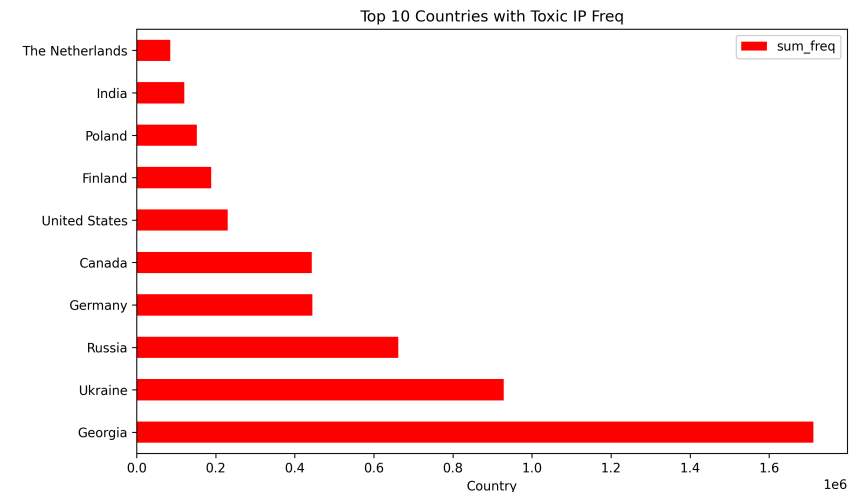
Data:

```
# Aggregating the frequencies by country
country_freq = (
    merged_ips
    .groupby('country')
    .agg(sum_freq = ('frequency', 'sum'))
    .sort_values('sum_freq', ascending = False)
    .head(10)
    .reset_index()
)
country_freq.head(n=2)
```

```
##      country  sum_freq
## 0  Georgia    1712171
## 1  Ukraine     928770
```

Plotting:

```
country_freq.plot(
    x = 'country', y = 'sum_freq', kind = 'barh',
    title = 'Top 10 Countries with Toxic IP Freq',
    xlabel = 'Country', ylabel = 'Frequency',
    figsize = (10, 6),
    color = 'red'
)
```



Plotting with Pandas (scatter Plot)

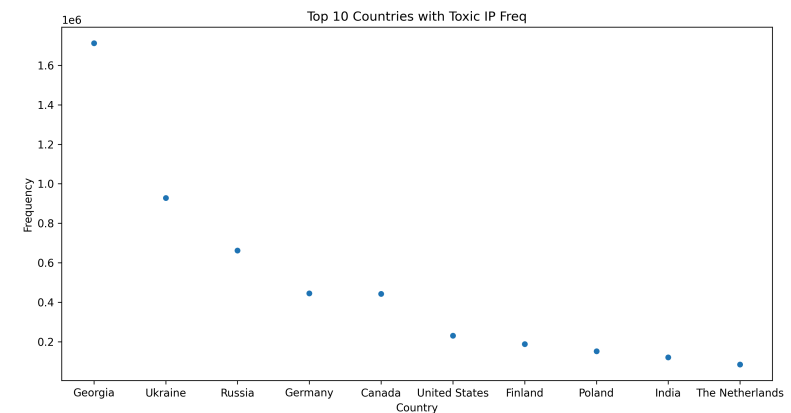
Data:

```
country_freq.head(n=5)
```

```
##   country  sum_freq
## 0  Georgia  1712171
## 1  Ukraine   928770
## 2   Russia   661849
## 3  Germany   444641
## 4   Canada   443046
```

Plotting:

```
country_freq.plot(
    # scatter plots are better with two numeric vars
    # (this example is for illustration only)
    x = 'country', y = 'sum_freq', kind = 'scatter',
    title = 'Top 10 Countries with Toxic IP Freq',
    xlabel = 'Country', ylabel = 'Frequency',
    figsize = (12, 6)
)
```



Class Activity to Assess your Understanding so Far

10:00

Task

Task 1

Task 2

Task 3

- Read the `simulated_attack_data.csv` file into a pandas data frame.
- Then, answer the questions in the next three tabs.

Automated Visualizations in Python

The `ydata-profiling` Package

Data quality profiling and exploratory data analysis (EDA) are crucial steps in any business analytics application.

`ydata-profiling` automates and standardizes the generation of detailed reports, complete with statistics and visualizations.

The significance of the package lies in how it streamlines the process of understanding and preparing data for analysis in a single line of code!

Usage of ydata-profiling

```
import pandas as pd
from ydata_profiling import ProfileReport

profile = ProfileReport(sim_attack_df, title="Pandas Profiling Report", explorative=True)

# the next line is needed since I am not using Colab for making the slides
profile.to_file("../figures/sim_attack_data_report.html")
```

Output of ydata-profiling

Pandas Profiling Report

Overview Variables Interactions Correlations Missing values Sample

Overview

Brought to you by [YData](#)

Overview

Alerts 5

[Reproduction](#)

Dataset statistics

Number of variables	11
Number of observations	1000
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0

Variable types

DateTime	1
Text	2
Categorical	2
Numeric	6

Class Activity

Task

- Identify a similar package to `ydata-profiling` in Python.
- Then, use the package to generate a report for the `merged_ips` data set.
- Share the report with your neighboring classmate.
- Discuss the insights and visualizations in your approach(es).

Recap

Summary of Main Points

By now, you should be able to do the following:

- Create quick visualizations using the `plot` method from `pandas` (with an understanding of the effect of different backends).
- Utilize `auto-viz` type plots to create a quick EDA of your data.

Review and Clarification

- **Class Notes:** Take some time to revisit your class notes for key insights and concepts.
- **Zoom Recording:** The recording of today's class will be made available on Canvas approximately 3-4 hours after the end of class.
- **Questions:** Please don't hesitate to ask for clarification on any topics discussed in class. It's crucial not to let questions accumulate.