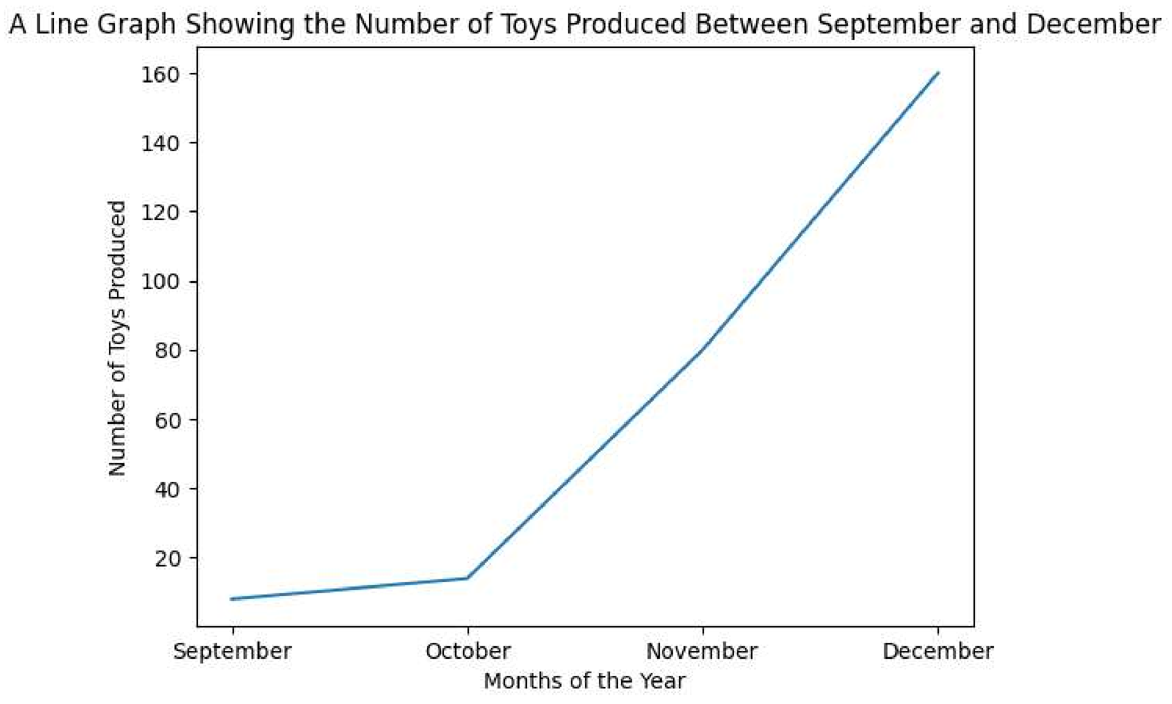
Brittany Walker

Old school web journal. Sisyphean pursuit of skill.

Advent of Cyber 2023 – Log Analysis



Yes!
I am so excited for this lab. I love Python, and I've been looking forward to an opportunity to play with some popular data science modules like Pandas and Matplotlib, both of which are featured in today's challenge.
Data Science, as explained here, is interpreting data to answer questions. It's statistics for the digital age, helping understand trends and patterns.

As TryHackMe explains, data science hinges on five responsibilities. These include:

1) Collection. This is exactly what it sounds like – where are we getting the data to analyze?
2) Processing. Also called Data Cleaning – this involves getting the collected data into a standardized format. Harder than it sounds, especially for datasets involving thousands or even millions of distinct points.
3) Mining. Analysis phase 1 – this is where we look for patterns and correlations that might give rise to more in-depth analysis. What stands out about the data already?

4) Analysis. In my opinion, the fun part! We're not only testing hypotheses, but looking for actionable insights about the data that can be communicated to stakeholders. We don't just want to know what's happening – we want to know what it means.
5) Communication and Visualization. All that hard work is worthless if we can't communicate it simply and effectively! Most of the interested parties won't be data scientists. We need to format the information we've gathered in a way that makes intuitive sense.

Data science is becoming more prominent in cybersecurity for a number of reasons. SIEMs, or Security Info and Event Management systems, collect huge amounts of data and correlate it wherever possible to identify risks and threats. A fundamental principle in developing security posture is **Risk Analysis**, where you identify the severity and likelihood of an event so that you can make an informed decision about how to prepare or respond. By using data from prior incidents or even our daily logs, we can perform threat trend analysis, develop predictive modeling, and use real information to aid in our Risk Analysis.

In our Day 2 challenge, we'll be using a Jupyter Notebook to follow along on a Python3 crash course. Then, we'll do the same for an Intro to Pandas, a Python module for handling raw data, and use Matplotlib (another module) to visualize our results.

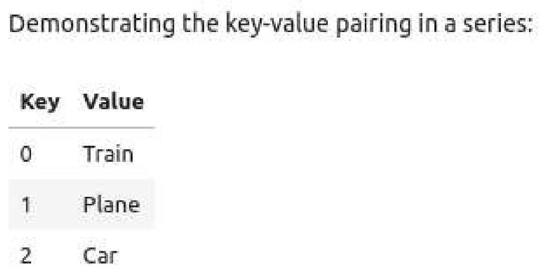
Notebook 1: Python Crash Course

This notebook contains a high-level overview of some programming concepts. These include data types, variables, and data structures. I like the comparison made between variables and labels. I've also heard variables compared to boxes, but I feel like that comparison is misleading. A variable is really a reference to memory storage – not the storage itself. Interestingly, this explains why the id() function will flag differently named functions as identical when they refer to the same object in memory. It's like having two names for the same person – they reference the same thing. TryHackMe also covers the list data structure, meant to store collections of values.

Notebook 2: Intro to Pandas

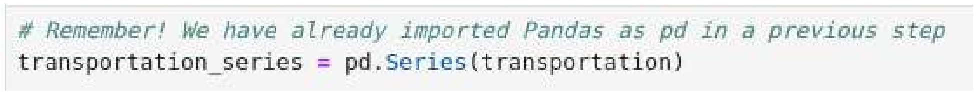
Pandas is described as a Python library designed to work with collections of related information, called data sets. Pandas allows us to perform most of the data science operations listed above, with the exception of visualization.

In Pandas, there's a data structure called a series that uses a key-value pair, similar to the dictionary data structure. It's described as being like a column in a table – one dimensional, indexed by its keys.

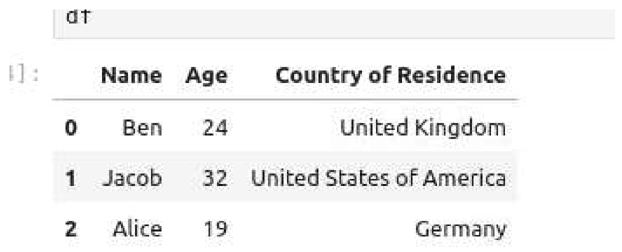

Demonstrating the key-value pairing in a series:

Key	Value
0	Train
1	Plane
2	Car

In this lab, we're converting a list into a series. We're importing the Pandas library as and using the pd.Series function to create a new series variable out of a previously defined list variable.


```
# Remember! We have already imported Pandas as pd in a previous step
transportation_series = pd.Series(transportation)
```

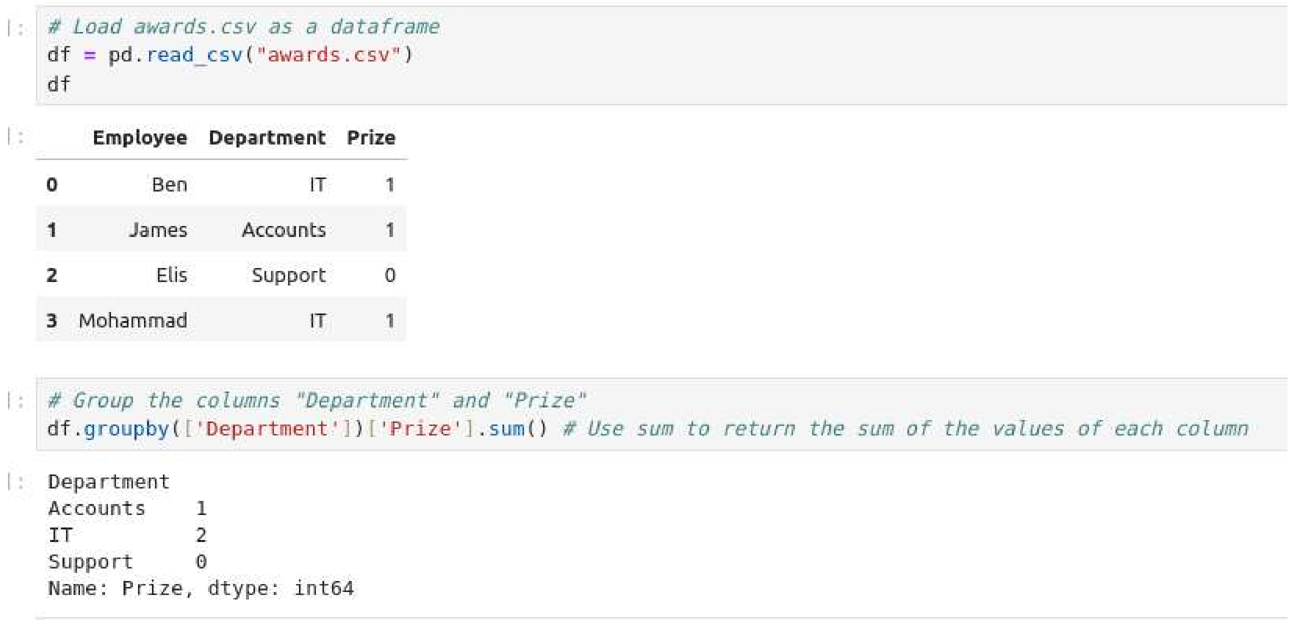
Then, we'll use Dataframes to extend the series into a grouping. Dataframes are more similar to a 2-D database with rows and columns, rather than the series, which we compared to a single column. If you've ever worked in a spreadsheet, this structure will look familiar.


```
df
```

	Name	Age	Country of Residence
0	Ben	24	United Kingdom
1	Jacob	32	United States of America
2	Alice	19	Germany

We can return a single row from the Dataframe by using the Pandas df.loc[] and passing the index number of the row. These rows are zero-ordered, meaning they start at zero instead of one, as pictured above.

Still covering Pandas, we get to grouping. Grouping allows us to group data into categories for analysis, like comparisons. You can read a csv file directly into a dataframe (be still, my heart!) and use GroupBy() to sum the values in each column.



```python
# Load awards.csv as a dataframe
df = pd.read_csv("awards.csv")
df
```

	Employee	Department	Prize
0	Ben	IT	1
1	James	Accounts	1
2	Elis	Support	0
3	Mohammad	IT	1

```python
# Group the columns "Department" and "Prize"
df.groupby(['Department'])['Prize'].sum() # Use sum to return the sum of the values of each column
```

```
Department
Accounts    1
IT          2
Support     0
Name: Prize, dtype: int64
```

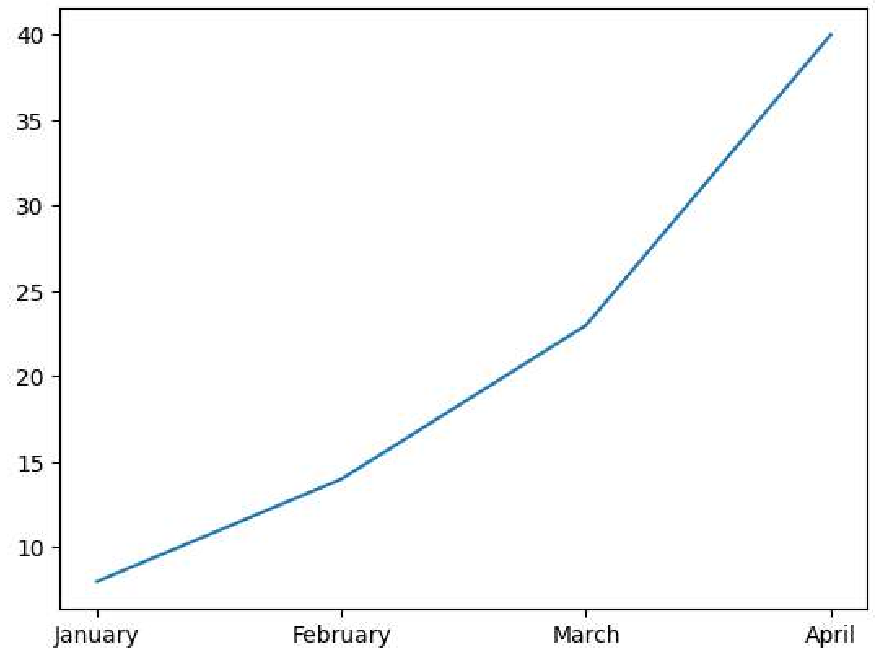
You can even use the df.groupby().describe() method to summarize the data with common metrics like count, mean, standard deviation, and quartiles.

Notebook 3: Intro to Matplotlib

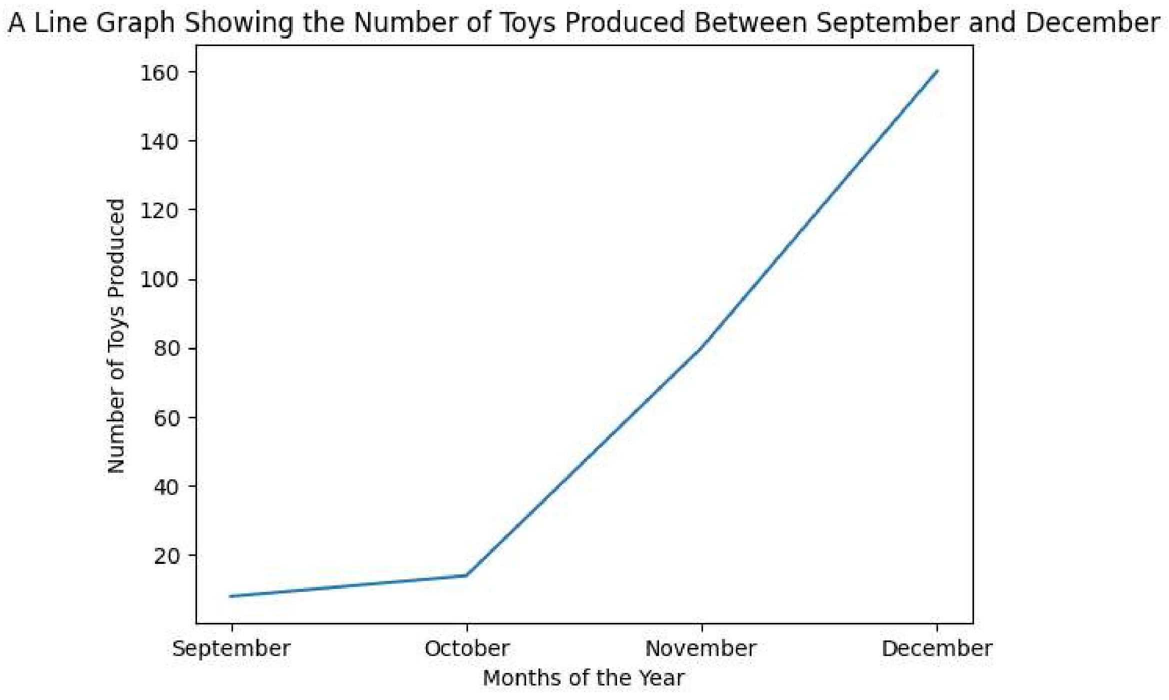
Reads just like it's spelled, a rare gem. Matplotlib is used to create visualizations for data. It defaults to a line graph, but you have so many more options than that. It also defaults to opening the plot, or graph, in a new window. Normally fine, but our challenge creators did a little wizardry to keep it inline for the Jupyter notebook. We import matplotlib.pyplot as plt (imagine writing the former every time..) and pass some values to the plot() method in a simple array to get started.

```
4]: # We'll start by making a very simple plot using an array before we get into the nitty-gritty.
    # Remembering it goes X -> Y. Along the corridor and up the stairs!
    plt.plot(['January', 'February', 'March', 'April' ],[8,14,23,40])
```

```
4]: [<matplotlib.lines.Line2D at 0x7f065748c730>]
```



Beautiful! Then we use the plt.ylabel, plt.xlabel, and plt.title to make our graph easier to comprehend. Have you forgotten that we're trying to save Christmas?



I actually really wish that I had known how straightforward these modules were. I spend a lot of time working with .csv files for my day job, and Pandas looks way more intuitive than the Python CSV module – not least because you can reference text column titles rather than indexing.

Onwards and upwards! We create a bar graph using all the skills we just covered and select a beautiful sky blue for the color. Nice.

Notebook 4: Capstone

Now for the security part of our challenge! We're going to use our new skills to analyze some captured packet traffic. The quick rundown is that network traffic is broken down into packets containing information like their source, destination, and of course the contents. These packets can be analyzed by network tools like Wireshark for insight into malicious activity. Today, though, we're going to do it from scratch.

The Jupyter notebook convienently opens by importing the necessary libraries and reading a .csv packet capture file into a dataframe named df.

We use print(df.count()) to sum the total number of packets in the file. Then, to figure out which IP address sent the most traffic, we use df.groupby(['Source']).size() to count the number of times each address shows up as a source in the file. Some of them are pretty close, but we have a clear winner.
Lastly, we'll use a similar counting operation on the Protocol column to see what the most frequent protocol was. I accidentally typed value.counts initially – that wasn't it.

We're looking for df.value_counts(['Protocol']). I'm not sure if it's got a built in sorting function, but our answer was conveniently at the top of the list. I will absolutely be playing around with these modules again very soon.

Did you follow along on the site? Tell me if you hit any roadblocks (or typos!) in the comments.

Sponsored Content

Oklahoma Find Out How Others in Oklahoma City Are Getting Weed Delivered
HelloMood | Sponsored

Forget The Blue Pill, Use This Household Food To Fight ED
urologytip.pro | Sponsored

If You Need To Kill Time On Your Computer, This Vintage Game Is A Must-...
Forge of Empires | Sponsored

Cheer Pics That Make Us Sweat
FollowSports | Sponsored

4 Signs Telling That Parasites Are Living Inside Your Body!
medicine24today.com | Sponsored

Find Postage Solutions: Search Post Office Stamps Here
Top Searches Go | Sponsored

The Worst Way to Withdraw From Retirement Accounts
smartasset.com | Sponsored

Forget Furosemide, Use This Household Item To Help Drain Edema Fluid
healthlabnews.com | Sponsored

Hyundai's New Santa Fe SUV Is Turning Heads (See Prices)
Commonsearches | Search Ads | Sponsored

bumblB33 **December 13, 2023** **Uncategorized** **Edit**

Published by bumblB33

Painter & Pythonista. I love cybersecurity, and I'm currently self-teaching. Started writing to get thoughts out and talk projects. Drop me a line! **View more posts**

Leave a Reply