Learn Alien Vault Ip Reputation Database

Lab-1

CYT-250Threat Investigation

Elaborate by:

Leandro Delgado

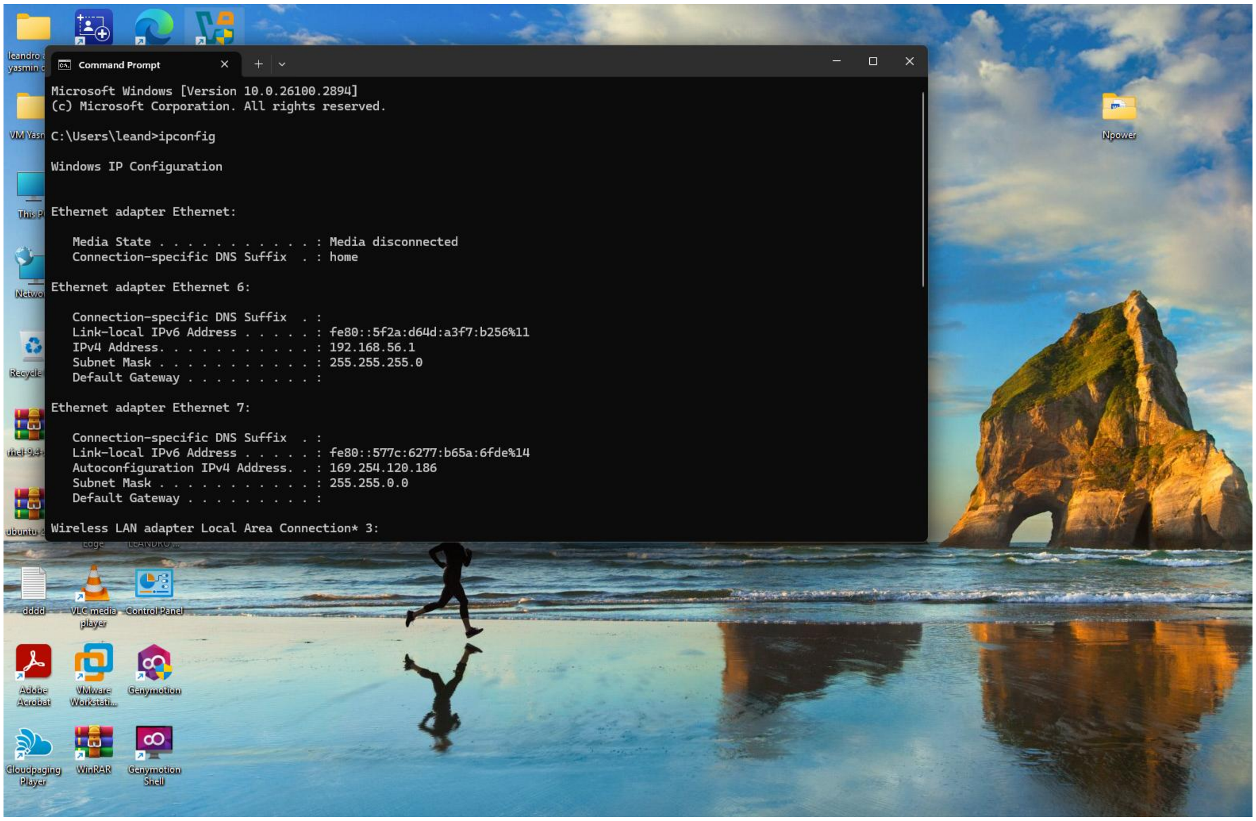
Student Number: 114416241

Professor: Tatiana Outkina

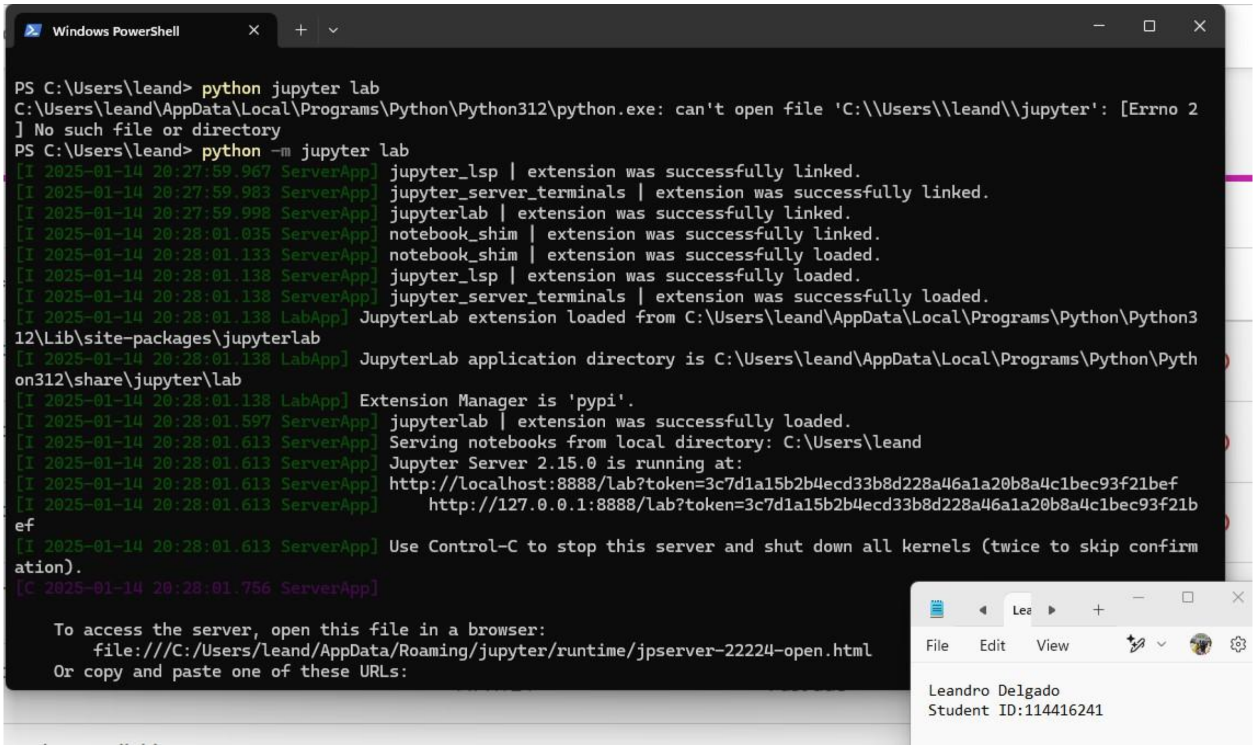
CYT245 Lab1 1. Learn AlienVault IP Reputation database – 4%

Individual task

Preparation – 0-Screen. At the start, make screenshot of the starting screen. The screenshot must contain indication of the laptop ownership (like user name).

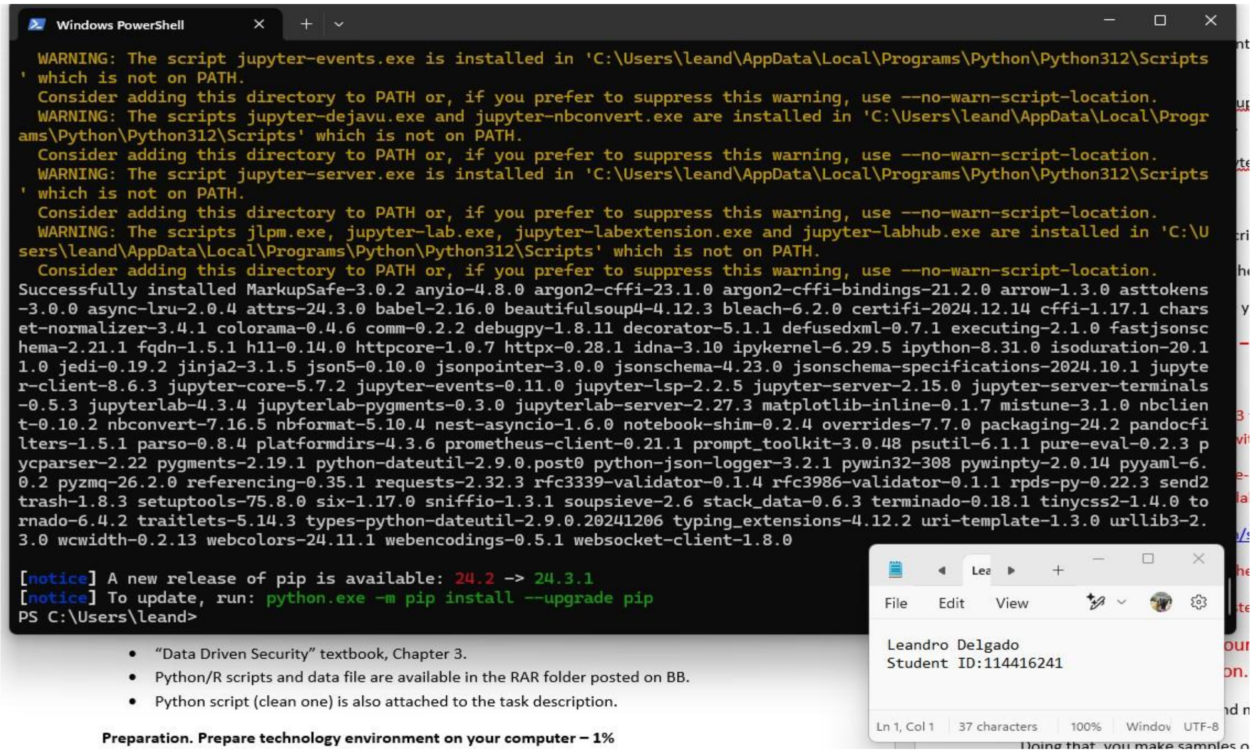


```
Microsoft Windows [Version 10.0.26100.2894]
(c) Microsoft Corporation. All rights reserved.

C:\Users\leand>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : home

Ethernet adapter Ethernet 6:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::5f2a:d64d:a3f7:b256%11
   IPv4 Address. . . . . . . . . . . : 192.168.56.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Ethernet adapter Ethernet 7:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::577c:6277:b65a:6fde%14
   Autoconfiguration IPv4 Address. . : 169.254.120.186
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . :

Wireless LAN adapter Local Area Connection* 3:
```



```
PS C:\Users\leand> python jupyter lab
C:\Users\leand\AppData\Local\Programs\Python\Python312\python.exe: can't open file 'C:\\Users\\leand\\jupyter': [Errno 2
] No such file or directory
PS C:\Users\leand> python -m jupyter lab
[I 2025-01-14 20:27:59.967 ServerApp] jupyter_lsp | extension was successfully linked.
[I 2025-01-14 20:27:59.983 ServerApp] jupyter_server_terminals | extension was successfully linked.
[I 2025-01-14 20:27:59.998 ServerApp] jupyterlab | extension was successfully linked.
[I 2025-01-14 20:28:01.035 ServerApp] notebook_shim | extension was successfully linked.
[I 2025-01-14 20:28:01.133 ServerApp] notebook_shim | extension was successfully loaded.
[I 2025-01-14 20:28:01.138 ServerApp] jupyter_lsp | extension was successfully loaded.
[I 2025-01-14 20:28:01.138 ServerApp] jupyter_server_terminals | extension was successfully loaded.
[I 2025-01-14 20:28:01.138 LabApp] JupyterLab extension loaded from C:\Users\leand\AppData\Local\Programs\Python\Python3
12\Lib\site-packages\jupyterlab
[I 2025-01-14 20:28:01.138 LabApp] JupyterLab application directory is C:\Users\leand\AppData\Local\Programs\Python\Pyth
on312\share\jupyter\lab
[I 2025-01-14 20:28:01.138 LabApp] Extension Manager is 'pypi'.
[I 2025-01-14 20:28:01.597 ServerApp] jupyterlab | extension was successfully loaded.
[I 2025-01-14 20:28:01.613 ServerApp] Serving notebooks from local directory: C:\Users\leand
[I 2025-01-14 20:28:01.613 ServerApp] Jupyter Server 2.15.0 is running at:
[I 2025-01-14 20:28:01.613 ServerApp] http://localhost:8888/lab?token=3c7d1a15b2b4ecd33b8d228a46a1a20b8a4c1bec93f21bef
[I 2025-01-14 20:28:01.613 ServerApp]     http://127.0.0.1:8888/lab?token=3c7d1a15b2b4ecd33b8d228a46a1a20b8a4c1bec93f21b
ef
[I 2025-01-14 20:28:01.613 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirm
ation).
[C 2025-01-14 20:28:01.756 ServerApp]

    To access the server, open this file in a browser:
        file:///C:/Users/leand/AppData/Roaming/jupyter/runtime/jpserver-22224-open.html
    Or copy and paste one of these URLs:
```

```
Leandro Delgado
Student ID:114416241
```



```
  WARNING: The script jupyter-events.exe is installed in 'C:\Users\leand\AppData\Local\Programs\Python\Python312\Scripts
' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
  WARNING: The scripts jupyter-dejavu.exe and jupyter-nbconvert.exe are installed in 'C:\Users\leand\AppData\Local\Progr
ams\Python\Python312\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
  WARNING: The script jupyter-server.exe is installed in 'C:\Users\leand\AppData\Local\Programs\Python\Python312\Scripts
' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
  WARNING: The scripts jlpm.exe, jupyter-lab.exe, jupyter-labextension.exe and jupyter-labhub.exe are installed in 'C:\U
sers\leand\AppData\Local\Programs\Python\Python312\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed MarkupSafe-3.0.2 anyio-4.8.0 argon2-cffi-23.1.0 argon2-cffi-bindings-21.2.0 arrow-1.3.0 asttokens
-3.0.0 async-lru-2.0.4 attrs-24.3.0 babel-2.16.0 beautifulsoup4-4.12.3 bleach-6.2.0 certifi-2024.12.14 cffi-1.17.1 chars
et-normalizer-3.4.1 colorama-0.4.6 comm-0.2.2 debugpy-1.8.11 decorator-5.1.1 defusedxml-0.7.1 executing-2.1.0 fastjsonsc
hema-2.21.1 fqdn-1.5.1 h11-0.14.0 httpcore-1.0.7 httpx-0.28.1 idna-3.10 ipykernel-6.29.5 ipython-8.31.0 isoduration-20.1
1.0 jedi-0.19.2 jinja2-3.1.5 json5-0.10.0 jsonpointer-3.0.0 jsonschema-4.23.0 jsonschema-specifications-2024.10.1 jupyte
r-client-8.6.3 jupyter-core-5.7.2 jupyter-events-0.11.0 jupyter-lsp-2.2.5 jupyter-server-2.15.0 jupyter-server-terminals
-0.5.3 jupyterlab-4.3.4 jupyterlab-pygments-0.3.0 jupyterlab-server-2.27.3 matplotlib-inline-0.1.7 mistune-3.1.0 nbclien
t-0.10.2 nbconvert-7.16.5 nbformat-5.10.4 nest-asyncio-1.6.0 notebook-shim-0.2.4 overrides-7.7.0 packaging-24.2 pandocfi
lters-1.5.1 parso-0.8.4 platformdirs-4.3.6 prometheus-client-0.21.1 prompt_toolkit-3.0.48 psutil-6.1.1 pure-eval-0.2.3 p
ycparser-2.22 pygments-2.19.1 python-dateutil-2.9.0.post0 python-json-logger-3.2.1 pywin32-308 pywinpty-2.0.14 pyyaml-6.
0.2 pyzmq-26.2.0 referencing-0.35.1 requests-2.32.3 rfc3339-validator-0.1.4 rfc3986-validator-0.1.1 rpds-py-0.22.3 send2
trash-1.8.3 setuptools-75.8.0 six-1.17.0 sniffio-1.3.1 soupsieve-2.6 stack_data-0.6.3 terminado-0.18.1 tinycss2-1.4.0 to
rnado-6.4.2 traitlets-5.14.3 types-python-dateutil-2.9.0.20241206 typing_extensions-4.12.2 uri-template-1.3.0 urllib3-2.
3.0 wcwidth-0.2.13 webcolors-24.11.1 webencodings-0.5.1 websocket-client-1.8.0

[notice] A new release of pip is available: 24.2 -> 24.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\leand>
```

```
Leandro Delgado
Student ID:114416241
```

- "Data Driven Security" textbook, Chapter 3.
- Python/R scripts and data file are available in the RAR folder posted on BB.
- Python script (clean one) is also attached to the task description.

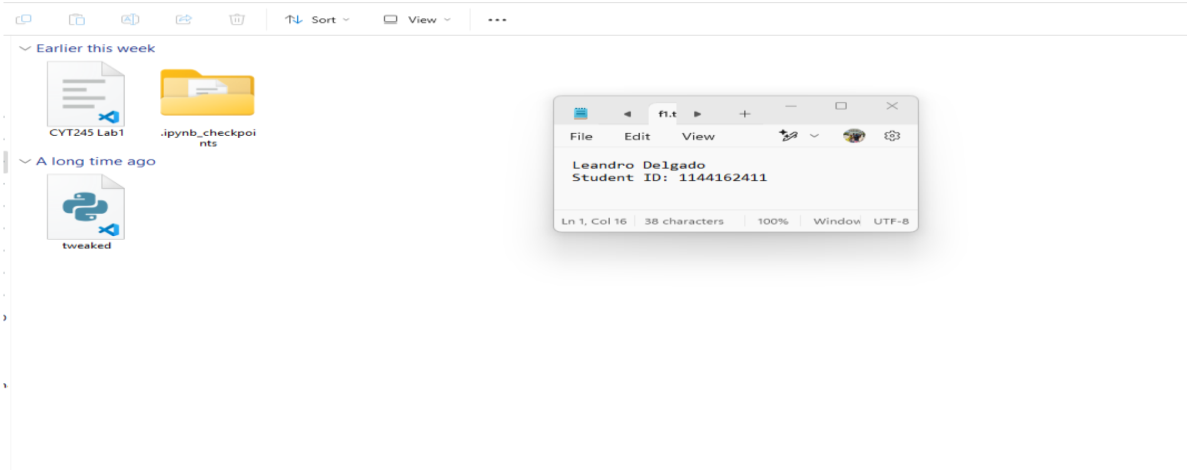
Preparation. Prepare technology environment on your computer – 1%

Objective:

Lab Focus	Tools Used	Activities	Goals
AlienVault IP Reputation database	Python, Pandas	Set up environment, run Python scripts, explore database	Generate statistics, visualize data, understand data in cybersecurity context
Threat intelligence			Analyze and interpret reputation data to spot potential threats

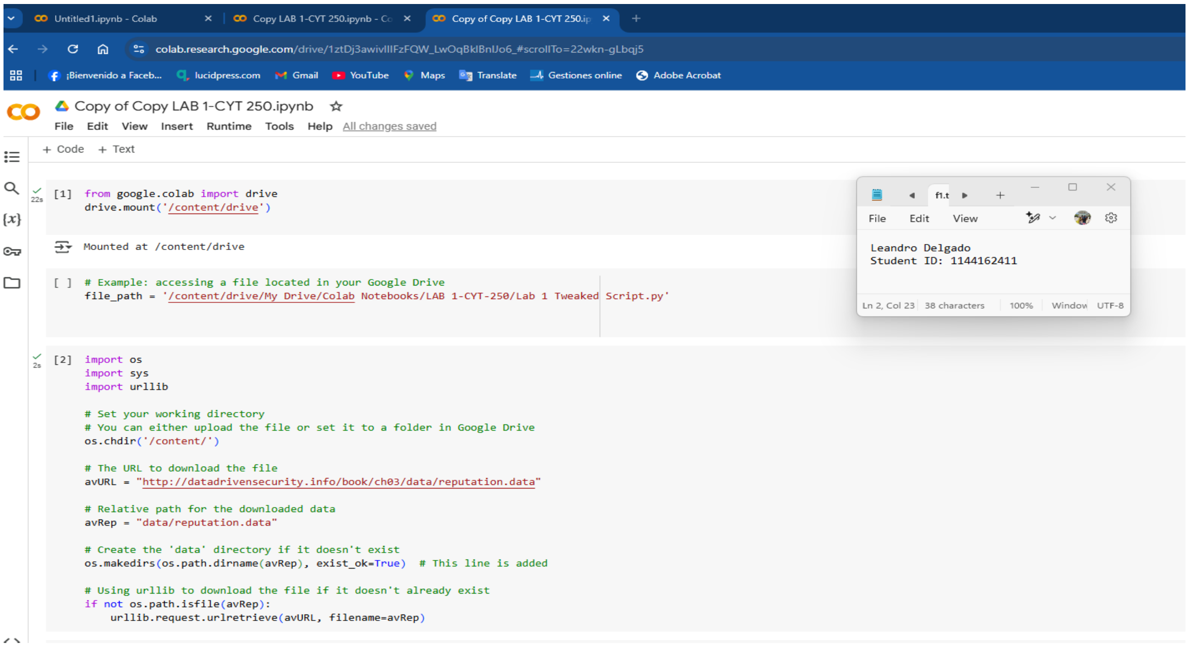
- **Step 1. Unzip the book.rar and move the folder book to your Anaconda environment.**

To proceed to reach this task, I have downloaded the Script provided by the lab document. I extracted the compressed file and paste into the directory



```
Leandro Delgado
Student ID: 1144162411
```

- **Step 2. Open the Python script file and run Listing 1 portion in your notebook Google Collab).**


Untitled1.ipynb - Colab
Copy LAB 1-CYT 250.ipynb - Co
Copy of Copy LAB 1-CYT 250.ip
colab.research.google.com/drive/1ztDj3awivlIIFzFQW_LwOqBklBnlJo6_#scrollTo=22wkn-gLbqj5
¡Bienvenido a Faceb...
lucidpress.com
Gmail
YouTube
Maps
Translate
Gestiones online
Adobe Acrobat
Copy of Copy LAB 1-CYT 250.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
f1.t
File Edit View
Ln 2, Col 23 38 characters 100% Window UTF-8

```
[1] from google.colab import drive
    drive.mount('/content/drive')

    Mounted at /content/drive

[ ] # Example: accessing a file located in your Google Drive
    file_path = '/content/drive/My Drive/Colab Notebooks/LAB 1-CYT-250/Lab 1 Tweaked Script.py'

[2] import os
    import sys
    import urllib

    # Set your working directory
    # You can either upload the file or set it to a folder in Google Drive
    os.chdir('/content/')

    # The URL to download the file
    avURL = "http://datadrivensecurity.info/book/ch03/data/reputation.data"

    # Relative path for the downloaded data
    avRep = "data/reputation.data"

    # Create the 'data' directory if it doesn't exist
    os.makedirs(os.path.dirname(avRep), exist_ok=True)  # This line is added

    # Using urllib to download the file if it doesn't already exist
    if not os.path.isfile(avRep):
        urllib.request.urlretrieve(avURL, filename=avRep)
```

```
Leandro Delgado
Student ID: 1144162411
```

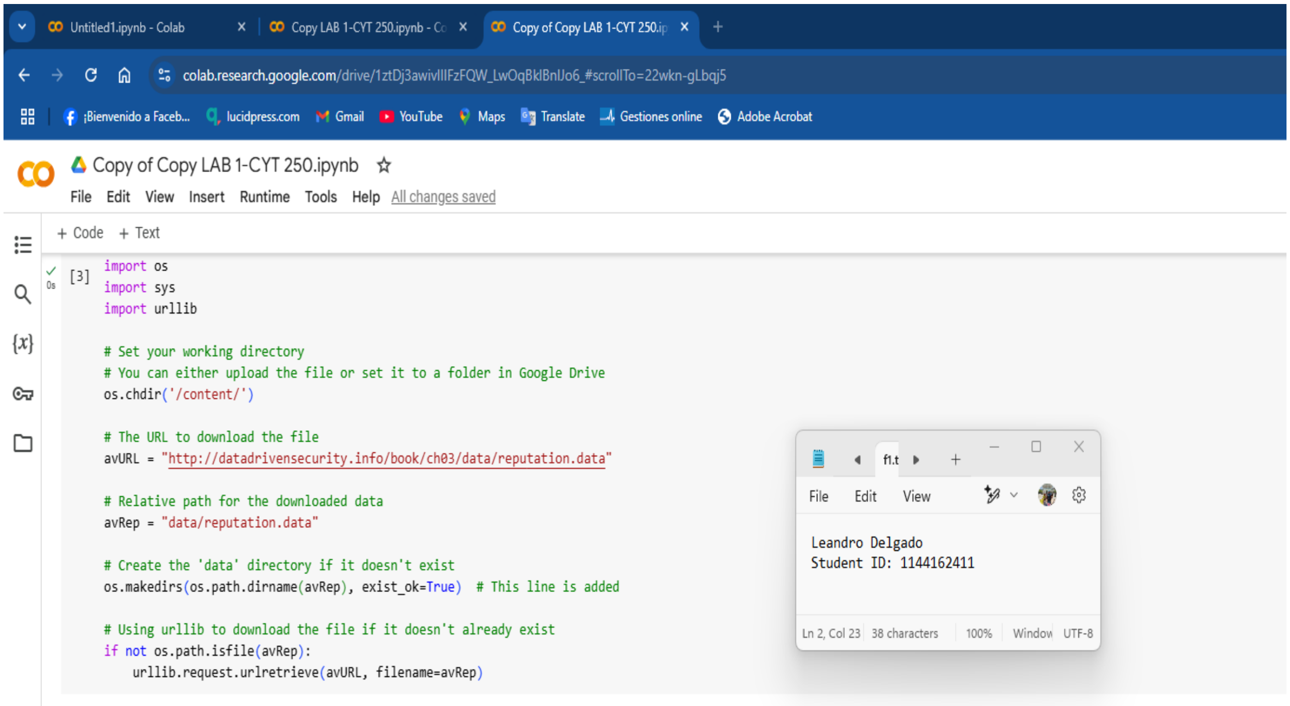
- **Step 3. Run the Listing 3-3. You set relative path for the downloaded data.**

# Listing 3-1

Import os

Import sys

Os.chdir (os.path.expanduser (“~”0 + path). Once the 'Reputation Data' file was downloaded, a folder named 'dataDrivenSecurity_book' was created to store the script. Afterward, I proceeded to execute the Python.


Untitled1.ipynb - Colab
Copy LAB 1-CYT 250.ipynb - Co
Copy of Copy LAB 1-CYT 250.ip
colab.research.google.com/drive/1ztDj3awivlIIFzFQW_LwOqBklBnlJo6_#scrollTo=22wkn-gLbqj5
¡Bienvenido a Faceb...
lucidpress.com
Gmail
YouTube
Maps
Translate
Gestiones online
Adobe Acrobat
Copy of Copy LAB 1-CYT 250.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
f1.t
File Edit View
Ln 2, Col 23 38 characters 100% Window UTF-8

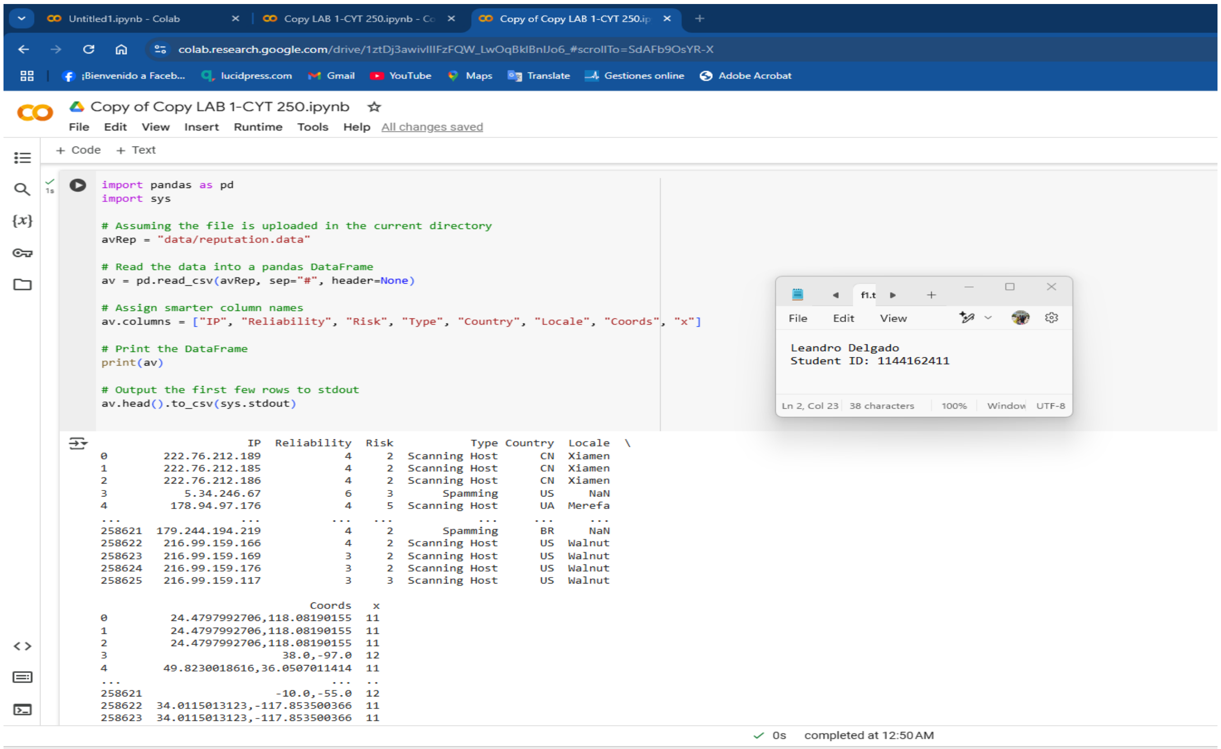
```
[3] import os
    import sys
    import urllib

    # Set your working directory
    # You can either upload the file or set it to a folder in Google Drive
    os.chdir('/content/')

    # The URL to download the file
    avURL = "http://datadrivensecurity.info/book/ch03/data/reputation.data"

    # Relative path for the downloaded data
    avRep = "data/reputation.data"

    # Create the 'data' directory if it doesn't exist
    os.makedirs(os.path.dirname(avRep), exist_ok=True)  # This line is added

    # Using urllib to download the file if it doesn't already exist
    if not os.path.isfile(avRep):
        urllib.request.urlretrieve(avURL, filename=avRep)
```

```
Leandro Delgado
Student ID: 1144162411
```

Step 4. Run Listing 3-5. At this point of time, you will obtain the result showing first 5 rows from the file.

This code defines the structure of IP Reputation Database. Run the code and observe the result.

Answer the following questions:


Untitled1.ipynb - Colab
Copy LAB 1-CYT 250.ipynb - Co
Copy of Copy LAB 1-CYT 250.ip
colab.research.google.com/drive/1ztDj3awivIIlFzFQW_LwOqBkIBnIJo6_#scrollTo=SdAFb9OsYR-X
¡Bienvenido a Faceb...
lucidpress.com
Gmail
YouTube
Maps
Translate
Gestiones online
Adobe Acrobat
Copy of Copy LAB 1-CYT 250.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
f1.t
File Edit View
Ln 2, Col 23 38 characters 100% Window UTF-8
0s completed at 12:50 AM

```python
import pandas as pd
import sys

# Assuming the file is uploaded in the current directory
avRep = "data/reputation.data"

# Read the data into a pandas DataFrame
av = pd.read_csv(avRep, sep="#", header=None)

# Assign smarter column names
av.columns = ["IP", "Reliability", "Risk", "Type", "Country", "Locale", "Coords", "x"]

# Print the DataFrame
print(av)

# Output the first few rows to stdout
av.head().to_csv(sys.stdout)
```

```
Leandro Delgado
Student ID: 1144162411
```

```
                    IP  Reliability  Risk            Type Country  Locale  \
0       222.76.212.189            4     2  Scanning Host      CN  Xiamen
1       222.76.212.185            4     2  Scanning Host      CN  Xiamen
2       222.76.212.186            4     2  Scanning Host      CN  Xiamen
3          5.34.246.67            6     3        Spamming      US     NaN
4        178.94.97.176            4     5  Scanning Host      UA  Merefa
...                ...          ...   ...             ...     ...     ...
258621  179.244.194.219            4     2        Spamming      BR     NaN
258622   216.99.159.166            4     2  Scanning Host      US  Walnut
258623   216.99.159.169            3     2  Scanning Host      US  Walnut
258624   216.99.159.176            3     2  Scanning Host      US  Walnut
258625   216.99.159.117            3     3  Scanning Host      US  Walnut

                            Coords   x
0        24.4797992706,118.08190155  11
1        24.4797992706,118.08190155  11
2        24.4797992706,118.08190155  11
3                      38.0,-97.0  12
4       49.8230018616,36.0507011414  11
...                            ...  ..
258621                  -10.0,-55.0  12
258622  34.0115013123,-117.853500366  11
258623  34.0115013123,-117.853500366  11
```

1. **What is Pandas name for the IP Reputation Database csv file?**

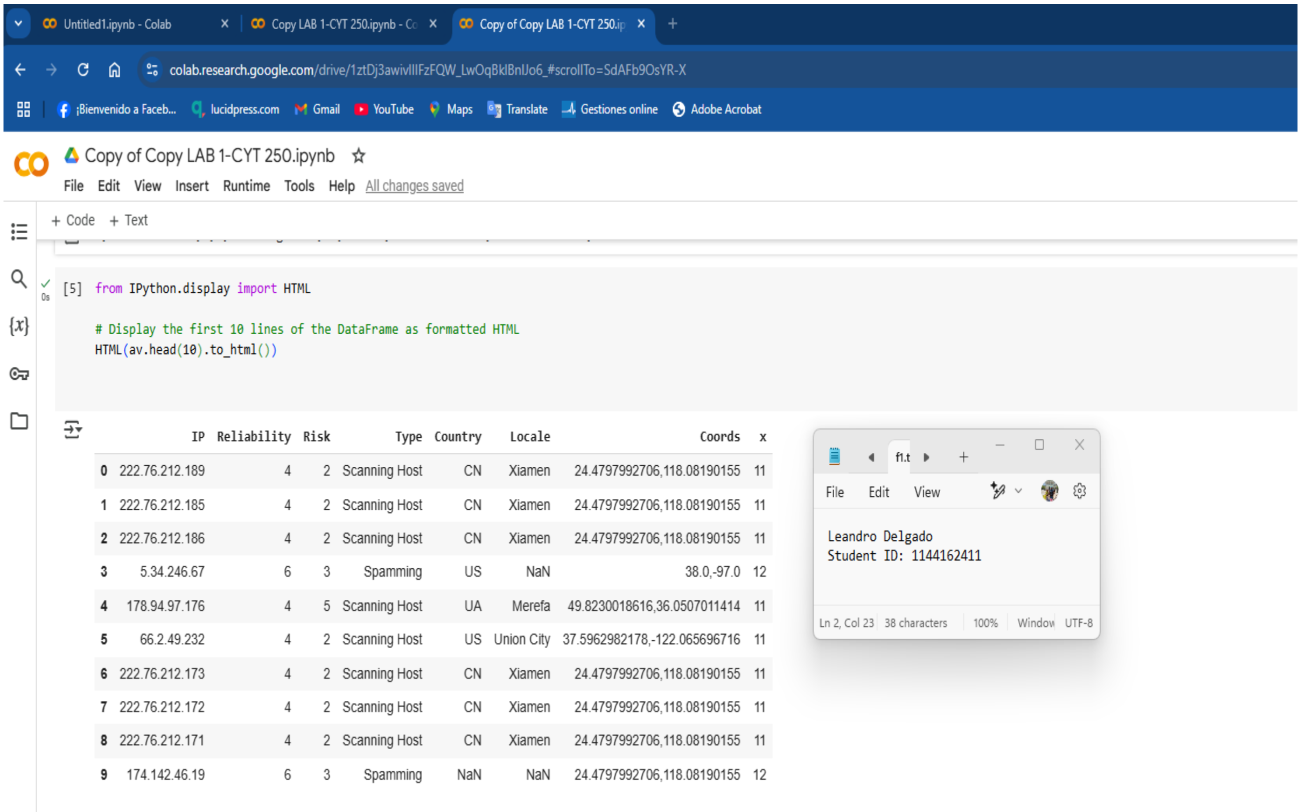
   The CSV file for the IP Reputation Database is stored in a variable called **"av"** in Pandas.

2. **What are Columns names of the Pandas data frame?**

   The Pandas DataFrame includes the following column names: IP, Reliability, Risk, Type, Country, Locale, Coords, x.

Step 5. Run Listing 3-6. You will see HTML formatted output of the same data frame.

To see, the HTML formatted outputof the data frame, I executed the following script:


Untitled1.ipynb - Colab
Copy LAB 1-CYT 250.ipynb - Co
Copy of Copy LAB 1-CYT 250.ip
colab.research.google.com/drive/1ztDj3awivlllFzFQW_LwOqBklBnlJo6_#scrollTo=SdAFb9OsYR-X
¡Bienvenido a Faceb...
lucidpress.com
Gmail
YouTube
Maps
Translate
Gestiones online
Adobe Acrobat
Copy of Copy LAB 1-CYT 250.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
[5]
f1.t
File Edit View
Ln 2, Col 23 38 characters 100% Window UTF-8

```
from IPython.display import HTML

# Display the first 10 lines of the DataFrame as formatted HTML
HTML(av.head(10).to_html())
```

	IP	Reliability	Risk	Type	Country	Locale	Coords	x
0	222.76.212.189	4	2	Scanning Host	CN	Xiamen	24.4797992706,118.08190155	11
1	222.76.212.185	4	2	Scanning Host	CN	Xiamen	24.4797992706,118.08190155	11
2	222.76.212.186	4	2	Scanning Host	CN	Xiamen	24.4797992706,118.08190155	11
3	5.34.246.67	6	3	Spamming	US	NaN	38.0,-97.0	12
4	178.94.97.176	4	5	Scanning Host	UA	Merefa	49.8230018616,36.0507011414	11
5	66.2.49.232	4	2	Scanning Host	US	Union City	37.5962982178,-122.065696716	11
6	222.76.212.173	4	2	Scanning Host	CN	Xiamen	24.4797992706,118.08190155	11
7	222.76.212.172	4	2	Scanning Host	CN	Xiamen	24.4797992706,118.08190155	11
8	222.76.212.171	4	2	Scanning Host	CN	Xiamen	24.4797992706,118.08190155	11
9	174.142.46.19	6	3	Spamming	NaN	NaN	24.4797992706,118.08190155	12

Leandro Delgado
Student ID: 1144162411

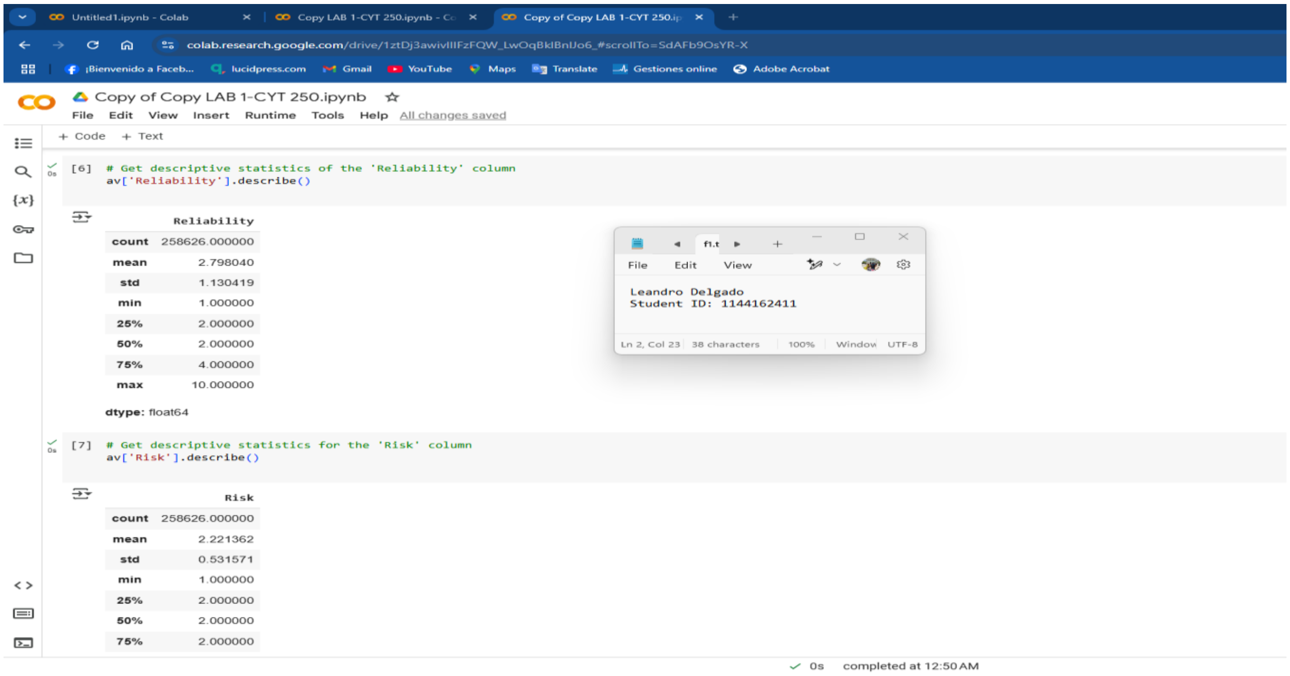
Question:

1. **What are Python code line lines that allow doing so (copy and paste from the code)**

The line of code displayed the first 10 Row of the Data Frame Av as formatted HTML. The code showed is **HTML (av.head(10).to_html( )).**

Step 6. Run Listing 3-8. You are now start exploring data. This portion of code demonstrates understanding of quantitative category of data,

So, basically, this data has values that I can use for calculations. To make sense of it, I need to calculate some basic 'descriptive statistics. These stats will help us report and visualize the data better.



```
[6]  # Get descriptive statistics of the 'Reliability' column
     av['Reliability'].describe()
```

```
         Reliability
count  258626.000000
mean        2.798040
std         1.130419
min         1.000000
25%         2.000000
50%         2.000000
75%         4.000000
max        10.000000

dtype: float64
```

```
Leandro Delgado
Student ID: 1144162411
```

```
[7]  # Get descriptive statistics for the 'Risk' column
     av['Risk'].describe()
```

```
                Risk
count  258626.000000
mean        2.221362
std         0.531571
min         1.000000
25%         2.000000
50%         2.000000
75%         2.000000
```

✓ 0s completed at 12:50 AM

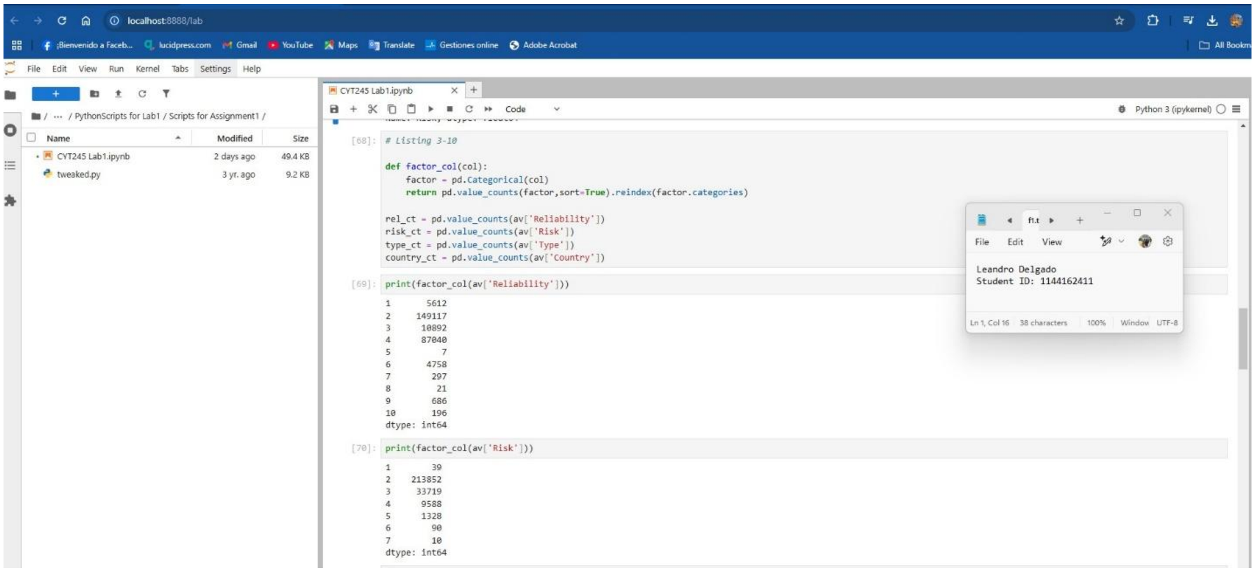
Answer the following questions:

1. What is the Pandas function to generate descriptive statistics?

The panda Function to generate Descriptive Statistics for a DataFrame or Series is. Av [‘Reliability’]. Describe (). These lines of code will calculate basic statistics (such as count, mean, standard deviation, minimum value, 25th percentile, 50th percentile, 75th percentile, and maximum value) for the 'Reliability' and 'Risk' columns in the DataFrame **av**."

Step 7. the number of malicious nodes calculated by Reliability, Risk, Type, and Country separately. With the last outcome you can see the number of malicious nodes by Country.

Next, to better understand qualitative data in Pandas, we run the next script' to analyze malicious nodes across different categories



```
[68]: # Listing 3-10

      def factor_col(col):
          factor = pd.Categorical(col)
          return pd.value_counts(factor,sort=True).reindex(factor.categories)

      rel_ct = pd.value_counts(av['Reliability'])
      risk_ct = pd.value_counts(av['Risk'])
      type_ct = pd.value_counts(av['Type'])
      country_ct = pd.value_counts(av['Country'])
```

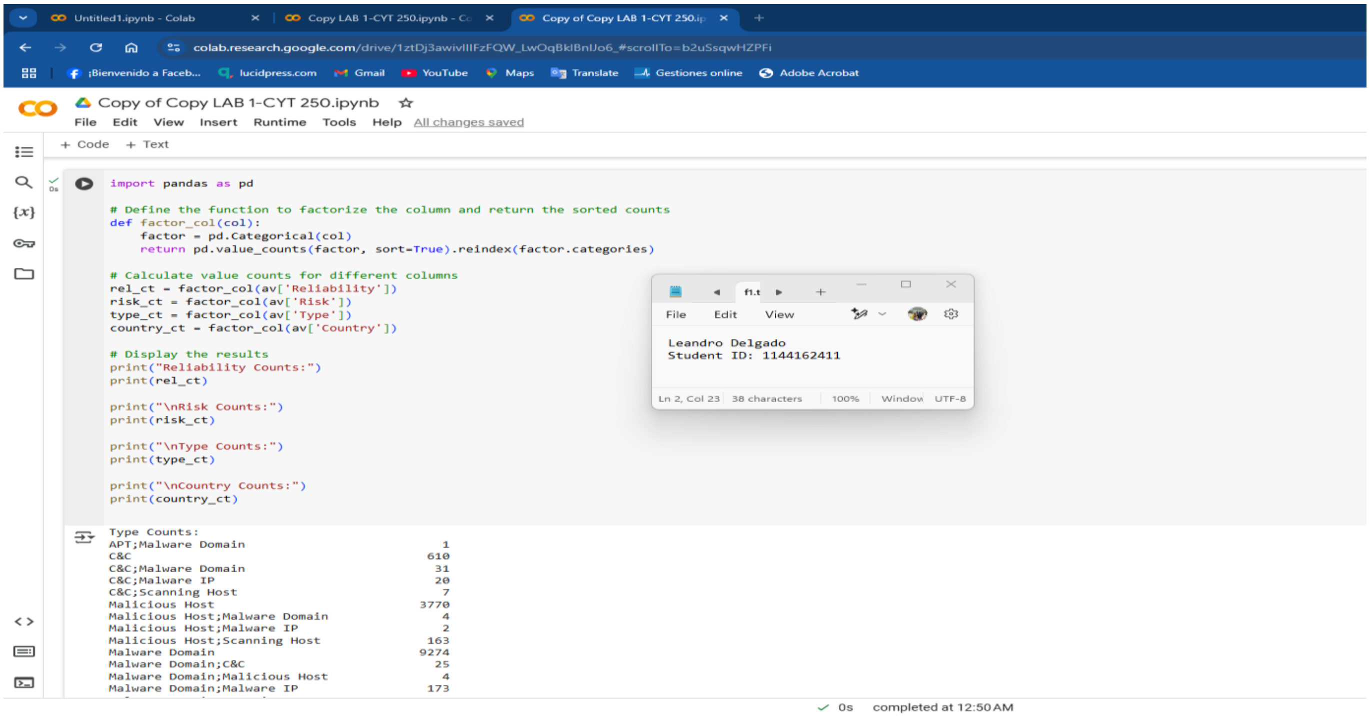
```
[69]: print(factor_col(av['Reliability']))
```

```
1       5612
2     149117
3      10892
4      87040
5          7
6       4758
7        297
8         21
9        686
10       196
dtype: int64
```

```
[70]: print(factor_col(av['Risk']))
```

```
1       39
2   213852
3    33719
4     9588
5     1328
6       90
7       10
dtype: int64
```

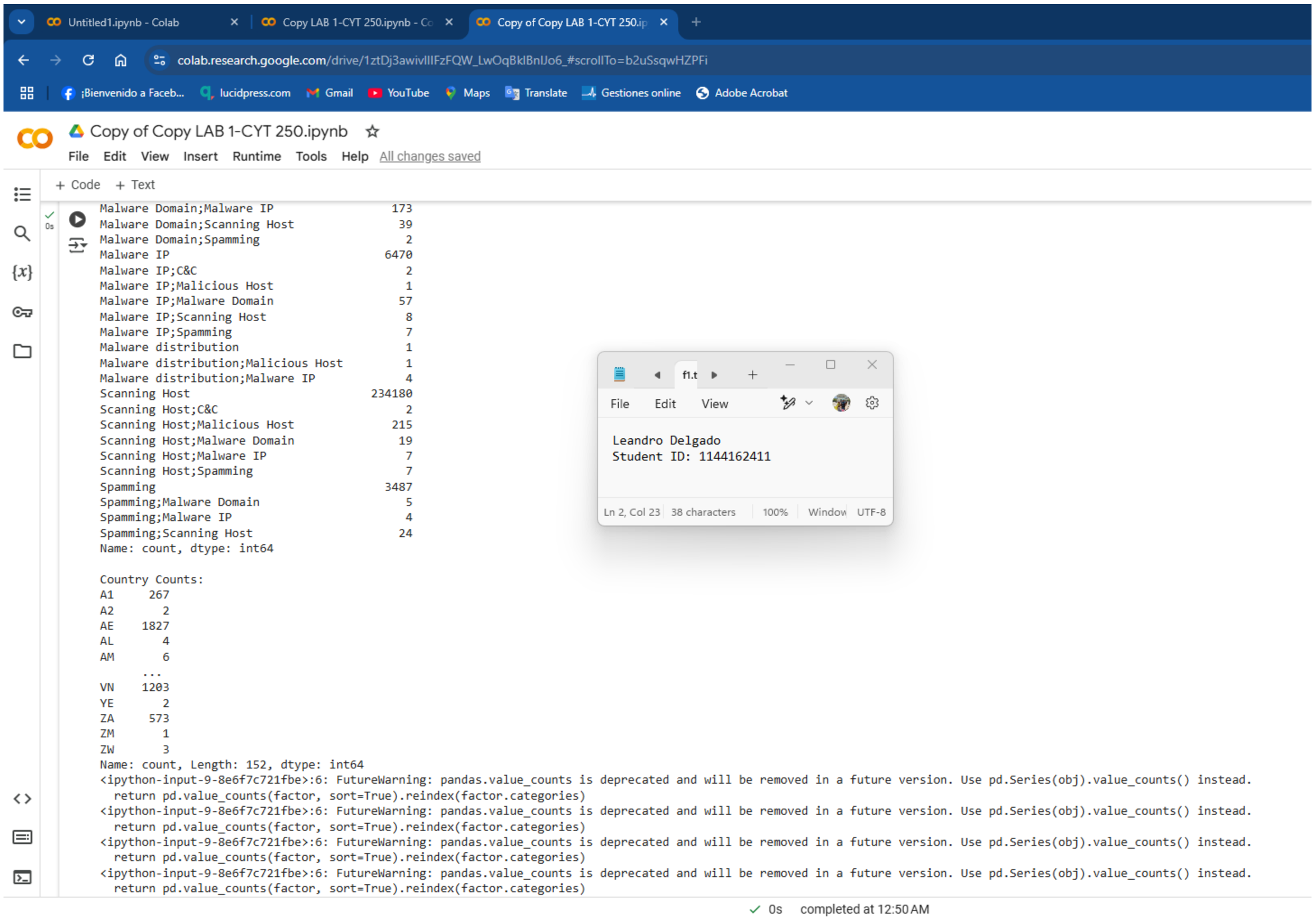
```
Leandro Delgado
Student ID: 1144162411
```



```
import pandas as pd

# Define the function to factorize the column and return the sorted counts
def factor_col(col):
    factor = pd.Categorical(col)
    return pd.value_counts(factor, sort=True).reindex(factor.categories)

# Calculate value counts for different columns
rel_ct = factor_col(av['Reliability'])
risk_ct = factor_col(av['Risk'])
type_ct = factor_col(av['Type'])
country_ct = factor_col(av['Country'])

# Display the results
print("Reliability Counts:")
print(rel_ct)

print("\nRisk Counts:")
print(risk_ct)

print("\nType Counts:")
print(type_ct)

print("\nCountry Counts:")
print(country_ct)
```

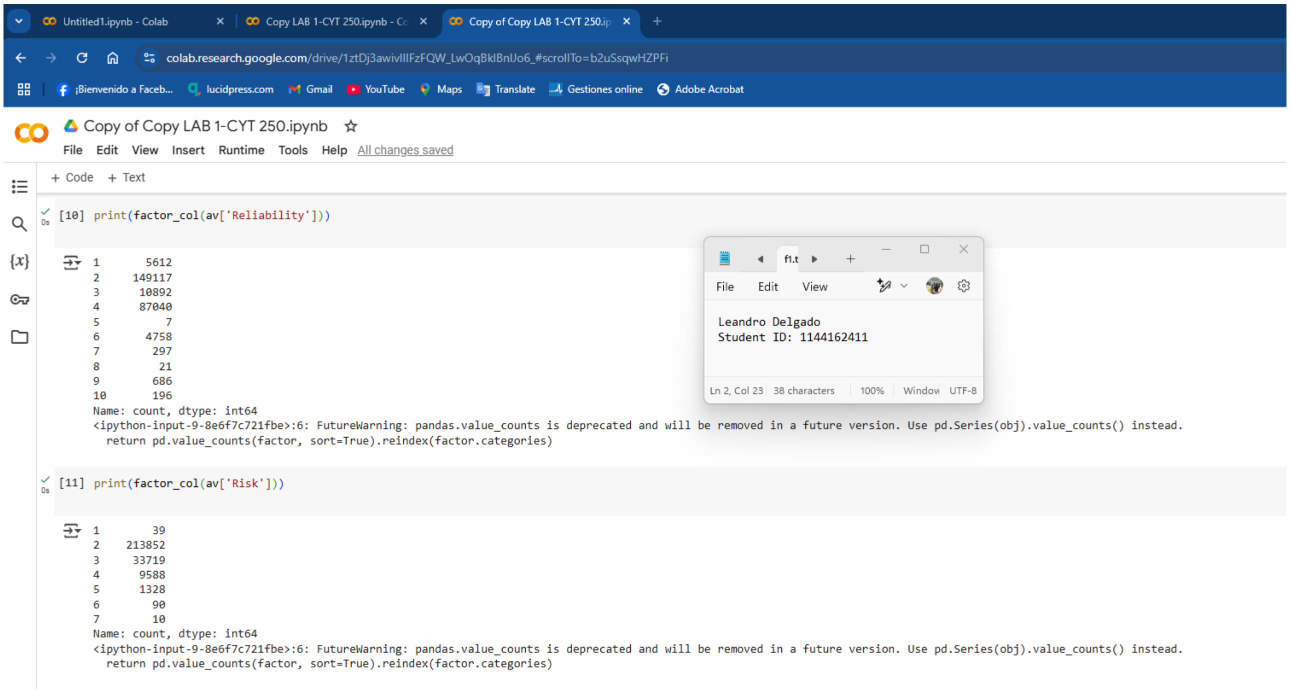
```
Leandro Delgado
Student ID: 1144162411
```

```
Type Counts:
APT;Malware Domain                      1
C&C                                   610
C&C;Malware Domain                     31
C&C;Malware IP                         20
C&C;Scanning Host                       7
Malicious Host                       3770
Malicious Host;Malware Domain           4
Malicious Host;Malware IP               2
Malicious Host;Scanning Host          163
Malware Domain                       9274
Malware Domain;C&C                     25
Malware Domain;Malicious Host           4
Malware Domain;Malware IP             173
```



```
Malware Domain;Malware IP             173
Malware Domain;Scanning Host           39
Malware Domain;Spamming                 2
Malware IP                           6470
Malware IP;C&C                          2
Malware IP;Malicious Host               1
Malware IP;Malware Domain              57
Malware IP;Scanning Host                8
Malware IP;Spamming                     7
Malware distribution                    1
Malware distribution;Malicious Host     1
Malware distribution;Malware IP         4
Scanning Host                      234180
Scanning Host;C&C                       2
Scanning Host;Malicious Host          215
Scanning Host;Malware Domain           19
Scanning Host;Malware IP                7
Scanning Host;Spamming                  7
Spamming                             3487
Spamming;Malware Domain                 5
Spamming;Malware IP                     4
Spamming;Scanning Host                 24
Name: count, dtype: int64

Country Counts:
A1     267
A2       2
AE    1827
AL       4
AM       6
      ...
VN    1203
YE       2
ZA     573
ZM       1
ZW       3
Name: count, Length: 152, dtype: int64
<ipython-input-9-8e6f7c721fbe>:6: FutureWarning: pandas.value_counts is deprecated and will be removed in a future version. Use pd.Series(obj).value_counts() instead.
  return pd.value_counts(factor, sort=True).reindex(factor.categories)
<ipython-input-9-8e6f7c721fbe>:6: FutureWarning: pandas.value_counts is deprecated and will be removed in a future version. Use pd.Series(obj).value_counts() instead.
  return pd.value_counts(factor, sort=True).reindex(factor.categories)
<ipython-input-9-8e6f7c721fbe>:6: FutureWarning: pandas.value_counts is deprecated and will be removed in a future version. Use pd.Series(obj).value_counts() instead.
  return pd.value_counts(factor, sort=True).reindex(factor.categories)
<ipython-input-9-8e6f7c721fbe>:6: FutureWarning: pandas.value_counts is deprecated and will be removed in a future version. Use pd.Series(obj).value_counts() instead.
  return pd.value_counts(factor, sort=True).reindex(factor.categories)
```

```
Leandro Delgado
Student ID: 1144162411
```



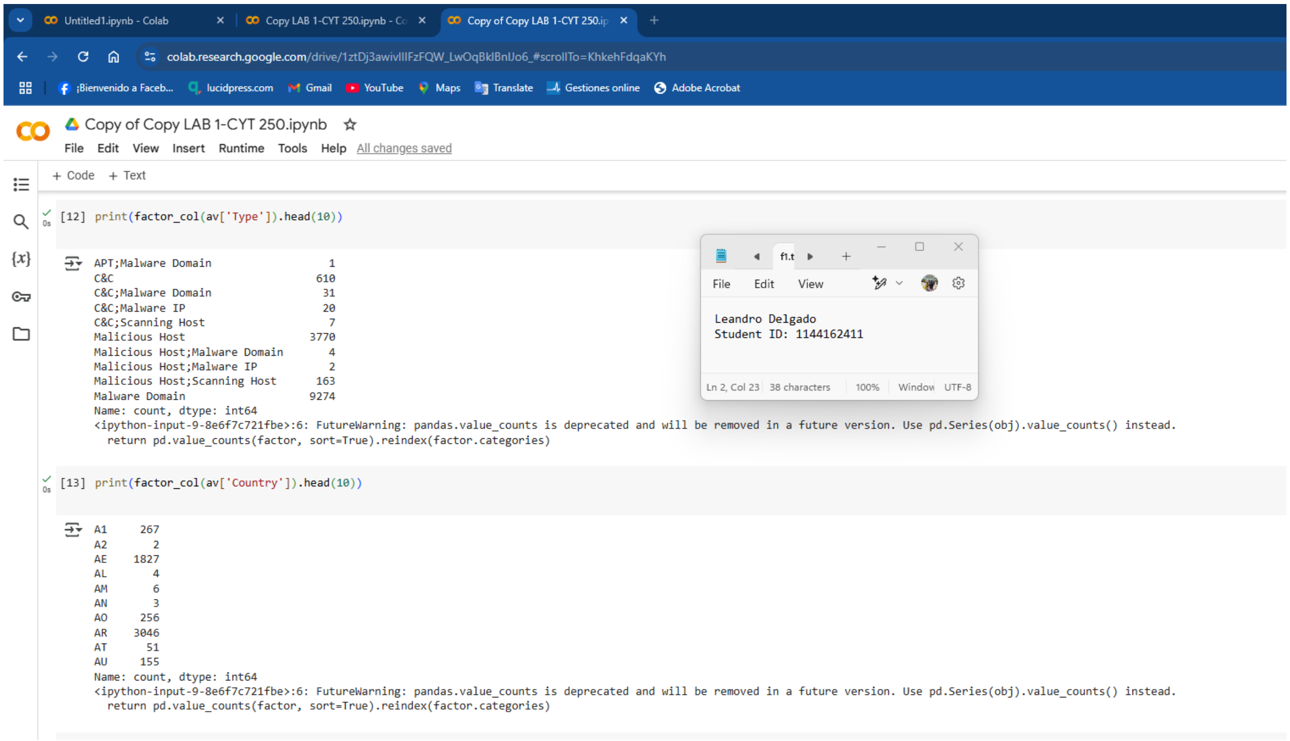
```
[10] print(factor_col(av['Reliability']))
```

```
1        5612
2      149117
3       10892
4       87040
5           7
6        4758
7         297
8          21
9         686
10        196
Name: count, dtype: int64
<ipython-input-9-8e6f7c721fbe>:6: FutureWarning: pandas.value_counts is deprecated and will be removed in a future version. Use pd.Series(obj).value_counts() instead.
  return pd.value_counts(factor, sort=True).reindex(factor.categories)
```

```
Leandro Delgado
Student ID: 1144162411
```

```
[11] print(factor_col(av['Risk']))
```

```
1        39
2    213852
3     33719
4      9588
5      1328
6        90
7        10
Name: count, dtype: int64
<ipython-input-9-8e6f7c721fbe>:6: FutureWarning: pandas.value_counts is deprecated and will be removed in a future version. Use pd.Series(obj).value_counts() instead.
  return pd.value_counts(factor, sort=True).reindex(factor.categories)
```



```
[12] print(factor_col(av['Type']).head(10))
```

```
APT;Malware Domain                    1
C&C                                 610
C&C;Malware Domain                   31
C&C;Malware IP                       20
C&C;Scanning Host                     7
Malicious Host                     3770
Malicious Host;Malware Domain         4
Malicious Host;Malware IP             2
Malicious Host;Scanning Host        163
Malware Domain                     9274
Name: count, dtype: int64
<ipython-input-9-8e6f7c721fbe>:6: FutureWarning: pandas.value_counts is deprecated and will be removed in a future version. Use pd.Series(obj).value_counts() instead.
  return pd.value_counts(factor, sort=True).reindex(factor.categories)
```

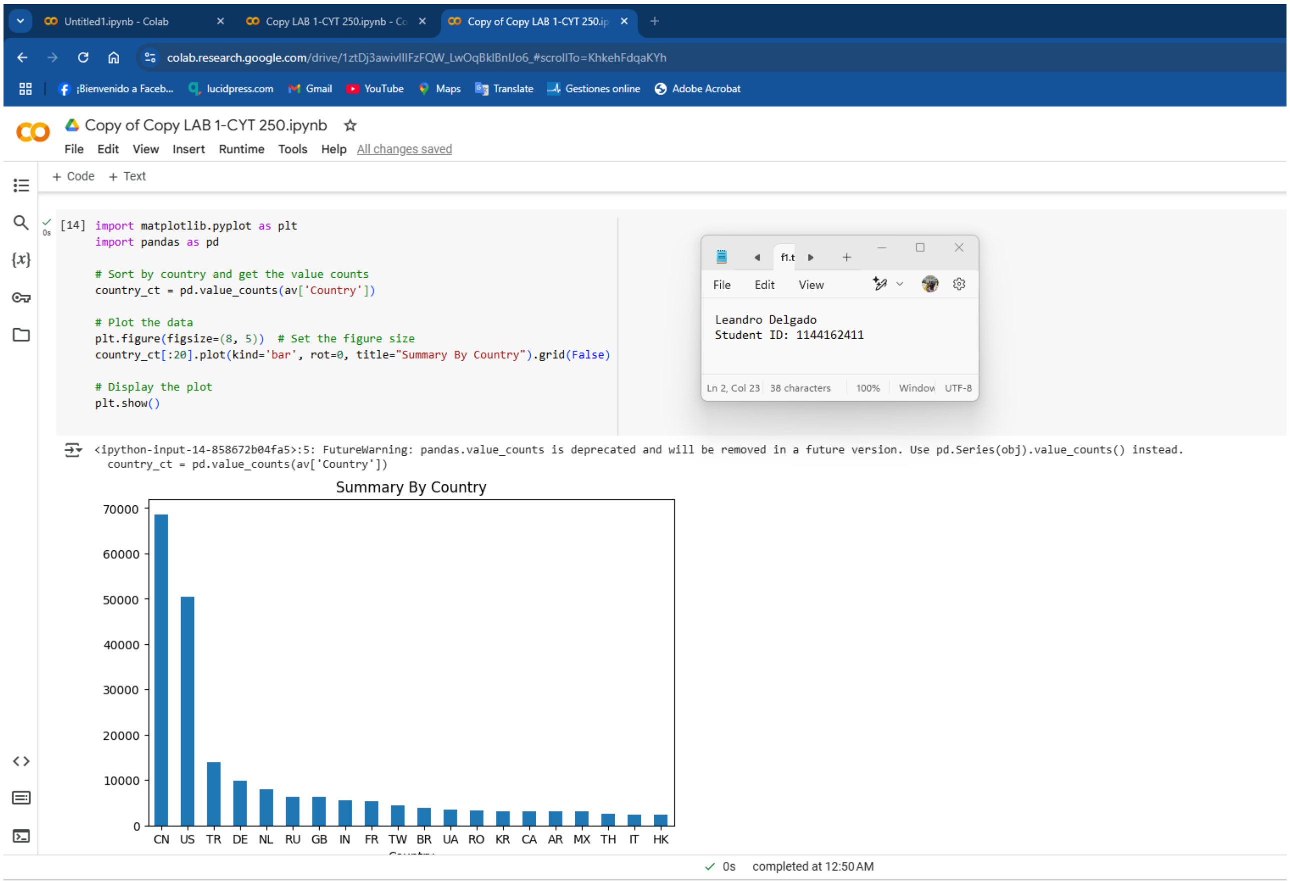
```
Leandro Delgado
Student ID: 1144162411
```

```
[13] print(factor_col(av['Country']).head(10))
```

```
A1     267
A2       2
AE    1827
AL       4
AM       6
AN       3
AO     256
AR    3046
AT      51
AU     155
Name: count, dtype: int64
<ipython-input-9-8e6f7c721fbe>:6: FutureWarning: pandas.value_counts is deprecated and will be removed in a future version. Use pd.Series(obj).value_counts() instead.
  return pd.value_counts(factor, sort=True).reindex(factor.categories)
```

Step 8. Run Listing 3-14. Number of records from the data frame will be shown as the graph, named Summary by Country.

Now, to visualize the number of records from the DataFrame as a graph, we ran the following script:



Untitled1.ipynb - Colab
Copy LAB 1-CYT 250.ipynb - Co
Copy of Copy LAB 1-CYT 250.ip
colab.research.google.com/drive/1ztDj3awivlIlFzFQW_LwOqBklBnlJo6_#scrollTo=KhkehFdqaKYh
¡Bienvenido a Faceb...
lucidpress.com
Gmail
YouTube
Maps
Translate
Gestiones online
Adobe Acrobat
Copy of Copy LAB 1-CYT 250.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
f1.t
File Edit View
Ln 2, Col 23 38 characters 100% Window UTF-8
Summary By Country
70000
60000
50000
40000
30000
20000
10000
0
CN US TR DE NL RU GB IN FR TW BR UA RO KR CA AR MX TH IT HK
Country
✓ 0s completed at 12:50 AM

```
[14] import matplotlib.pyplot as plt
     import pandas as pd

     # Sort by country and get the value counts
     country_ct = pd.value_counts(av['Country'])

     # Plot the data
     plt.figure(figsize=(8, 5))  # Set the figure size
     country_ct[:20].plot(kind='bar', rot=0, title="Summary By Country").grid(False)

     # Display the plot
     plt.show()
```

```
Leandro Delgado
Student ID: 1144162411
```

```
<ipython-input-14-858672b04fa5>:5: FutureWarning: pandas.value_counts is deprecated and will be removed in a future version. Use pd.Series(obj).value_counts() instead.
  country_ct = pd.value_counts(av['Country'])
```

Questions:

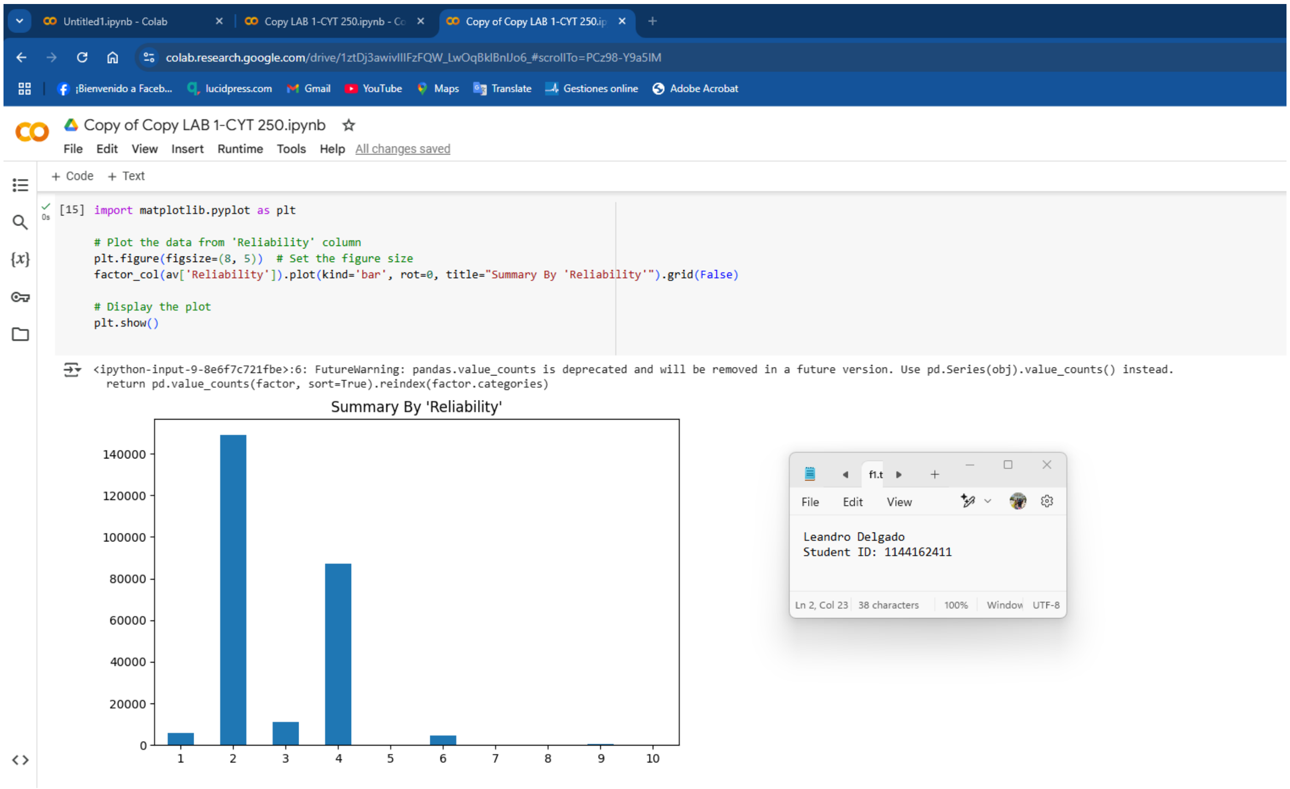
- If a country does not have valid country code, will the records be taken for calculation? Answer: No

Why?

If a country code is missing (blank or empty), it still contributes to the frequency count in pd.value_counts(av['Country']). However, when plotting with country CT[:20].plot(kind='bar'), only the top 20 countries with the highest counts are displayed. Empty country codes won't appear in the graph due to this slicing.

Step 9. Listing 3-15. The result shows Reliability chart for top 10 countries (see Figure 3-6).

To show the Reliability chart for the top 10 countries, we ran 'listing 3-15' from the provided Python script.



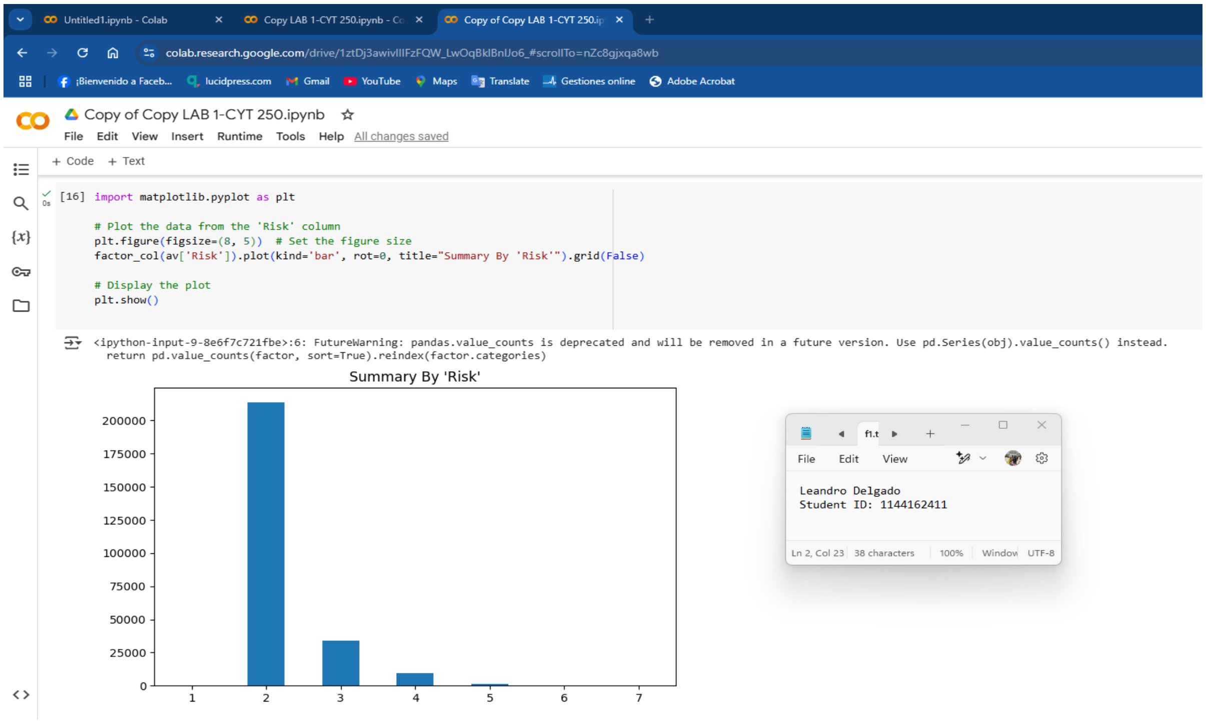
```
import matplotlib.pyplot as plt

# Plot the data from 'Reliability' column
plt.figure(figsize=(8, 5))  # Set the figure size
factor_col(av['Reliability']).plot(kind='bar', rot=0, title="Summary By 'Reliability'").grid(False)

# Display the plot
plt.show()
```

```
<ipython-input-9-8e6f7c721fbe>:6: FutureWarning: pandas.value_counts is deprecated and will be removed in a future version. Use pd.Series(obj).value_counts() instead.
  return pd.value_counts(factor, sort=True).reindex(factor.categories)
```

Leandro Delgado
Student ID: 1144162411

Step 10. Listing 3-16. The result shows Risk chart for top 10 countries (see Figure 3-7).

To display the Risk chart for the top 10 countries, we ran 'listing 3-16' from the provided Python script



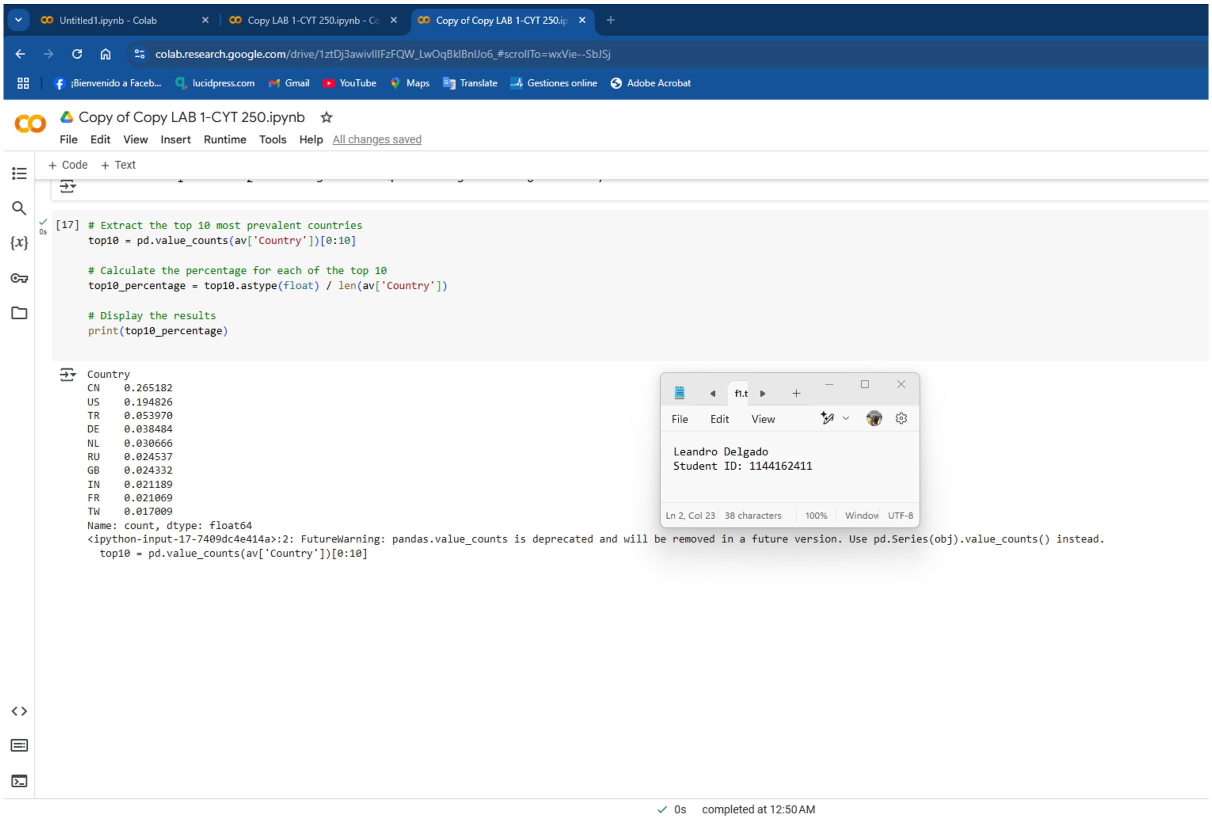
```
import matplotlib.pyplot as plt

# Plot the data from the 'Risk' column
plt.figure(figsize=(8, 5))  # Set the figure size
factor_col(av['Risk']).plot(kind='bar', rot=0, title="Summary By 'Risk'").grid(False)

# Display the plot
plt.show()
```

```
<ipython-input-9-8e6f7c721fbe>:6: FutureWarning: pandas.value_counts is deprecated and will be removed in a future version. Use pd.Series(obj).value_counts() instead.
  return pd.value_counts(factor, sort=True).reindex(factor.categories)
```

Leandro Delgado
Student ID: 1144162411

Step 11. Run Listing 3-18. The result will show data by country in percentage.

Finally, we displayed the data by country in percentage, using the script to visualize it.



```
[17] # Extract the top 10 most prevalent countries
     top10 = pd.value_counts(av['Country'])[0:10]

     # Calculate the percentage for each of the top 10
     top10_percentage = top10.astype(float) / len(av['Country'])

     # Display the results
     print(top10_percentage)
```

```
Country
CN    0.265182
US    0.194826
TR    0.053970
DE    0.038484
NL    0.030666
RU    0.024537
GB    0.024332
IN    0.021189
FR    0.021069
TW    0.017009
Name: count, dtype: float64
<ipython-input-17-7409dc4e414a>:2: FutureWarning: pandas.value_counts is deprecated and will be removed in a future version. Use pd.Series(obj).value_counts() instead.
  top10 = pd.value_counts(av['Country'])[0:10]
```

```
Leandro Delgado
Student ID: 1144162411
```

Question:

- What line of Python code does this calculation (copy and paste)?

The Python code that performs this calculation is as follows: top10.astype(float) / len(av['Country']).

Lab Summary:

In this lab, we set up our environment with Python and Pandas to ensure Jupyter Notebook was ready to go. First, we unzipped the "Scripts for Assignment1.rar" file and moved it to our Anaconda environment. We then ran and debugged the Python scripts to access and display the first few rows of the IP Reputation Database. After that, we took a closer look at the database, figured out the column names, and generated some HTML outputs. We used Pandas to perform descriptive statistics on both numerical and categorical data. To visualize the data, we created graphs showing the number of records by country and reliability charts for the top ten countries. Finally, we calculated the percentage of malicious nodes by country, which helped us better understand data-driven security practices and threat intelligence.