Data Science for Cybersecurity

InfoSecWorld 2023

Hands-on Lab

Thomas Scanlon

Clarence Worrell

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

[Distribution Statement A] Approved for public release and unlimited distribution

Notice

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by Carnegie Mellon University or its Software Engineering Institute.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

DM23-1010

Agenda

- **Data science tools**
- **BETH cybersecurity dataset**
- **Exploratory analysis**
- **Hands-on exercise**

Data Science Tools

No-code / low-code

• Excel	(spreadsheet)
• Orange	(drag-and-drop machine learning)

Coding-based tools

• Python	(general programming language, many data science libraries)
• R	(statistics)
• MATLAB	

Many **other options**

Environment for the Lab

- **Google Colaboratory (Colab)**
  - Online alternative to local Anaconda installation
  - Free data analysis and machine learning tool
  - Write and execute python code in a browser
  - Mix rich text, coding, and code output into a well-formatted PDF report
  - No installations required
  - **Requires a Gmail account**
  - https://colab.research.google.com

BETH Cybersecurity Dataset

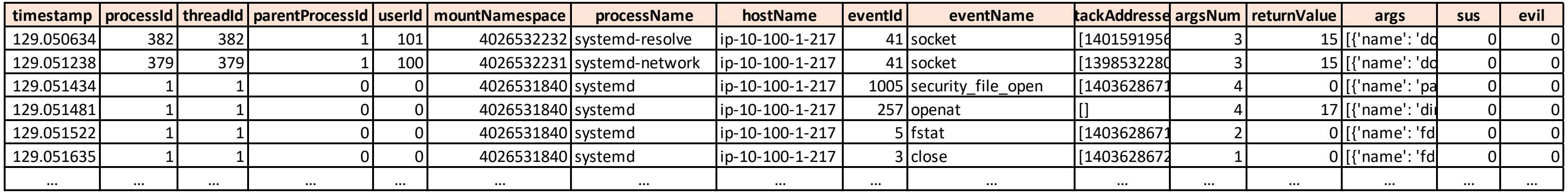
BETH* is real cybersecurity dataset published in 2021 as a benchmark for anomaly detection researchers

- 8 million records, generated by 23 hosts, during 5 discontiguous hours
- Each host includes benign traffic and s at most one single attack
- Each record is labeled as to whether it is “benign” or “malicious”

*Highnam, K., Arulkumaran, K., Hanif, Z., & Jennings, N. R. (2021). "BETH dataset: Real Cybersecurity Data for Anomaly Detection Research." ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning. http://www.gatsby.ucl.ac.uk/~balaji/udl2021/accepted-papers/UDL2021-paper-033.pdf

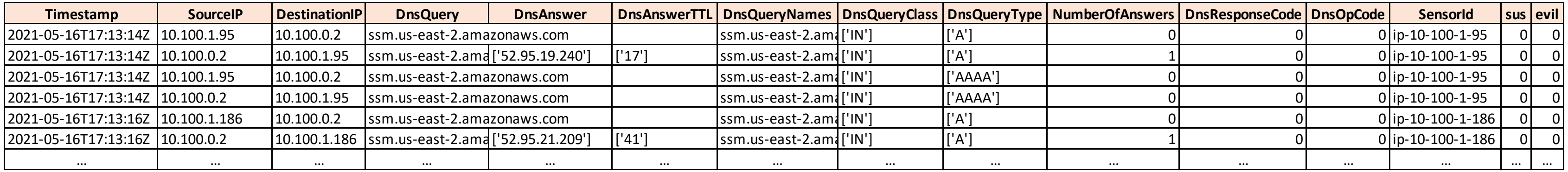
BETH Cybersecurity Dataset (cont.)

System logfiles



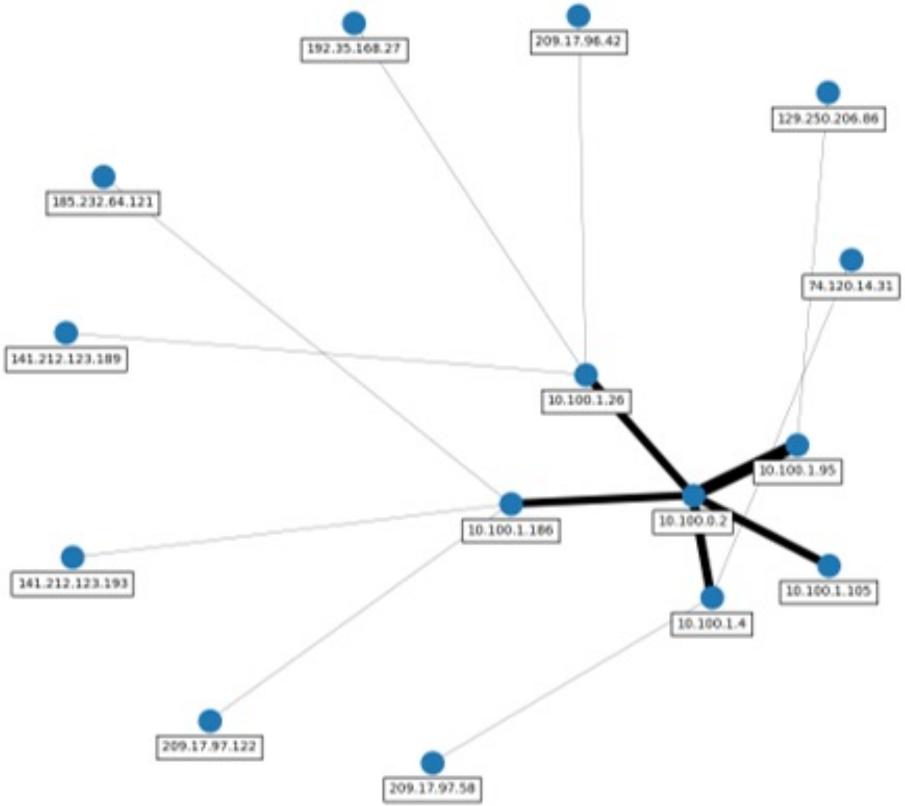
timestamp	processId	threadId	parentProcessId	userId	mountNamespace	processName	hostName	eventId	eventName	tackAddresse	argsNum	returnValue	args	sus	evil
129.050634	382	382	1	101	4026532232	systemd-resolve	ip-10-100-1-217	41	socket	[1401591956	3	15	[{'name': 'do	0	0
129.051238	379	379	1	100	4026532231	systemd-network	ip-10-100-1-217	41	socket	[1398532280	3	15	[{'name': 'do	0	0
129.051434	1	1	0	0	4026531840	systemd	ip-10-100-1-217	1005	security_file_open	[1403628671	4	0	[{'name': 'pa	0	0
129.051481	1	1	0	0	4026531840	systemd	ip-10-100-1-217	257	openat	[]	4	17	[{'name': 'di	0	0
129.051522	1	1	0	0	4026531840	systemd	ip-10-100-1-217	5	fstat	[1403628671	2	0	[{'name': 'fd	0	0
129.051635	1	1	0	0	4026531840	systemd	ip-10-100-1-217	3	close	[1403628672	1	0	[{'name': 'fd	0	0
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…

DNS query logfiles

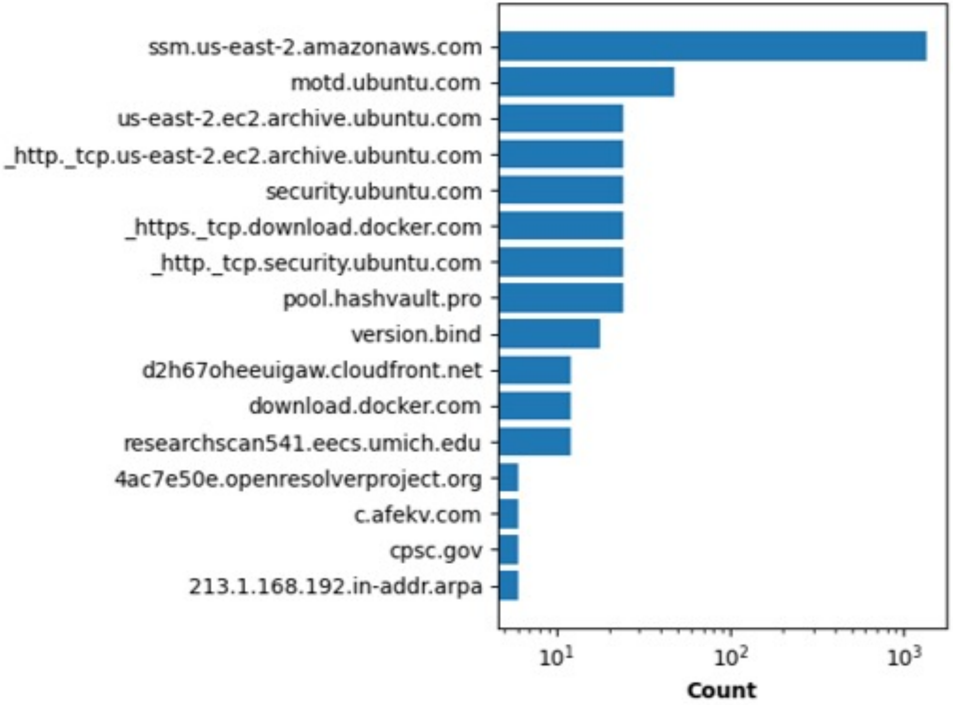


Timestamp	SourceIP	DestinationIP	DnsQuery	DnsAnswer	DnsAnswerTTL	DnsQueryNames	DnsQueryClass	DnsQueryType	NumberOfAnswers	DnsResponseCode	DnsOpCode	SensorId	sus	evil
2021-05-16T17:13:14Z	10.100.1.95	10.100.0.2	ssm.us-east-2.amazonaws.com			ssm.us-east-2.am	['IN']	['A']	0	0	0	ip-10-100-1-95	0	0
2021-05-16T17:13:14Z	10.100.0.2	10.100.1.95	ssm.us-east-2.ama	['52.95.19.240']	['17']	ssm.us-east-2.am	['IN']	['A']	1	0	0	ip-10-100-1-95	0	0
2021-05-16T17:13:14Z	10.100.1.95	10.100.0.2	ssm.us-east-2.amazonaws.com			ssm.us-east-2.am	['IN']	['AAAA']	0	0	0	ip-10-100-1-95	0	0
2021-05-16T17:13:14Z	10.100.0.2	10.100.1.95	ssm.us-east-2.amazonaws.com			ssm.us-east-2.am	['IN']	['AAAA']	0	0	0	ip-10-100-1-95	0	0
2021-05-16T17:13:16Z	10.100.1.186	10.100.0.2	ssm.us-east-2.amazonaws.com			ssm.us-east-2.am	['IN']	['A']	0	0	0	ip-10-100-1-186	0	0
2021-05-16T17:13:16Z	10.100.0.2	10.100.1.186	ssm.us-east-2.ama	['52.95.21.209']	['41']	ssm.us-east-2.am	['IN']	['A']	1	0	0	ip-10-100-1-186	0	0
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…

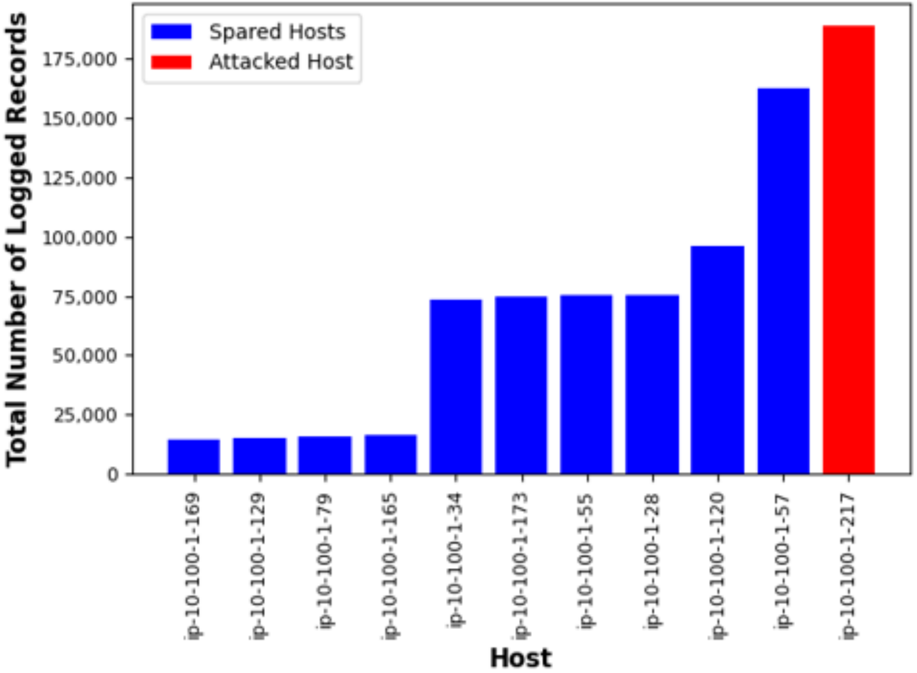
DNS Query Traffic between IP Addresses (BETH Dataset)



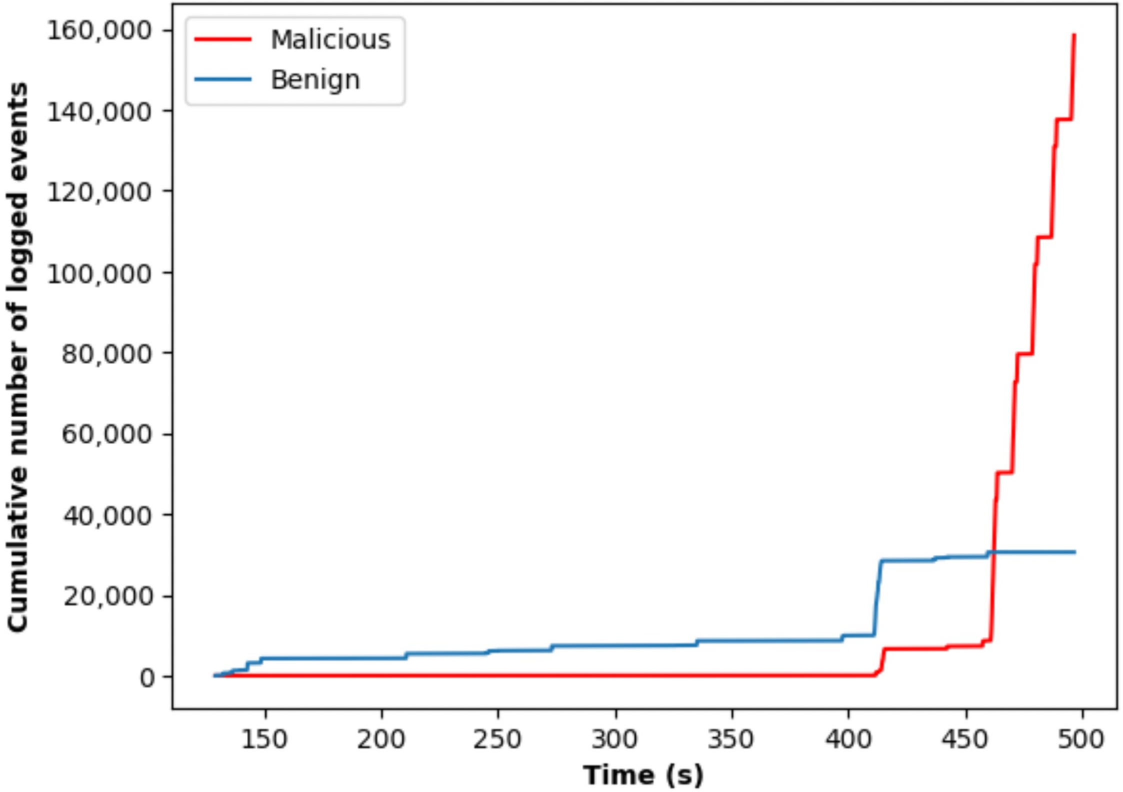
DNS Query Volume by Domain Name (BETH Dataset)



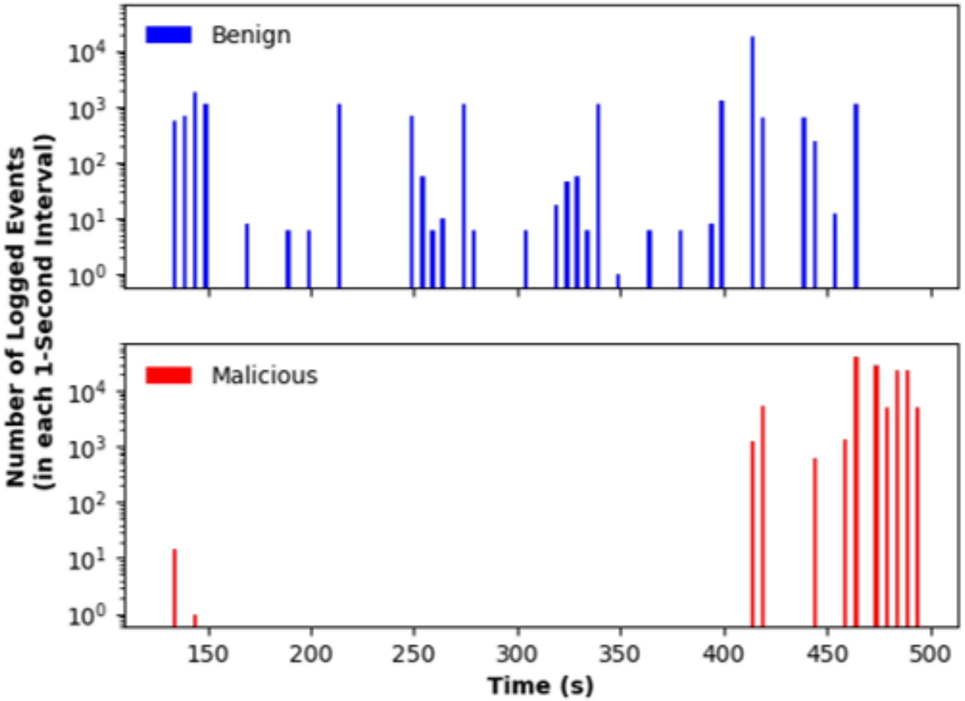
Logged Events by Host (BETH Dataset)



Logged Events in Time on the Attacked Host (BETH Dataset)



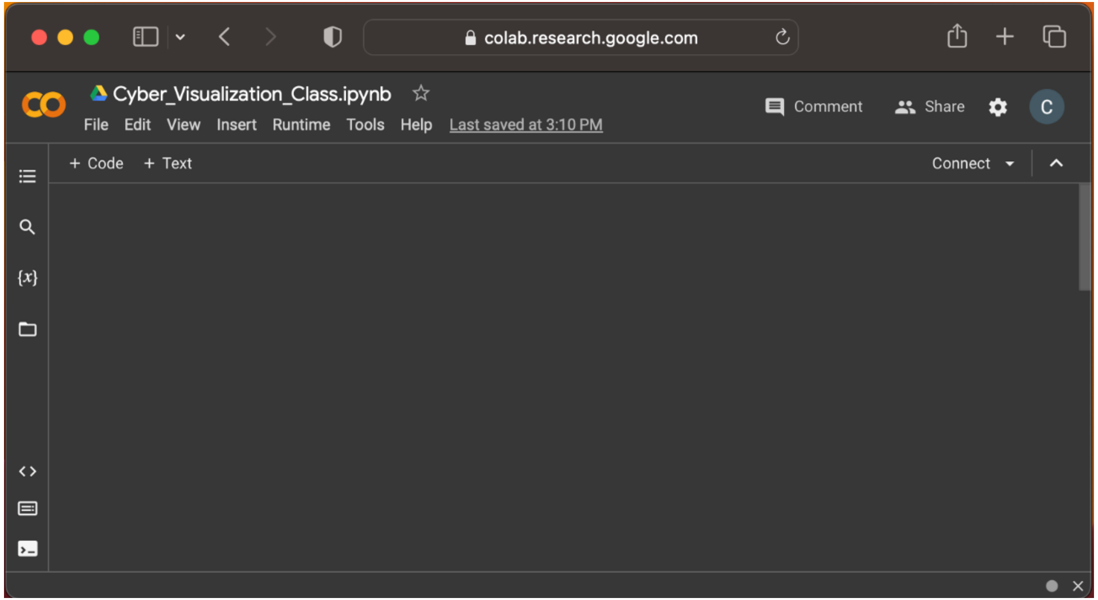
Logged Events in Time on the Attacked Host (BETH Dataset)



Let’s set up Google Colab

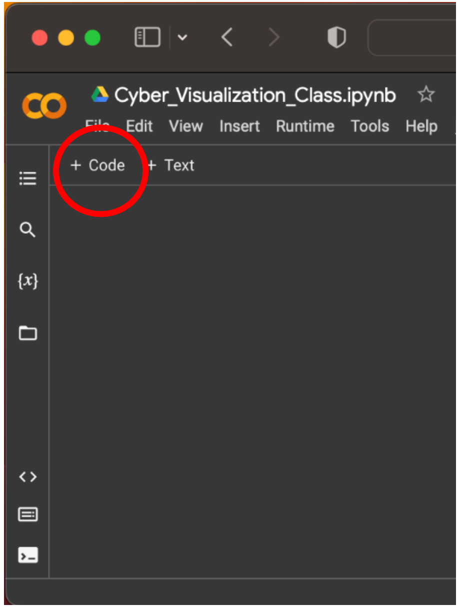
Colab: Getting Started – 1

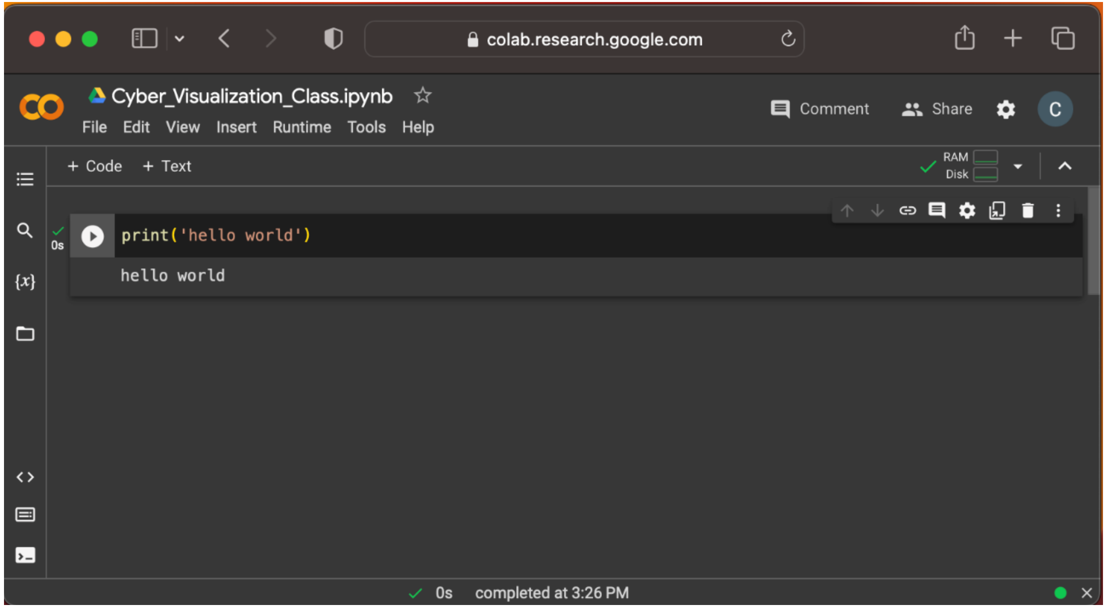
1. Sign into your Gmail account in your browser
2. Go to https://colab.research.google.com
3. Click **File** > **New Notebook**


colab.research.google.com
CO
Cyber_Visualization_Class.ipynb
File Edit View Insert Runtime Tools Help
Last saved at 3:10 PM
Comment
Share
C
+ Code + Text
Connect

Colab: Getting Started – 2

1. Click **+ Code** to create a code cell
2. Type `print('hello world')`
3. Press **CNTL Enter**, or press the **Play** button to execute the cell


Cyber_Visualization_Class.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text


colab.research.google.com
Cyber_Visualization_Class.ipynb
File Edit View Insert Runtime Tools Help
Comment Share
+ Code + Text
RAM
Disk
0s

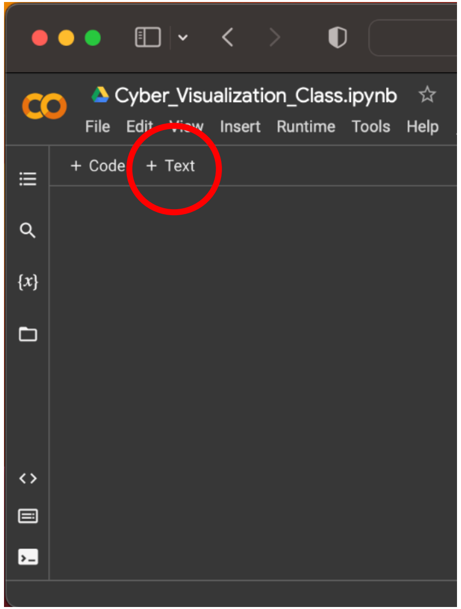
```
print('hello world')
```

```
hello world
```

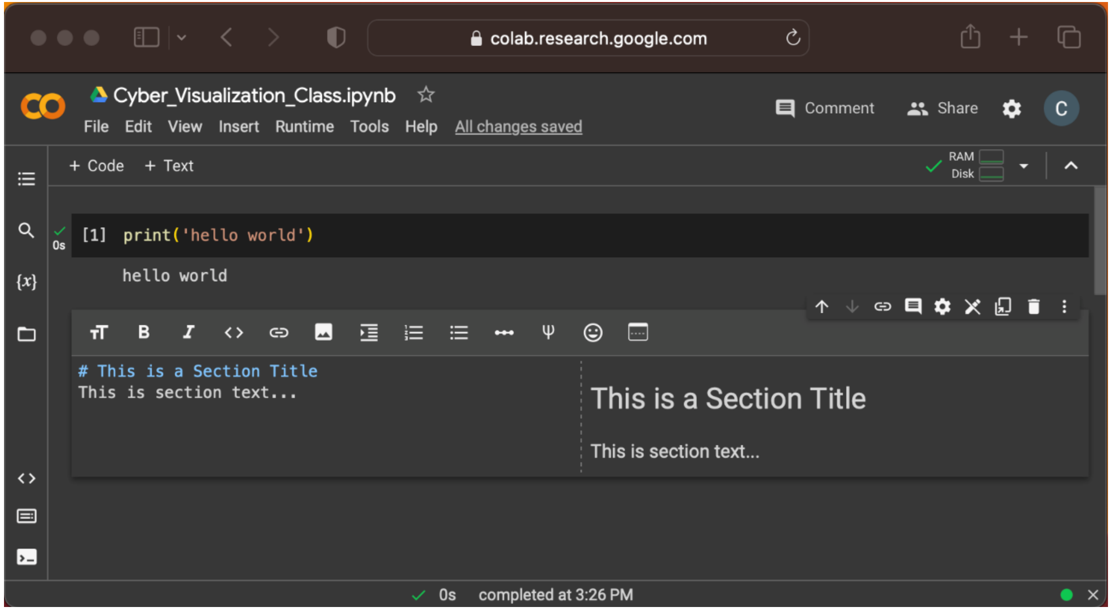
✓ 0s completed at 3:26 PM

Colab: Getting Started – 3

1. Click **+ Text** to create a text cell
2. Add formatted text throughout your notebook to explain the analysis
3. Use **#** (hashtag symbol) for section titles



Cyber_Visualization_Class.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text



colab.research.google.com
Cyber_Visualization_Class.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share C
+ Code + Text
RAM
Disk
0s

```
[1] print('hello world')

    hello world
```

```
# This is a Section Title
This is section text...
```

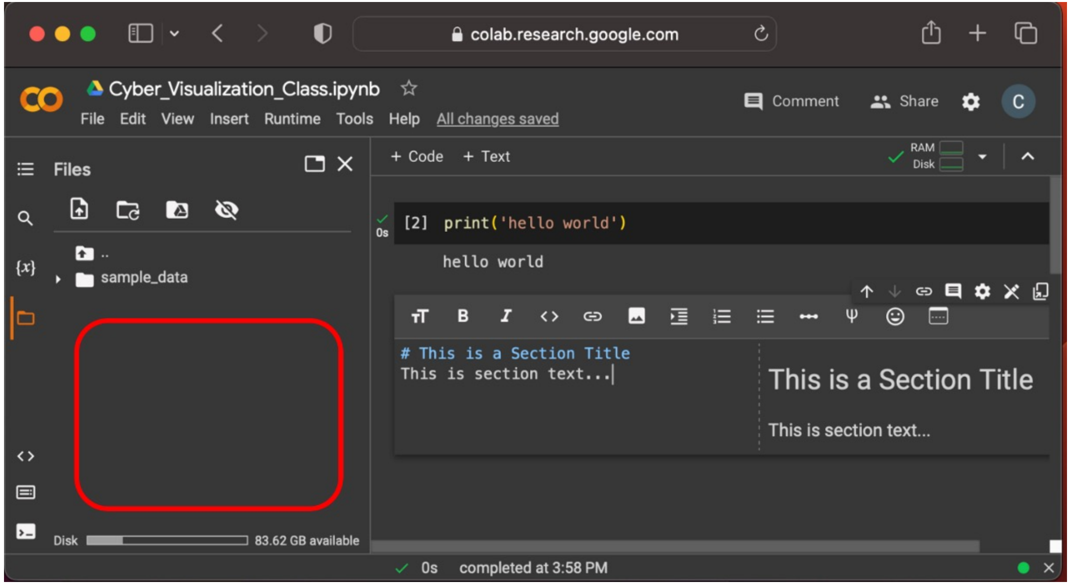
This is a Section Title

This is section text...

Import Data

1. Import **data_for_class_exercise.csv** into Colab

2. Drag-and-drop the file into the area marked **Files** (outlined in red in the image). It will take about a minute to upload.


colab.research.google.com
Cyber_Visualization_Class.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
C
+ Code + Text
RAM
Disk
Files
..
sample_data
Disk 83.62 GB available
0s

```
[2] print('hello world')
hello world
```

```
# This is a Section Title
This is section text...
```

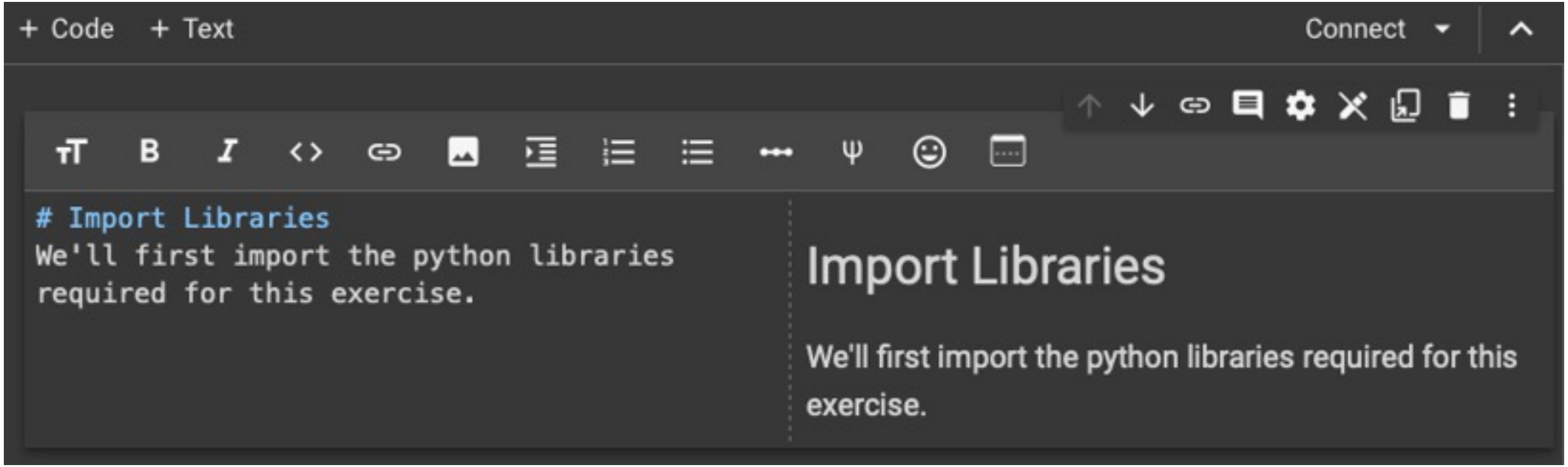
This is a Section Title

This is section text...

Import Libraries – 1

1. Click **+ Text** to add a new text cell
2. Add formatted text to indicate that we'll import libraries, for example

   Import Libraries
   We'll first import the python libraries required for this exercise.



```
# Import Libraries
We'll first import the python libraries
required for this exercise.
```

Import Libraries

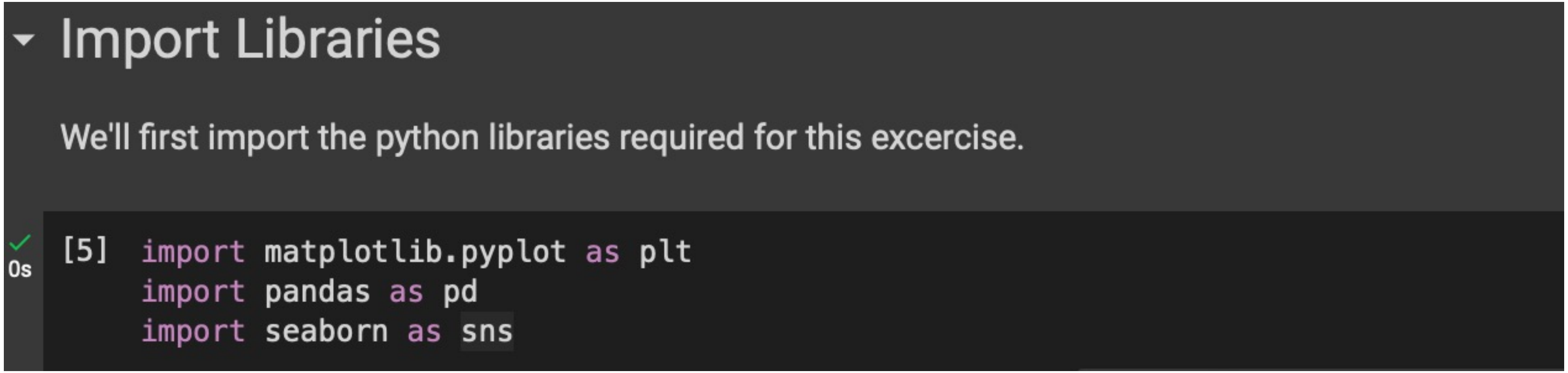
We'll first import the python libraries required for this exercise.

Import Libraries – 2

1. Click **+ Code** to add a new Code cell
2. Add the following code

```
[5]  import matplotlib.pyplot as plt
     import pandas as pd
     import seaborn as sns
```

3. Press **CNTL-Enter** to execute the cell



Import Libraries

We'll first import the python libraries required for this excercise.

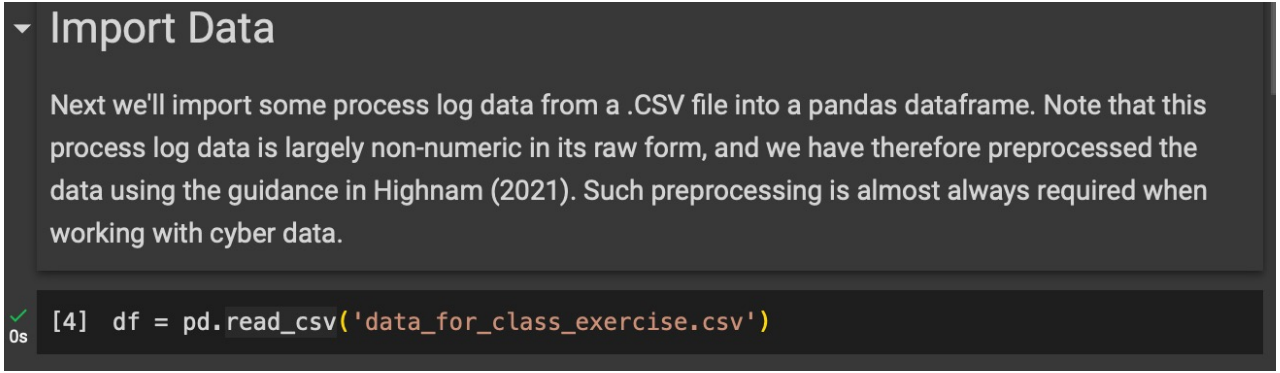
```
[5]  import matplotlib.pyplot as plt
     import pandas as pd
     import seaborn as sns
```

Import the .csv File of Data

1. Click **+ Text** to add a new text cell
2. Add formatted text to indicate that we'll import data, for example

   Next, we'll import some process log data from a .csv file into a pandas dataframe. Note that this process log data is largely non-numeric in its raw form and we have, therefore, preprocessed the data using the guidance in Highnam (2021). Such preprocessing is almost always required when working with cyber data.
3. Click **+ Code** to add a new code cell
4. Add the following code

   [4] df = pd.read_csv('data_for_class_exercise.csv')
5. Press **CNTL-Enter** to execute the cell



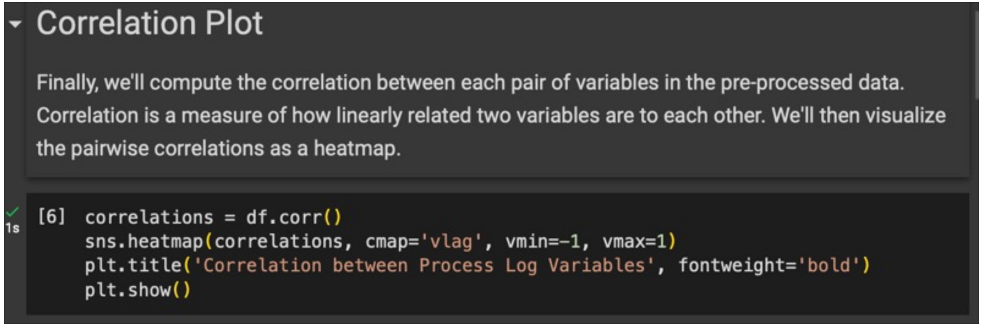
Import Data

Next we'll import some process log data from a .CSV file into a pandas dataframe. Note that this process log data is largely non-numeric in its raw form, and we have therefore preprocessed the data using the guidance in Highnam (2021). Such preprocessing is almost always required when working with cyber data.

```
[4]  df = pd.read_csv('data_for_class_exercise.csv')
```

Plot the Correlations Between Each Variable

1. Click **+ Text** to add a new text cell
2. Add formatted text to indicate that we’ll compute the correlation, for example
   Finally, we’ll compute the correlation between each pair of variables in the pre-processed data. Correlation is a measure of how linearly related two variables area to each other. We’ll then visualize the pairwise correlations as a heatmap.
3. Click **+ Code** to add a new code cell
4. Add the following code
   [6] correlations = df.corr()
   Sns.heatmap(correlations, cmap=‘vlag’, vmin=-1, vmax=1)
   Plt.title(‘Correlation between Process Log Variables’, fontweight=‘bold’)
   Plt.show()
5. Press **CNTL-Enter** to execute the cell



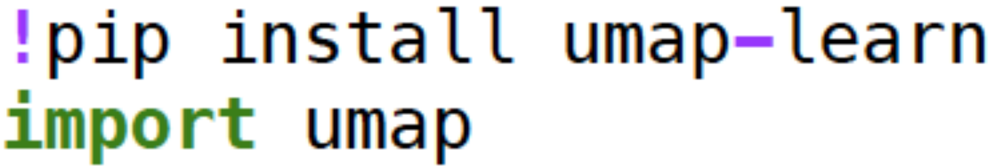
Correlation Plot

Finally, we'll compute the correlation between each pair of variables in the pre-processed data. Correlation is a measure of how linearly related two variables are to each other. We'll then visualize the pairwise correlations as a heatmap.

```
[6] correlations = df.corr()
    sns.heatmap(correlations, cmap='vlag', vmin=-1, vmax=1)
    plt.title('Correlation between Process Log Variables', fontweight='bold')
    plt.show()
```

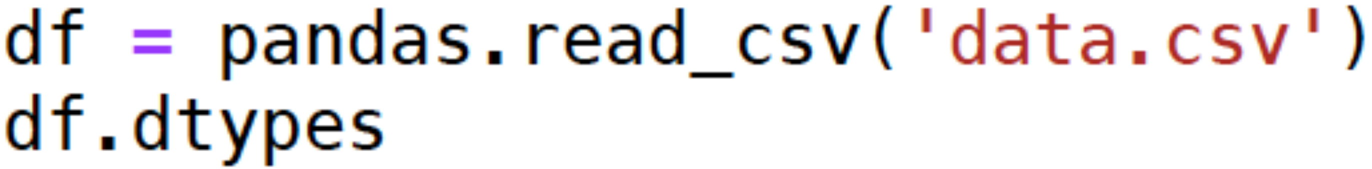
Hands-On Exercise

Install and import libraries



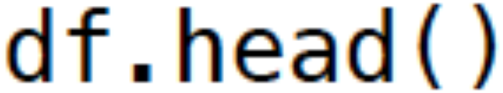
```
!pip install umap-learn
import umap
```

Read .csv into dataframe



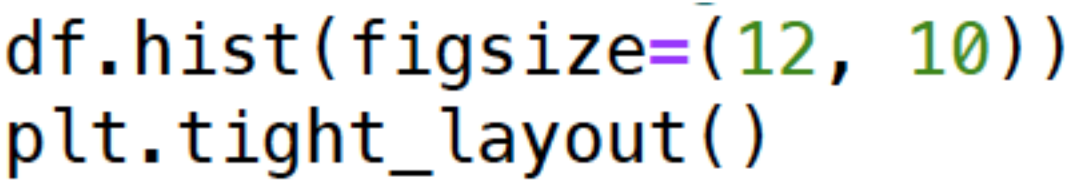
```
df = pandas.read_csv('data.csv')
df.dtypes
```

View first 5 records



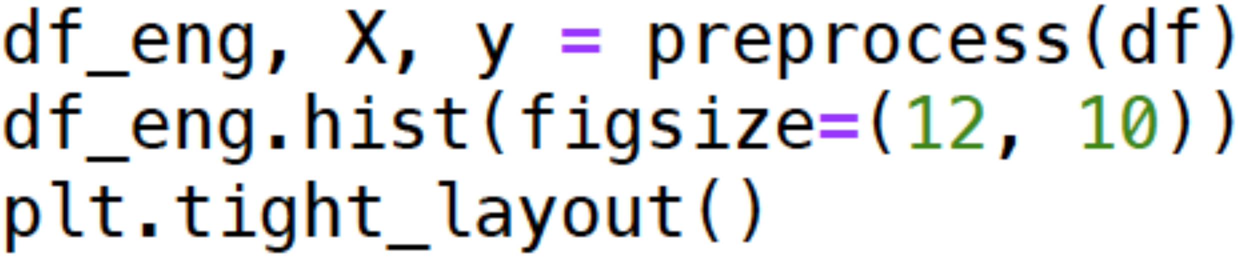
```
df.head()
```

Histograms of the raw data



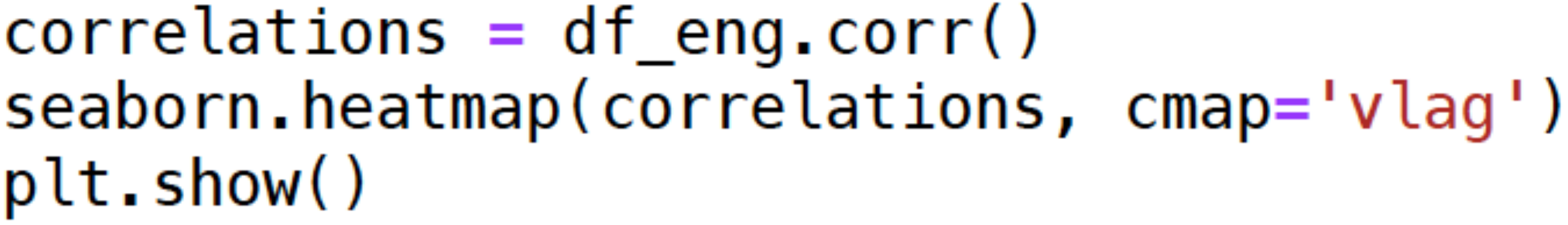
```
df.hist(figsize=(12, 10))
plt.tight_layout()
```

Histograms of the engineered features



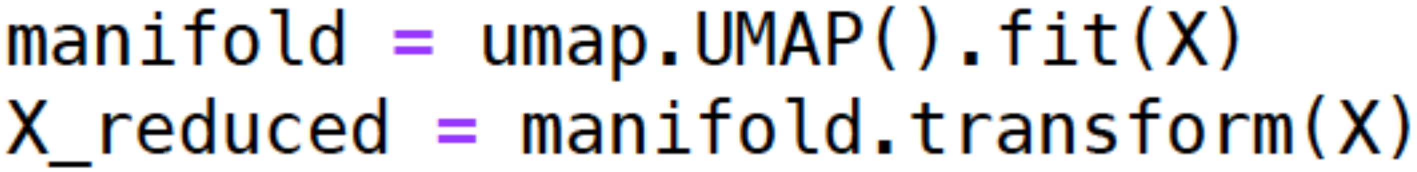
```
df_eng, X, y = preprocess(df)
df_eng.hist(figsize=(12, 10))
plt.tight_layout()
```

Correlations plot



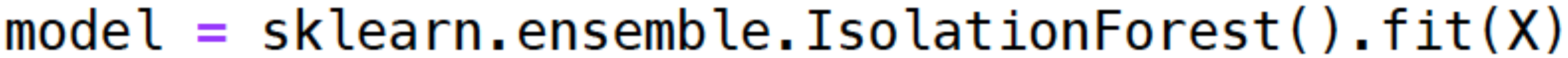
```
correlations = df_eng.corr()
seaborn.heatmap(correlations, cmap='vlag')
plt.show()
```

UMAP dimensionality reduction



```
manifold = umap.UMAP().fit(X)
X_reduced = manifold.transform(X)
```

Fit anomaly detecting isolation forest model



```
model = sklearn.ensemble.IsolationForest().fit(X)
```

End of Workshop