

Phase	Activity	Code and Output																																																																														
Extract	The dataset was extracted from the “RT_IOT2022.csv”.	<pre>import pandas as pd  data = pd.read_csv("RT_IOT2022.csv") data = pd.DataFrame(data) data.head()</pre> <table><thead><tr><th></th><th>no</th><th>id.orig_p</th><th>id.resp_p</th><th>proto</th><th>service</th><th>flow_duration</th><th>fwd_pkts_tot</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>38667</td><td>1883</td><td>tcp</td><td>mqtt</td><td>32.011598</td><td></td></tr><tr><td>1</td><td>1</td><td>51143</td><td>1883</td><td>tcp</td><td>mqtt</td><td>31.883584</td><td></td></tr><tr><td>2</td><td>2</td><td>44761</td><td>1883</td><td>tcp</td><td>mqtt</td><td>32.124053</td><td></td></tr><tr><td>3</td><td>3</td><td>60893</td><td>1883</td><td>tcp</td><td>mqtt</td><td>31.961063</td><td></td></tr><tr><td>4</td><td>4</td><td>51087</td><td>1883</td><td>tcp</td><td>mqtt</td><td>31.902362</td><td></td></tr></tbody></table> <p>5 rows × 85 columns</p>		no	id.orig_p	id.resp_p	proto	service	flow_duration	fwd_pkts_tot	0	0	38667	1883	tcp	mqtt	32.011598		1	1	51143	1883	tcp	mqtt	31.883584		2	2	44761	1883	tcp	mqtt	32.124053		3	3	60893	1883	tcp	mqtt	31.961063		4	4	51087	1883	tcp	mqtt	31.902362																															
	no	id.orig_p	id.resp_p	proto	service	flow_duration	fwd_pkts_tot																																																																									
0	0	38667	1883	tcp	mqtt	32.011598																																																																										
1	1	51143	1883	tcp	mqtt	31.883584																																																																										
2	2	44761	1883	tcp	mqtt	32.124053																																																																										
3	3	60893	1883	tcp	mqtt	31.961063																																																																										
4	4	51087	1883	tcp	mqtt	31.902362																																																																										
Transform	The columns were transformed into a different data type for easy data manipulation.	<pre>data[["flow_duration", "fwd_pkts_per_sec", "bwd_pkts_per_sec", "flow_pkts_per_sec", "down_up_ratio"]].apply(pd.to_numeric) data.head()</pre> <table><thead><tr><th></th><th>no</th><th>id.orig_p</th><th>id.resp_p</th><th>proto</th><th>service</th><th>flow_duration</th><th>fwd_pkts_tot</th><th>bwd_pkts_tot</th><th>fwd_data_pkts_tot</th><th>bwd_data_pkts_tot</th><th>...</th><th>active_atd</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>38667</td><td>1883</td><td>tcp</td><td>mqtt</td><td>32.011598</td><td>9</td><td>5</td><td>3</td><td>3</td><td>...</td><td>0.0</td></tr><tr><td>1</td><td>1</td><td>51143</td><td>1883</td><td>tcp</td><td>mqtt</td><td>31.883584</td><td>9</td><td>5</td><td>3</td><td>3</td><td>...</td><td>0.0</td></tr><tr><td>2</td><td>2</td><td>44761</td><td>1883</td><td>tcp</td><td>mqtt</td><td>32.124053</td><td>9</td><td>5</td><td>3</td><td>3</td><td>...</td><td>0.0</td></tr><tr><td>3</td><td>3</td><td>60893</td><td>1883</td><td>tcp</td><td>mqtt</td><td>31.961063</td><td>9</td><td>5</td><td>3</td><td>3</td><td>...</td><td>0.0</td></tr><tr><td>4</td><td>4</td><td>51087</td><td>1883</td><td>tcp</td><td>mqtt</td><td>31.902362</td><td>9</td><td>5</td><td>3</td><td>3</td><td>...</td><td>0.0</td></tr></tbody></table> <p>5 rows × 85 columns</p>		no	id.orig_p	id.resp_p	proto	service	flow_duration	fwd_pkts_tot	bwd_pkts_tot	fwd_data_pkts_tot	bwd_data_pkts_tot	...	active_atd	0	0	38667	1883	tcp	mqtt	32.011598	9	5	3	3	...	0.0	1	1	51143	1883	tcp	mqtt	31.883584	9	5	3	3	...	0.0	2	2	44761	1883	tcp	mqtt	32.124053	9	5	3	3	...	0.0	3	3	60893	1883	tcp	mqtt	31.961063	9	5	3	3	...	0.0	4	4	51087	1883	tcp	mqtt	31.902362	9	5	3	3	...	0.0
	no	id.orig_p	id.resp_p	proto	service	flow_duration	fwd_pkts_tot	bwd_pkts_tot	fwd_data_pkts_tot	bwd_data_pkts_tot	...	active_atd																																																																				
0	0	38667	1883	tcp	mqtt	32.011598	9	5	3	3	...	0.0																																																																				
1	1	51143	1883	tcp	mqtt	31.883584	9	5	3	3	...	0.0																																																																				
2	2	44761	1883	tcp	mqtt	32.124053	9	5	3	3	...	0.0																																																																				
3	3	60893	1883	tcp	mqtt	31.961063	9	5	3	3	...	0.0																																																																				
4	4	51087	1883	tcp	mqtt	31.902362	9	5	3	3	...	0.0																																																																				
Load	The output came to be to result the number per each attack types, as well as the average time for it.	<pre># Get the time duration each attack types  data_attack_type = data.groupby(["Attack_type"])["flow_duration"].mean() data_attack_type  Attack_type ARP_poisoning      15.893538 DDOS_Slowloris     14.699148 DOS_SYN_Hping       0.000003 MQTT_Publish       43.397013 Metasploit_Brute_Force_SSH  3.006557 NIAP_FIN_SCAN       0.023614 NIAP_OS_DETECTION   0.000008 NIAP_TCP_scan       0.000019 NIAP_UDP_SCAN       0.737766 NIAP_XMAS_TREE_SCAN  0.001171 Thing_Speak         0.934471 Wipro_bulb         586.845727 Name: flow_duration, dtype: float64  # Get the number of each attack types  data_attack_type_1 = data.groupby(["Attack_type"])["Attack_type"].count() data_attack_type_1  Attack_type ARP_poisoning      7750 DDOS_Slowloris     534 DOS_SYN_Hping     94659 MQTT_Publish       4146 Metasploit_Brute_Force_SSH  37 NIAP_FIN_SCAN       28 NIAP_OS_DETECTION  2000 NIAP_TCP_scan       1002 NIAP_UDP_SCAN      2590 NIAP_XMAS_TREE_SCAN 2010 Thing_Speak        8108 Wipro_bulb         253 Name: Attack_type, dtype: int64</pre>																																																																														

To conclude, the midterm quiz done was to