CMSC178DA

Activity 2

EDA by Generating a Pair Plot

Submitted by:

Peladas, Daenielle Rai

Section B (TF 10:30 AM - 12:00 PM)

The Case

A data analyst was given the task by the head of the MIS office to look into the performance of a remote procedure call (RPC) mechanism that was compared between two (2) mainframe operating systems (OS) - UNIX and ARGUS. The performance metric was total elapsed time, which was measured for various data sizes. The measurements are presented below:

Unix		Argus	
Data Bytes	Time	Data Bytes	Time
64	26.4	92	32.8
64	26.4	92	34.2
64	26.4	92	32.4
64	26.2	92	34.4
234	33.8	348	41.4
590	41.6	604	51.2
846	50.0	860	76.0
1060	48.4	1074	80.8
1082	49.0	1074	79.8
1088	42.0	1088	58.6
1088	41.8	1088	57.6
1088	41.8	1088	59.8
1088	42.0	1088	57.4

The data analyst poses this question to himself: Is there a relationship between the various data (byte) sizes and the total elapsed time of RPCs between UNIX and ARGUS?

Data Preprocessing

The `pandas` library was imported for data manipulation and analysis.

```python
import pandas as pd
```

Two dictionaries were created to store the data bytes and its corresponding time taken to be processed for the two different operating systems, Unix and Argus. Each dictionary was then converted into a separate dataframe using the DataFrame constructor from the pandas library.

```python
unix_rpc = {"Data Bytes":[64, 64, 64, 64, 234, 590, 846, 1060, 1082, 1088, 1088, 1088, 1088], "Time": [26.4, 26.4, 26.4, 26.2, 33.8, 41.6, 50.0, 48.4, 49.0, 42.0, 41.8, 41.8, 42]}
argus_rpc = {"Data Bytes":[92, 92, 92, 92, 348, 604, 860, 1074, 1074, 1088, 1088, 1088, 1088], "Time": [32.8, 34.2, 32.4, 34.4, 41.4, 51.2, 76.0, 80.8, 79.8, 58.6, 57.6, 59.8, 57.4]}

df1 = pd.DataFrame(unix_rpc)
df2 = pd.DataFrame(argus_rpc)
```

The `concat` function was used to efficiently combine DataFrames `df1` and `df2` horizontally (along columns). A new column named `OS` was created to identify the operating system for each data point. This column was assigned values of `"Unix"` for entries from `df1` and `"Argus"` for entries from `df2`, leveraging the lengths of the original DataFrames to ensure proper alignment.

```python
df = pd.concat([df1, df2])
df['OS'] = ['Unix'] * len(df1) + ['Argus'] * len(df2)
df
```

	Data Bytes	Time	OS
0	64	26.4	Unix
1	64	26.4	Unix
2	64	26.4	Unix
3	64	26.2	Unix
4	234	33.8	Unix
5	590	41.6	Unix
6	846	50.0	Unix
7	1060	48.4	Unix
8	1082	49.0	Unix
9	1088	42.0	Unix
10	1088	41.8	Unix
11	1088	41.8	Unix
12	1088	42.0	Unix

0	92	32.8	Argus
1	92	34.2	Argus
2	92	32.4	Argus
3	92	34.4	Argus
4	348	41.4	Argus
5	604	51.2	Argus
6	860	76.0	Argus
7	1074	80.8	Argus
8	1074	79.8	Argus
9	1088	58.6	Argus
10	1088	57.6	Argus
11	1088	59.8	Argus
12	1088	57.4	Argus

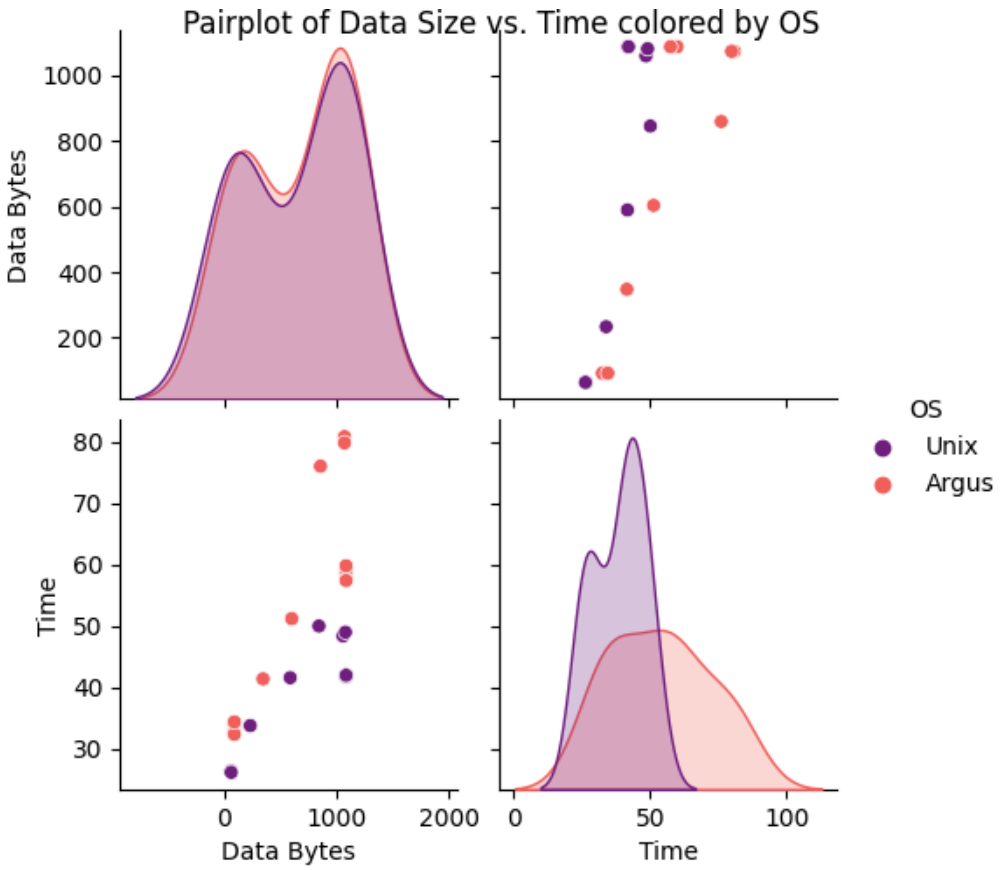
The Pair-Plot

The `seaborn` library is imported for creating boxplots, a visualization technique suitable for comparing distributions across multiple categories. The `matplotlib.pyplot` library (imported as `plt`) is imported for further customization of the plot.

```python
import seaborn as sns
import matplotlib.pyplot as plt
```

The `sns.pairplot` function is used to generate the pair-plot. The `df` dataframe is passed as input, containing the data bytes, time, and operating system type columns on both Argus and Unix. The differentiating color between all the variables will be based on the type of operating system as per specified in the `hue` input. The color `palette` is specified to `magma` to provide visually appealing color gradients within the boxes. Lastly, an appropriate `title` was added to make a more cohesive graphical representation of the data. In this case, the `suptitle` function was used for readability purposes.

```python
sns.pairplot(df, hue='OS', palette='magma')
plt.suptitle("Pairplot of Data Size vs. Time colored by OS", y=1.02)
plt.show()
```



Interpretation

An exploratory data analysis with pairplots was generated to explore the relationship between data size and total elapsed time (processing time) for RPCs executed on two operating systems (OS): UNIX and ARGUS.

The distribution of data sizes exhibited a bimodal characteristic, with two distinct peaks observed in the upper left histogram. This suggests that a significant portion of the data resided around two specific data size values, specifically points near 800 and 1000. Tis is apparent in both data sets that the observed overlap in data size ranges reduces the potential for significant bias in resulting processing time due to data size alone.

A positive correlation was evident between data size and processing time for both UNIX and ARGUS. This implies that as the size of the data being processed increases, the total elapsed time required for the RPC to be completely also increases. This aligns with the expected behavior of most computing tasks, where larger data volumes necessitate more processing time.

While a positive correlation exists for both OS, the upper right and lower left scatter plots revealed potential performance differences between UNIX and ARGUS. The data points for ARGUS displayed a steeper incline compared to UNIX as data size increased for both scatter plots. This suggests that processing time grew more rapidly for ARGUS when handling larger data sizes compared to UNIX.

There is also greater variability in the ARGUS data compared to UNIX for the processing time. This is evident in the distribution of data points across the upper right (data size vs. time), lower left (time vs. data size), and lower right (time histogram) panels. The ARGUS data points in the scatter plots (upper right and lower left) exhibit a wider spread compared to the tighter clusters observed for UNIX. Similarly, the ARGUS time histogram (lower right) displays a broader distribution resembling a bell curve, indicating less data concentration compared to the rightly skewed distribution of UNIX processing times. This suggests that ARGUS processing times exhibit greater inconsistency for similar data sizes, while UNIX demonstrates a tendency for faster and potentially more predictable processing times.

In conclusion, it is revealed that both UNIX and ARGUS exhibit a positive correlation between data size and RPC processing time. However, the analysis suggests potential performance advantages for UNIX. While there is a bimodal distribution of data sizes for both OS, with overlap mitigating bias concerns, ARGUS processing times display greater variability compared to UNIX. This variability is evident in the wider spread of data points within the scatter plots and the broader distribution of the ARGUS time histogram. Overall, these findings suggest that UNIX may offer faster and more consistent processing times for RPCs, particularly when dealing with larger data sizes.