Islamic University of Technology (IUT)

Organization of Islamic Cooperation (OIC)
Department of Electrical and Electronic Engineering (EEE)

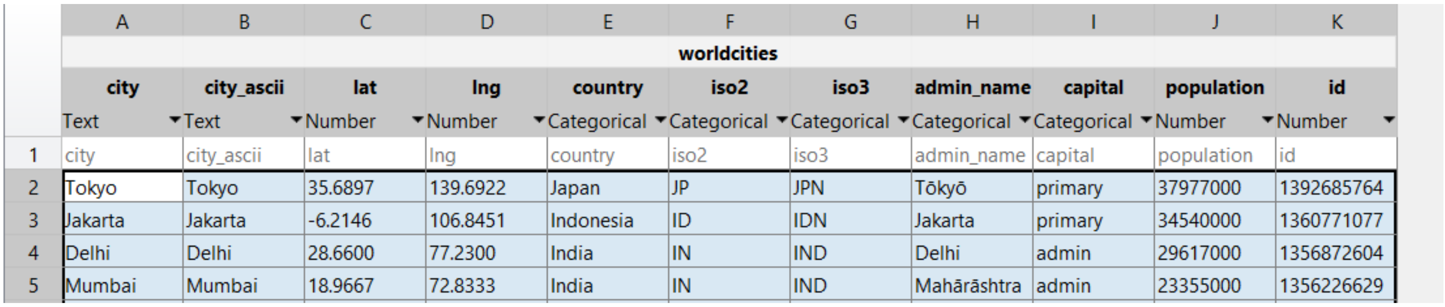
COURSE NO	**:**	**EEE 4416**
LAB NO	**:**	**08 (Part C)**
TOPIC	**:**	**Exploratory Data Analysis**

Exploratory Data Analysis

Exploratory Data Analysis (EDA) is the process of analyzing and summarizing datasets to uncover their main characteristics, often using visual methods and statistical techniques, before applying any machine learning models or formal analysis.

This notebook will provide a few basic guidelines for EDA using the ‘worldcities’ database.

1) Import the ‘Worldcities.csv’ file.
2) Check the variable data types before importing. As you can see, the variables are of type - text, number, and categorical.



	A	B	C	D	E	F	G	H	I	J	K
	worldcities										
	city	city_ascii	lat	lng	country	iso2	iso3	admin_name	capital	population	id
	Text	Text	Number	Number	Categorical	Categorical	Categorical	Categorical	Categorical	Number	Number
1	city	city_ascii	lat	lng	country	iso2	iso3	admin_name	capital	population	id
2	Tokyo	Tokyo	35.6897	139.6922	Japan	JP	JPN	Tōkyō	primary	37977000	1392685764
3	Jakarta	Jakarta	-6.2146	106.8451	Indonesia	ID	IDN	Jakarta	primary	34540000	1360771077
4	Delhi	Delhi	28.6600	77.2300	India	IN	IND	Delhi	admin	29617000	1356872604
5	Mumbai	Mumbai	18.9667	72.8333	India	IN	IND	Mahārāshtra	admin	23355000	1356226629

3) Summarize the table. [‘summary’ function]
4) Find how many missing entries there are in each column.

```
ismissing(worldcities)          % logical array of size 26562x11
sum(ismissing(worldcities))     % number of missing entries in each column
```

As you can see, 3 columns have missing entries – admin_name, capital, and population. What to do about these missing entries?

⇨ Check the ‘Missing Data Handling’ section.

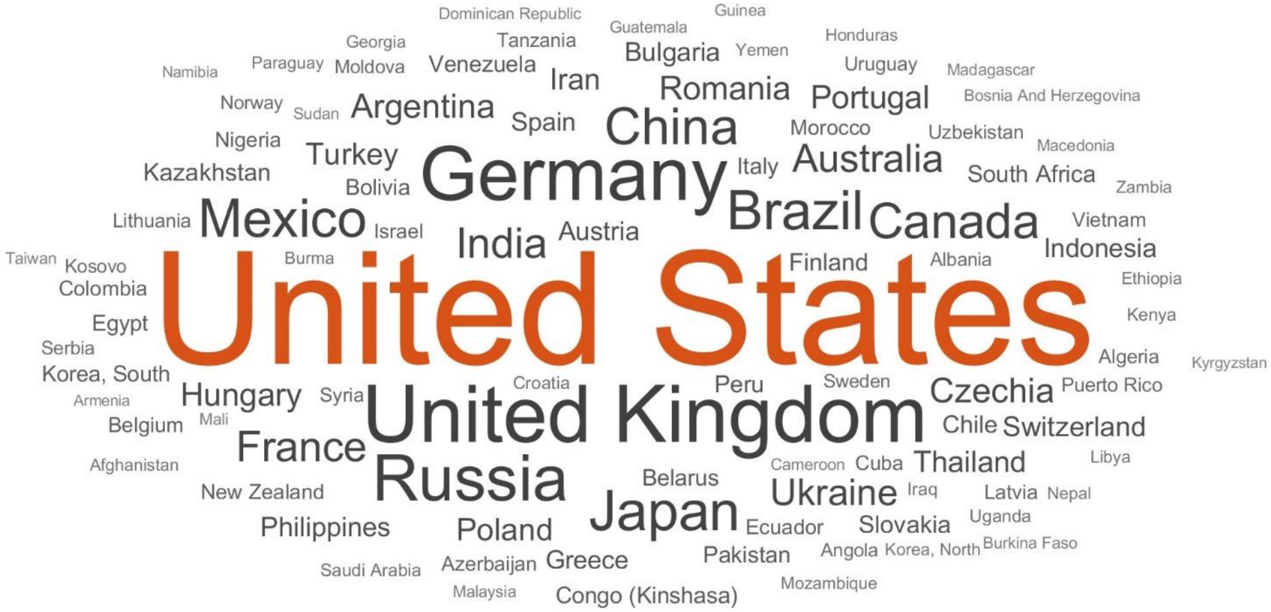
5) Remove some unnecessary columns, such as 2, 6, 7, 8, 9, and 11, from the table. The data now contains only 5 columns.

- using code
- using the **data viewer** window from the workspace

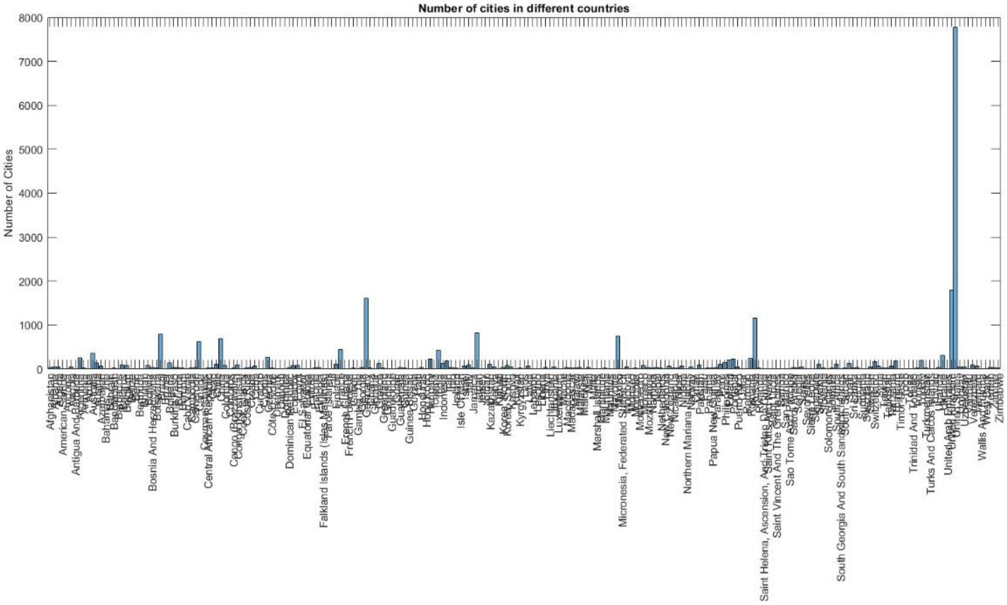
6) Remove the rows that have missing entries.

```
ismissing(worldcities.population)
worldcities(ismissing(worldcities.population),:) = []          % Boolean masking
```

7) See the word cloud of the country column.
It allows you to visualize which countries have a higher number of cities reported in the table. From the figure, the USA, UK, Germany, Russia, and Japan – these countries have more cities present in the data.



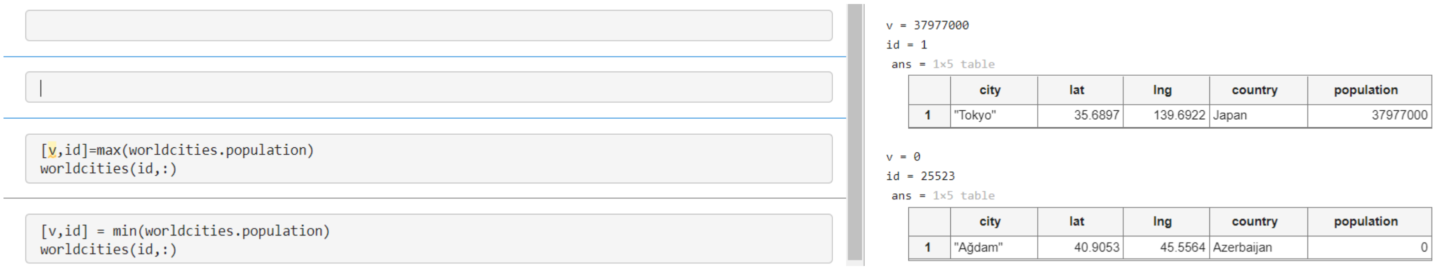
8) See the histogram of the country column (Use Plots app). Which country has more cities reported in the data? As you can see from the image, it is very difficult to pinpoint the country names as the figure has become very congested. What can we do about it?



9) Find out how many country names there are.
Since the 'country' is of categorical data type, it is easier to find the number of categories (i.e., the number of countries) in that column. If it were a string, it would not have been possible.

```
numel(categories(worldcities.country))      => 223
```

10) Find out which city has the highest and the lowest population.



```
[v,id]=max(worldcities.population)
worldcities(id,:)
```

```
[v,id] = min(worldcities.population)
worldcities(id,:)
```

```
v = 37977000
id = 1
ans = 1×5 table
```

	city	lat	lng	country	population
1	"Tokyo"	35.6897	139.6922	Japan	37977000

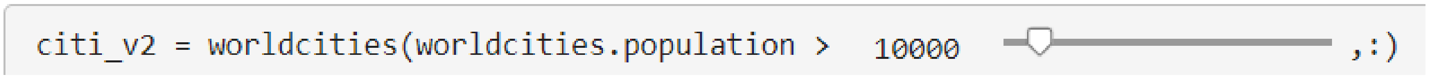
```
v = 0
id = 25523
ans = 1×5 table
```

	city	lat	lng	country	population
1	"Ağdam"	40.9053	45.5564	Azerbaijan	0

As you can see, Tokyo has the highest population. However, the city of Agdam in Azerbaijan is reported to have 0 population, which may not be accurate. How many more cities have been reported to have 0 population?

11) Sort the data based on population to get a better understanding of the data distribution (remember the 'sortrows' function?).
You will be able to observe that there are many cities with very small populations.

12) Let's create a subset of the original data with cities that have a population higher than 10000 or 50000.



```
citi_v2 = worldcities(worldcities.population >  10000  ,:)
```

In the code, I have used a **Numeric slider** to set the population. This provides better control and makes it easier to observe how many cities have a population higher than x.

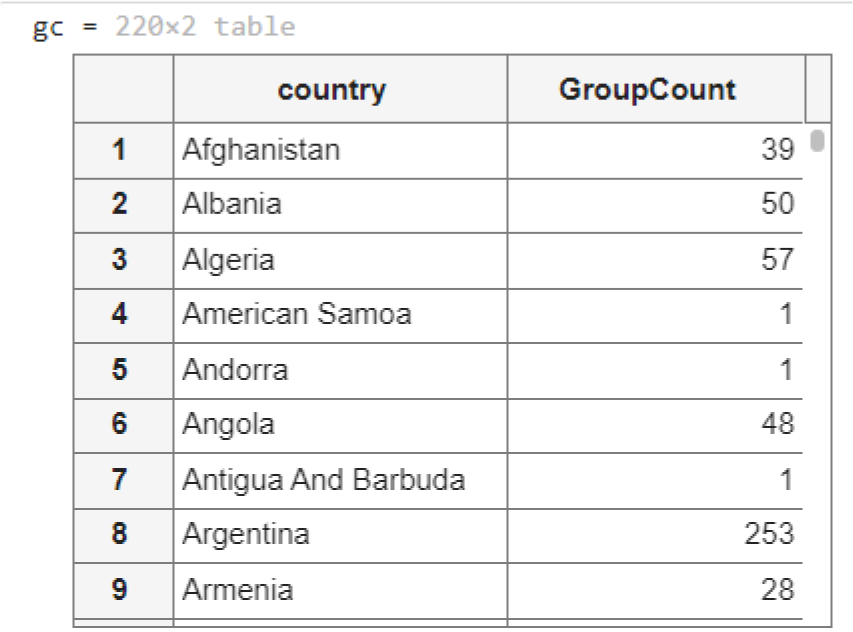
To place a numeric slider, go to Live Editor => Code => Control => Numeric slider. You will find other options such as drop-down box, button, etc, there as well.

You need to set the minimum, maximum, and step size for the numeric slider.

13) Previously, from the histogram plot, we could not obtain a clear idea of which country had how many cities in the data.
Another approach to obtain that information would be to group the data based on the country column. The 'groupsummary' function of MATLAB will count how many samples (rows) have the same country name.

```
groupsummary(worldcities,"country")
```

This function returns a table with two columns – country name and group count.



```
gc = 220×2 table
```

	country	GroupCount
1	Afghanistan	39
2	Albania	50
3	Algeria	57
4	American Samoa	1
5	Andorra	1
6	Angola	48
7	Antigua And Barbuda	1
8	Argentina	253
9	Armenia	28

14) From the above group summary table, create a subset with countries that have a groupcount value more than x. Use the numeric slider to set the value of x.

Missing Data Handling

An important part of the data preprocessing task is to handle the missing entries. It can be done in either of the following ways –

i. Remove the samples with missing entries (usually not preferred as it removes the entire data point, just because only one variable's data is missing).
ii. Remove the column – done only when the variable has too many missing entries (> 60%).
iii. Fill in the missing entries with **imputation** techniques. There are various imputation techniques, some of which are quite reliable.

- MATLAB has a ‘fillmissing’ function to perform data imputation.