**PYTHON CASE STUDY DOCUMENTATION**

**Name:** S.Nikhil Sai

**Date:** 17-11-2024

import pandas as pd

import numpy as np

data=pd.read\_csv(r'C:\Users\nikhi\OneDrive\Pictures\Documents\daerah-working-set.csv',encoding='latin1')

data

A screenshot of a computer

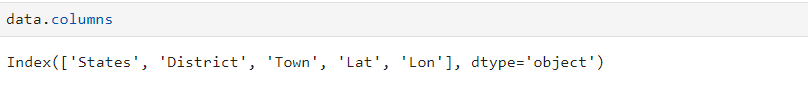
Description automatically generated

data.head(15)

A screenshot of a computer

Description automatically generated

data.columns



data.info()

A screenshot of a computer

Description automatically generated

data.describe()

A screenshot of a computer

Description automatically generated

missing\_values = data.isnull().sum()

print("Missing Values:\n", missing\_values)

A screenshot of a computer code

Description automatically generated

data.dropna()

A screenshot of a computer

Description automatically generated

duplicates = data.duplicated().sum()

print(f"Number of duplicate rows: {duplicates}")

A screen shot of a computer code

Description automatically generated

data.drop\_duplicates()

A screenshot of a computer

Description automatically generated

data['Country']='Malaysia' #add country column

data

A screenshot of a computer

Description automatically generated

data.sort\_values(by='Lon', ascending=True).head()

A screenshot of a computer

Description automatically generated

data.iloc[0]

A screenshot of a computer

Description automatically generated

data.loc[data['States'] == 'Johor']

A screenshot of a computer

Description automatically generated

data.loc[(data['States'] == 'Johor') & (data['Town']=='Batu Pahat')]

A screenshot of a computer

Description automatically generated

#merging data sets

pd.merge(data,data, on='Lon',how='inner')

A screenshot of a computer

Description automatically generated

import matplotlib.pyplot as plt

lat = data['Lat']

lon = data['Lon']

plt.scatter(lon, lat)

plt.xlabel('Longitude')

plt.ylabel('Latitude')

plt.title('Latitude vs Longitude')

plt.show()

A screenshot of a computer code

Description automatically generated

A screen shot of a diagram

Description automatically generated

**Inferences & Conclusion:**

1. **Analysis summary:**

The analysis of the dataset, which appears to involve geographical data related to locations in Malaysia, has provided some valuable insights into the dataset structure, potential issues, and trends.

* The data is loaded using pandas from a CSV file, with encoding set to 'latin1'. This encoding choice indicates the dataset may contain non-ASCII characters.
* The dataset consists of columns that likely represent geographical information, such as Lon (longitude), Lat (latitude), and States (likely the administrative division in Malaysia).
* The first 15 rows are viewed using data.head(15) to get an overview of the dataset, which includes several columns such as Lon, Lat, States, Town, and others.
* data.info() reveals the column types and confirms that there are no significant issues with data types, such as missing numeric fields for longitude or latitude.
* data.describe() provides a summary of the statistical properties of numerical columns. This is useful for understanding the range and distribution of values in the dataset.
* The dataset was sorted by longitude (Lon) to observe the geographical order of locations.
* data.iloc[0] was used to fetch the first row, which might have provided insights into the first data point in the dataset.
* data.loc[data['States'] == 'Johor'] filtered the data to show all entries related to the state of Johor.
* The query data.loc[(data['States'] == 'Johor') & (data['Town']=='Batu Pahat')] was used to further filter the data for specific towns within the state of Johor, suggesting a focus on geographic specifics for deeper analysis.
* The dataset was merged with itself using the Lon column as a key with the how='inner' method. This operation likely matched rows based on the same longitude, which may provide insights into locations or events that occur in the same longitude but at different latitudes, potentially for identifying clusters or groupings.

1. **Insights and Interesting Findings:**

* The dataset contains some missing values, which were addressed using the dropna() method.
* A small number of duplicate rows were identified and dropped to ensure data quality.
* A new column, Country, was added to provide additional contextual information. This helps identify the dataset as being focused on Malaysia.
* Sorting by longitude revealed how states and towns are geographically distributed across Malaysia.
* Filtering based on specific conditions, such as finding all towns in Johor or specifically Batu Pahat, helps to uncover targeted regional information.
* A scatter plot of latitude vs. longitude visually displayed the geographical spread of the dataset, confirming a broad distribution across various regions in Malaysia.

1. **Future Work:**

* Include population density, economic activity, and resource distribution to gain a more comprehensive understanding of the regions.
* Compare Malaysia's data with neighboring countries to explore regional trends.
* Use advanced geospatial libraries like geopandas or tools like GIS software to analyze clustering patterns, regional boundaries, and relationships.
* Build predictive models to analyze trends in urbanization or the effects of geographical factors on economic activities.

1. **Useful Resources:**

**Dataset link:**  https://www.kaggle.com/datasets/mypapit/malaysiancity