Fundamentals of Data Security

By Prof.Vishal Chugh

What is Data Security ?

Data security is the practice of protecting digital information from unauthorized access, corruption or loss throughout its entire lifecycle

Process to Secure and Access the Data

Encryption

Encryption is the process of converting information into a coded format, making it unreadable to anyone without the proper key

Decryption

It is the process of converting encrypted data back into its original, readable form

Major types of Encryption used in Industry

Symmetric

A single secret key is used to encrypt and decrypt data.

Asymmetric

Uses a pair of keys: a public key that can be shared with anyone for encryption and a private key that must be kept secret for decryption.

Pros

Symmetric

Generally faster and more efficient than asymmetric encryption, making it suitable for encrypting large amounts of data.

Asymmetric

Offers better security for key distribution, as the private key never needs to be shared.

Cons

Symmetric

Requires secure key distribution, as the same key must be shared between the sender and recipient.

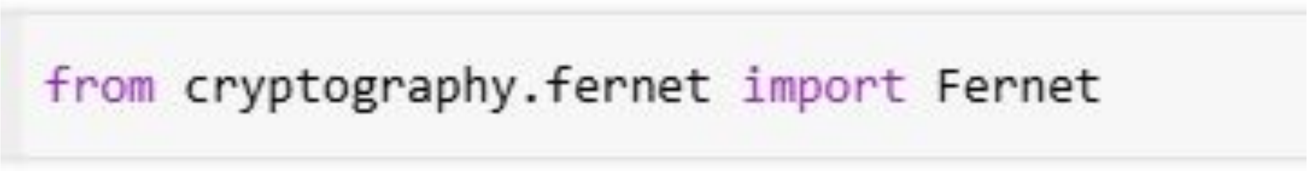
Asymmetric

Generally slower and less efficient than symmetric encryption, making it less suitable for large data volumes.

Encryption in Python

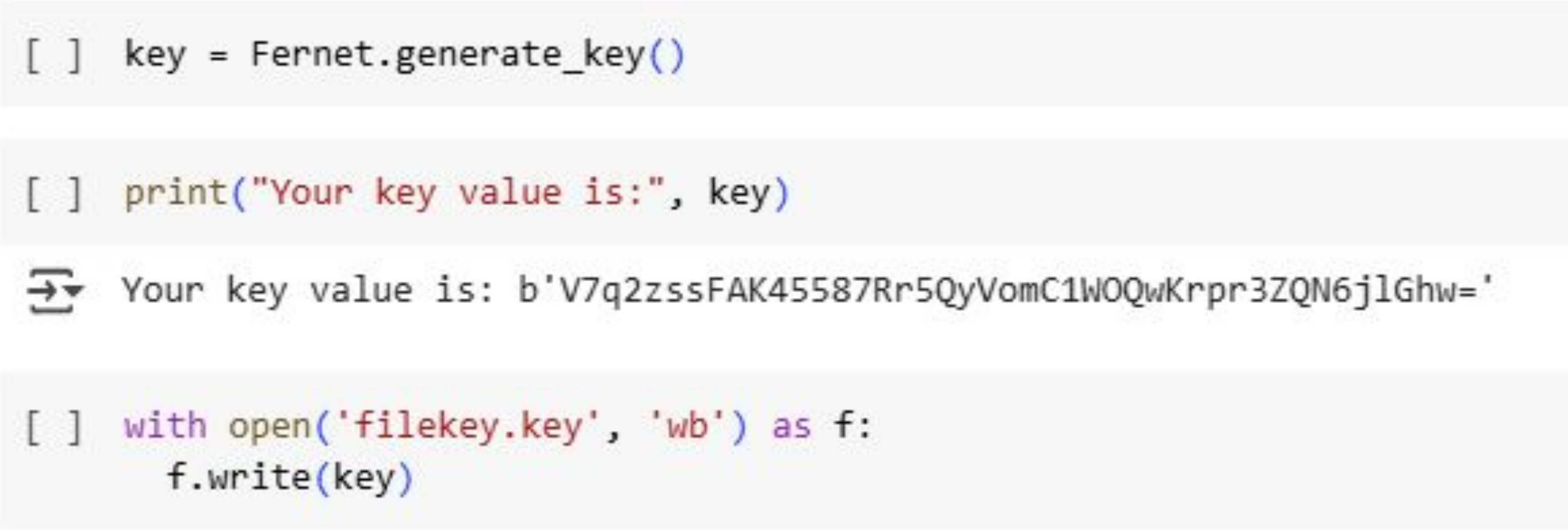
By Prof.Vishal Chugh

Import the Fernet from Cryptography



```
from cryptography.fernet import Fernet
```

Generate Key and store it in .key file



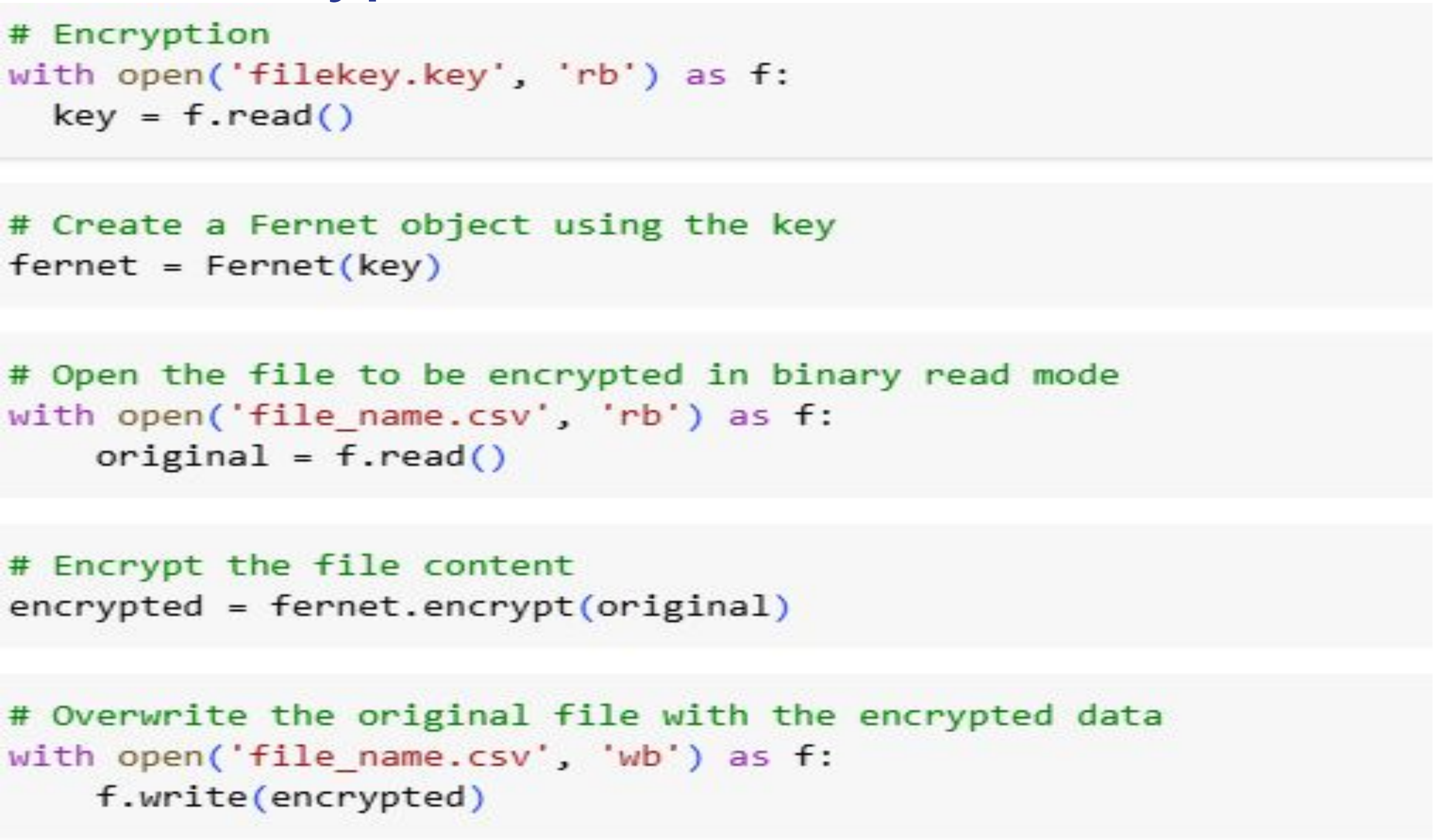
```
[ ]  key = Fernet.generate_key()
```

```
[ ]  print("Your key value is:", key)
```

```
Your key value is: b'V7q2zssFAK45587Rr5QyVomC1WOQwKrpr3ZQN6jlGhw='
```

```
[ ]  with open('filekey.key', 'wb') as f:
       f.write(key)
```

Create Encryption



```python
# Encryption
with open('filekey.key', 'rb') as f:
  key = f.read()
```

```python
# Create a Fernet object using the key
fernet = Fernet(key)
```

```python
# Open the file to be encrypted in binary read mode
with open('file_name.csv', 'rb') as f:
    original = f.read()
```

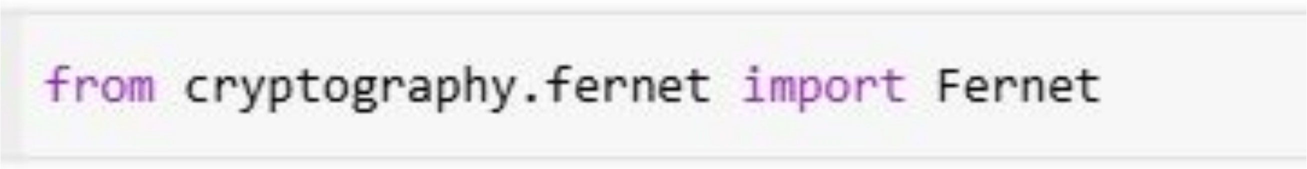
```python
# Encrypt the file content
encrypted = fernet.encrypt(original)
```

```python
# Overwrite the original file with the encrypted data
with open('file_name.csv', 'wb') as f:
    f.write(encrypted)
```

Decryption in Python

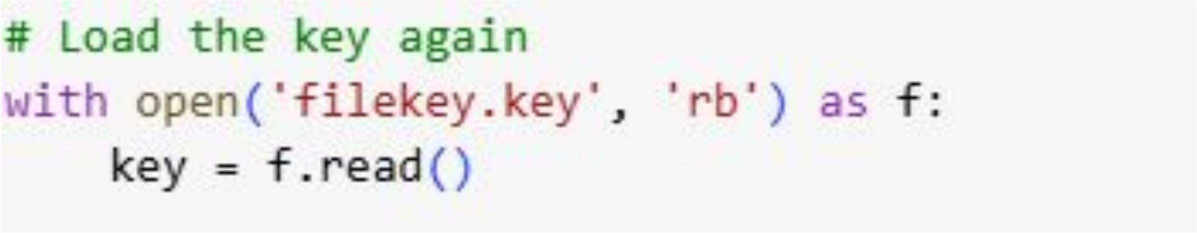
By Prof.Vishal Chugh

Import the Fernet from Cryptography

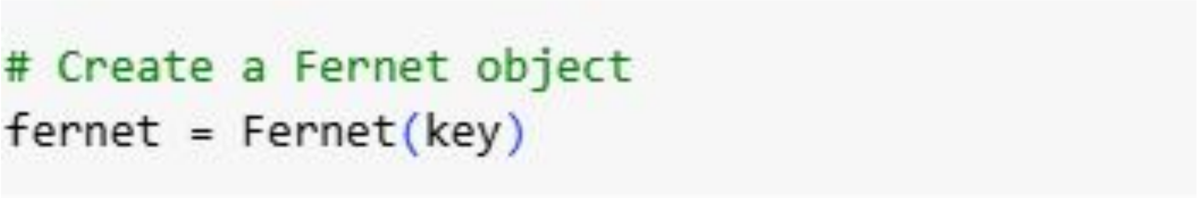


```
from cryptography.fernet import Fernet
```

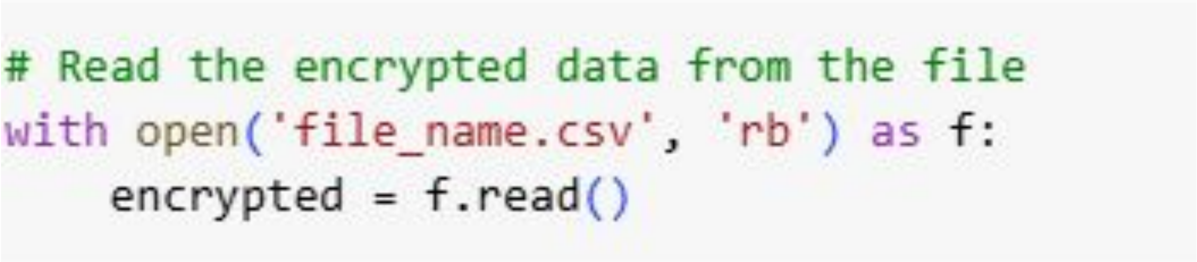
Load .key file and encrypted data file



```python
# Load the key again
with open('filekey.key', 'rb') as f:
    key = f.read()
```

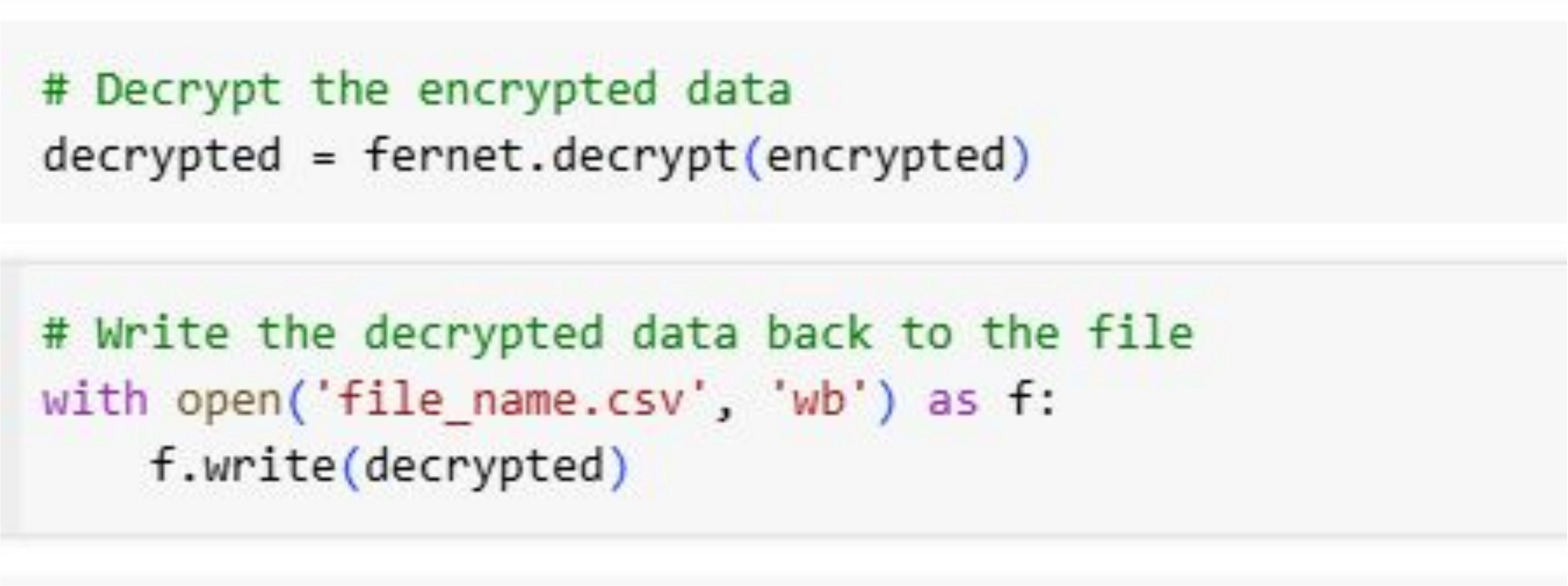


```python
# Create a Fernet object
fernet = Fernet(key)
```



```python
# Read the encrypted data from the file
with open('file_name.csv', 'rb') as f:
    encrypted = f.read()
```

Create Decryption

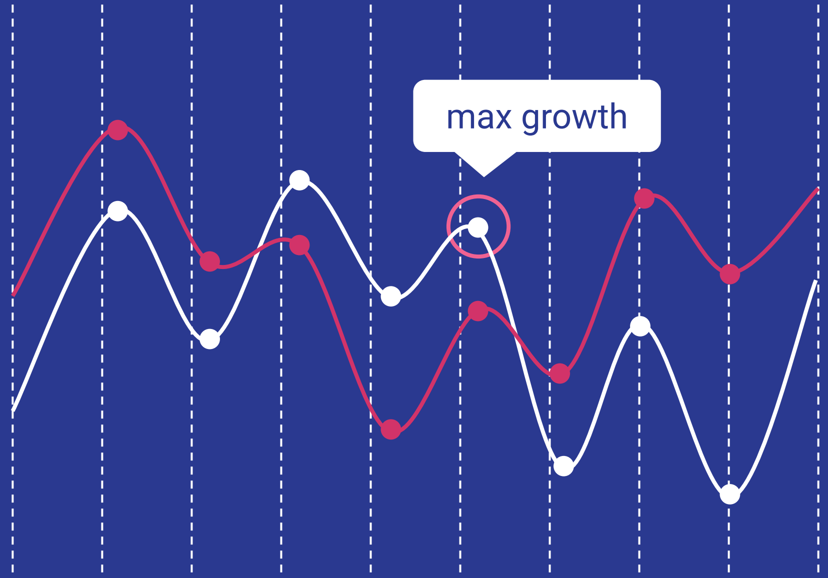


```python
# Decrypt the encrypted data
decrypted = fernet.decrypt(encrypted)
```

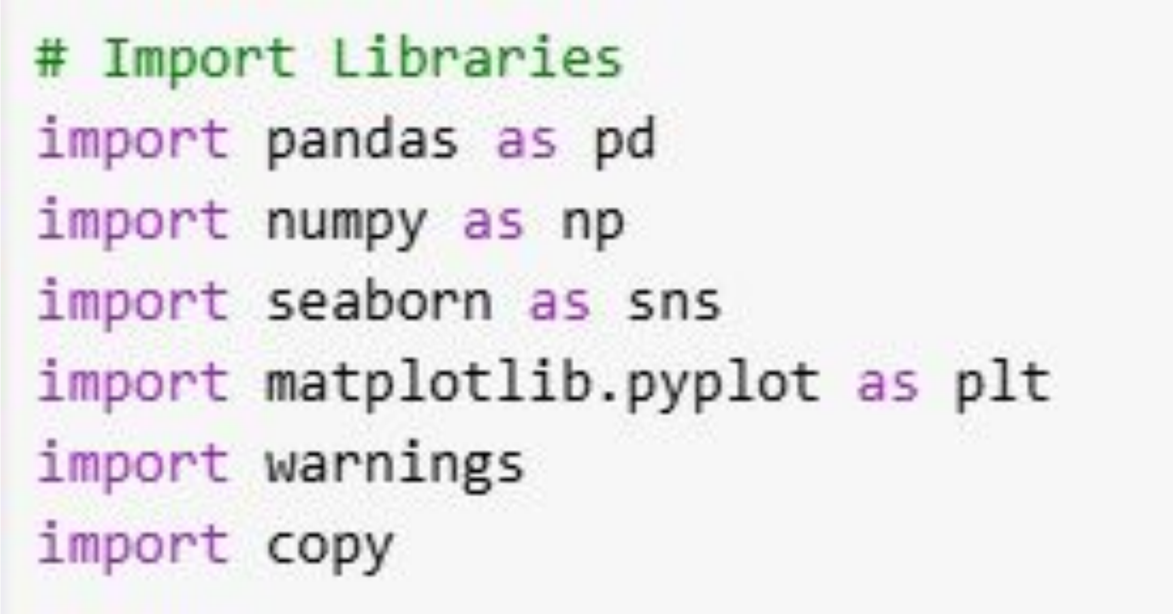
```python
# Write the decrypted data back to the file
with open('file_name.csv', 'wb') as f:
    f.write(decrypted)
```

Common Steps in Python

By Prof.Vishal Chugh

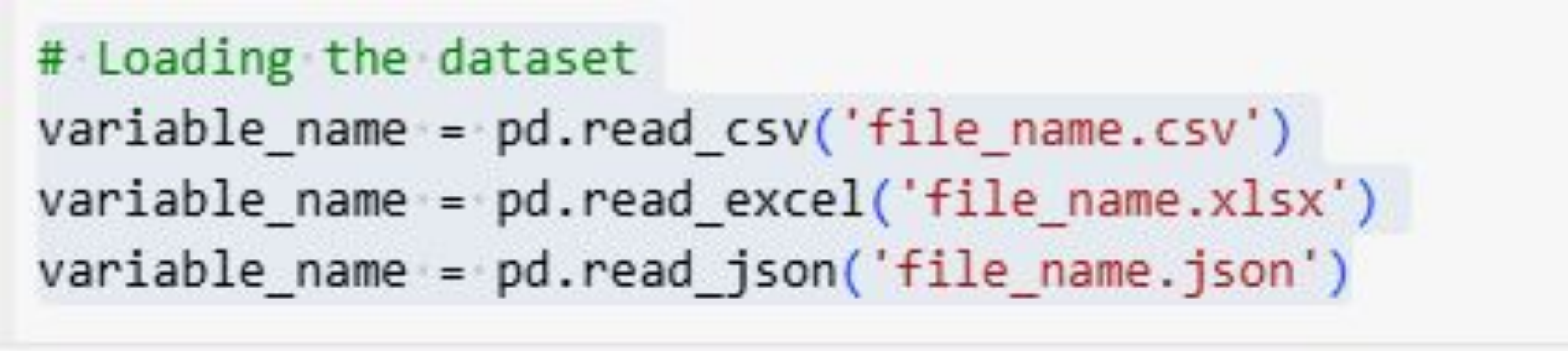


First Step Import Necessary Libraries



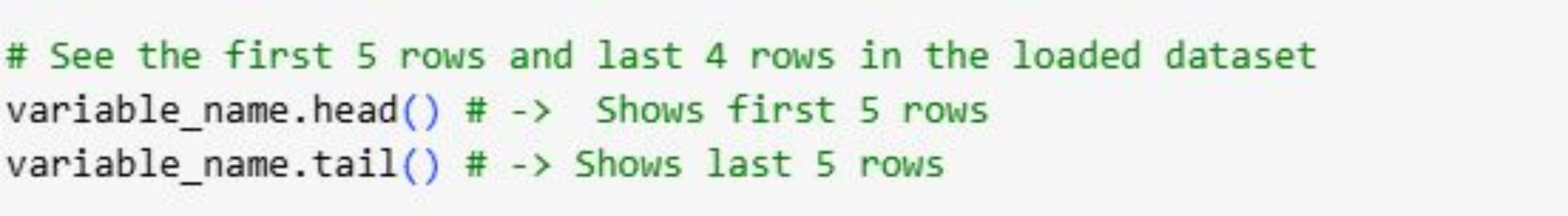
```
# Import Libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
import copy
```

Load data in dataframe using pandas



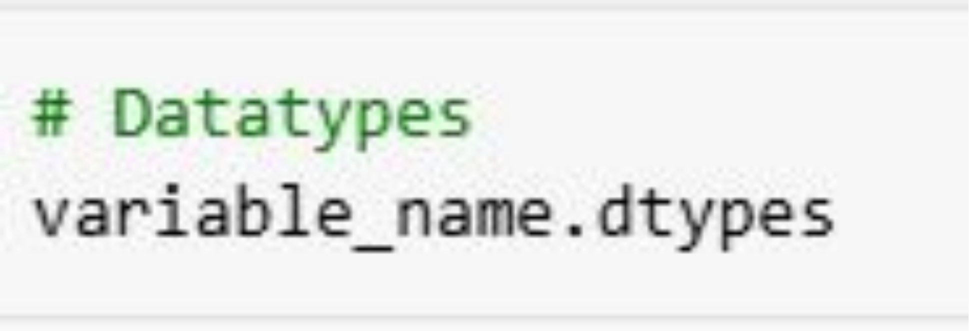
```
# Loading the dataset
variable_name = pd.read_csv('file_name.csv')
variable_name = pd.read_excel('file_name.xlsx')
variable_name = pd.read_json('file_name.json')
```

See the first 5 rows and last 5 rows of data



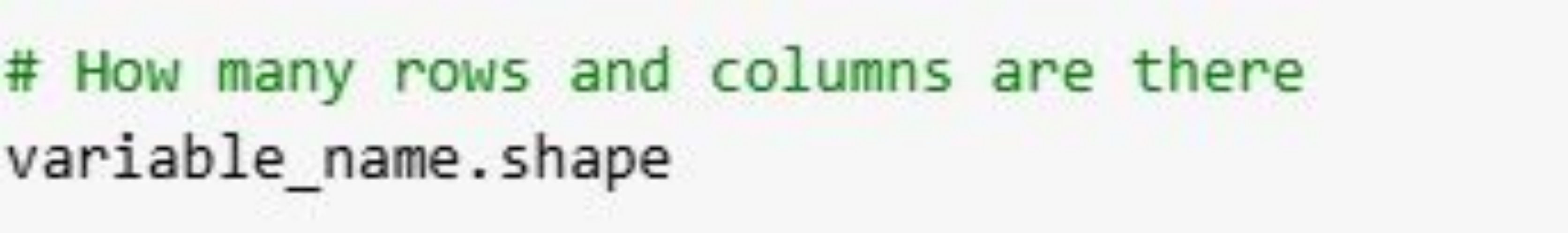
```
# See the first 5 rows and last 4 rows in the loaded dataset
variable_name.head() # ->  Shows first 5 rows
variable_name.tail() # -> Shows last 5 rows
```

Inspecting the data types of all columns



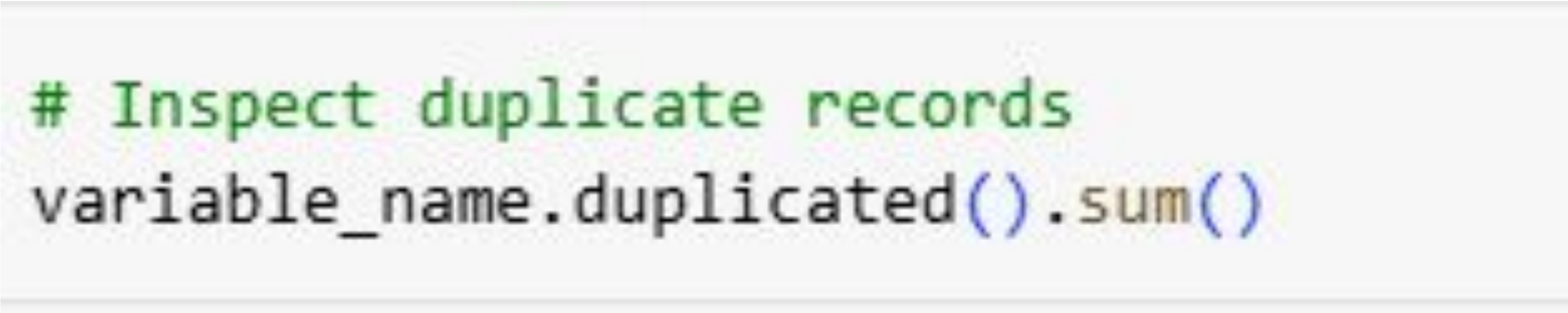
```
# Datatypes
variable_name.dtypes
```

Inspecting the number of rows and column



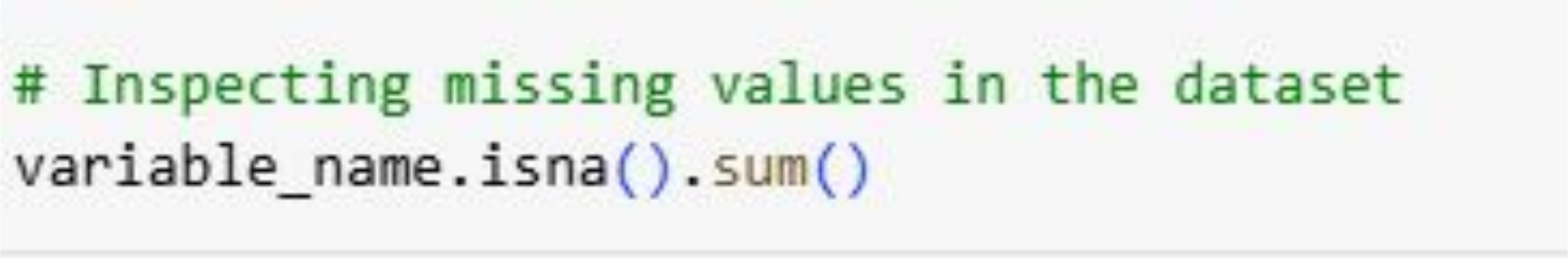
```
# How many rows and columns are there
variable_name.shape
```

Inspecting the Duplicate Records



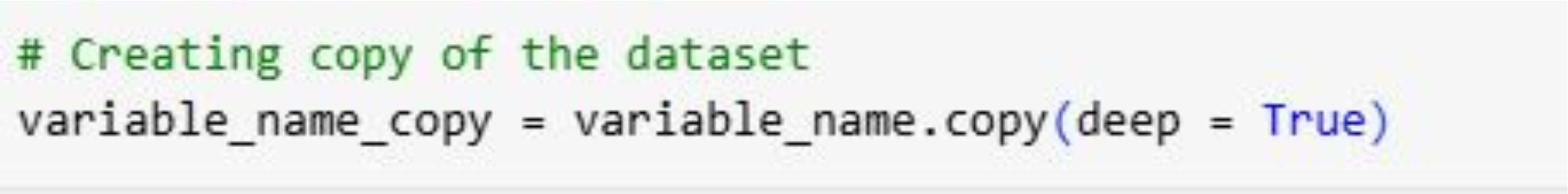
```
# Inspect duplicate records
variable_name.duplicated().sum()
```

Inspecting the missing values



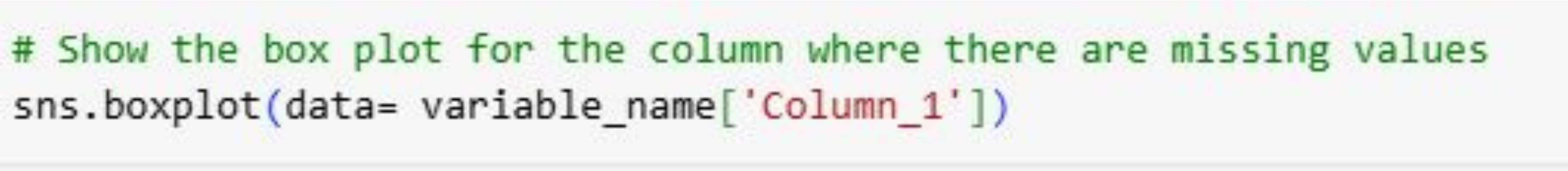
```
# Inspecting missing values in the dataset
variable_name.isna().sum()
```

Creating copy of the original dataset



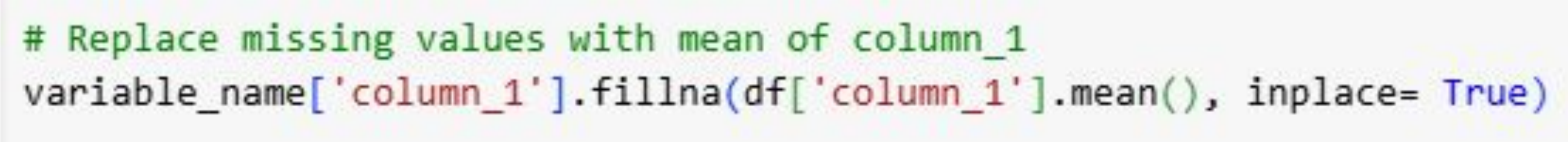
```
# Creating copy of the dataset
variable_name_copy = variable_name.copy(deep = True)
```

Creating Box Plot in missing values column to see outliers



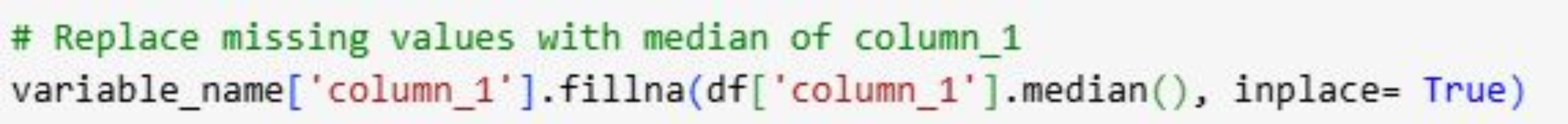
```
# Show the box plot for the column where there are missing values
sns.boxplot(data= variable_name['Column_1'])
```

For numeric data without outliers we use mean to fill missing values



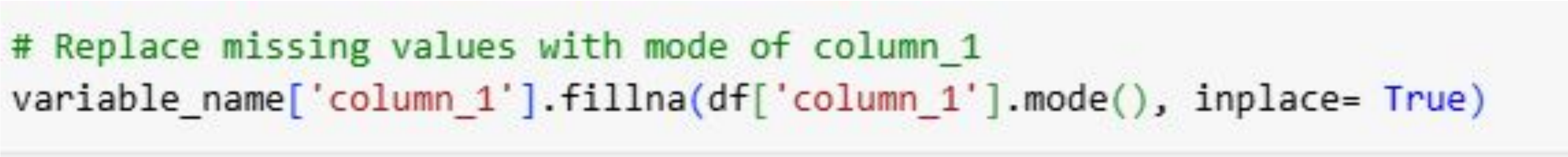
```
# Replace missing values with mean of column_1
variable_name['column_1'].fillna(df['column_1'].mean(), inplace= True)
```

For numeric data with outliers we use Median to fill missing values



```
# Replace missing values with median of column_1
variable_name['column_1'].fillna(df['column_1'].median(), inplace= True)
```

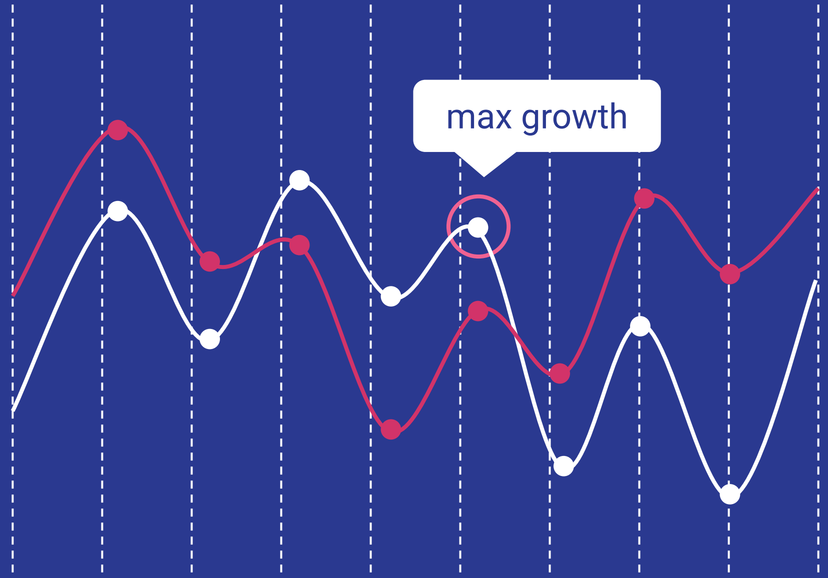
For Categorical data we use mode to fill missing values



```
# Replace missing values with mode of column_1
variable_name['column_1'].fillna(df['column_1'].mode(), inplace= True)
```

Case Study

By Prof.Vishal Chugh



Case Study: Customer Persona for "TrendMart"

Background

TrendMart is a retail & e-commerce lifestyle brand aiming to personalize marketing strategies. They have collected both **quantitative** (demographics, shopping behavior) and **qualitative** (goals, challenges, lifestyle) customer data.

The marketing team wants to group customers into distinct personas to:

- Target promotions effectively
- Choose the right communication channels
- Improve customer retention

Questions

Part 1: Data Preparation

1. Import both datasets in Python and merge them using `Customer_ID` as the key.
2. Display the first 5 rows of the merged dataset.

Part 2: Persona Grouping by Demographics

3. Group customers by **Location** (Urban/Rural) and find the **average Annual Income**.
4. Find the **most common Communication Preference** for Urban customers.
5. Group customers by **Technology Usage** and calculate the **average Online Purchase Frequency**.

Part 3: Persona Grouping by Qualitative Data

6. Find the number of customers whose **Goals and Aspirations** is `"Better health and fitness"`.
7. Group customers by **Lifestyle and values** and find the **average Annual Income**.
8. Find the most common **Decision-Making trigger** for customers with **High Technology Usage**.

Part 4: Combined Segmentation

9. Create a segment of customers who are:
    - Urban
    - Annual Income > ₹1,000,000
    - Technology Usage = "High"
      Display their **Name, Age, Occupation, and Goals**.

10. Count how many customers have `"Discounts and offers"` as their **Decision-Making trigger and** shop online more than 5 times