

Experiment No. 1

Aim:

Data Wrangling, I

Perform the following operations using Python on any open source dataset (e.g., data.csv)

- 1) Import all the required Python Libraries.
 - 2) Locate an open source data from the web (e.g., <https://www.kaggle.com>). Provide a clear description of the data and its source (i.e., URL of the web site).
 - 3) Load the Dataset into pandas dataframe.
 - 4) Data Preprocessing: check for missing values in the data using pandas `isnull()`, `describe()` function to get some initial statistics. Provide variable descriptions. Types of variables etc. Check the dimensions of the data frame.
 - 5) Data Formatting and Data Normalization: Summarize the types of variables by checking the data types (i.e., character, numeric, integer, factor, and logical) of the variables in the data set. If variables are not in the correct data type, apply proper type conversions.
 - 6) Turn categorical variables into quantitative variables in Python.
-

Requirement:

- Anaconda Installer
- Windows 10 OS
- Jupyter Notebook

Theory:

Data wrangling is the process of cleaning and unifying messy and complex data sets for easy access and analysis.

With the amount of data and data sources rapidly growing and expanding, it is getting increasingly essential for large amounts of available data to be organized for analysis. This process typically includes manually converting and mapping data from one raw form into another format to allow for more convenient consumption and organization of the data.

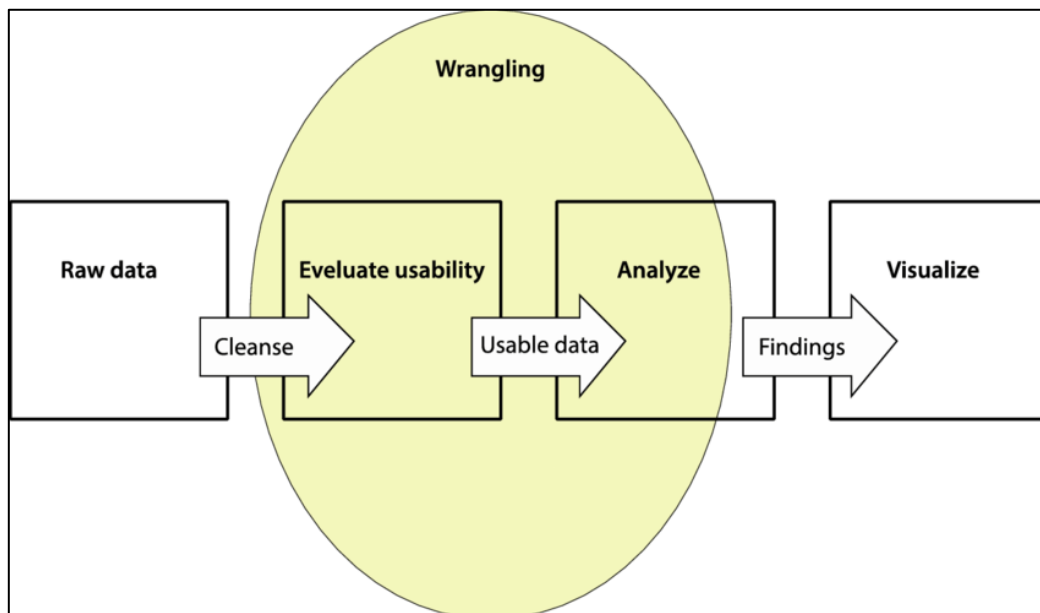
The Goals of Data Wrangling:

- Reveal a "deeper intelligence" by gathering data from multiple sources
- Provide accurate, actionable data in the hands of business analysts in a timely matter

- Reduce the time spent collecting and organizing unruly data before it can be utilized
- Enable data scientists and analysts to focus on the analysis of data, rather than the wrangling
- Drive better decision-making skills by senior leaders in an organization

Key steps to Data Wrangling:

- Data Acquisition: Identify and obtain access to the data within your sources.
- Joining Data: Combine the edited data for further use and analysis.
- Data Cleansing: Redesign the data into a usable and functional format and correct/remove any bad data.



Libraries Used:

Pandas: Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring and manipulating data.

Numpy: NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices.

Conclusion:

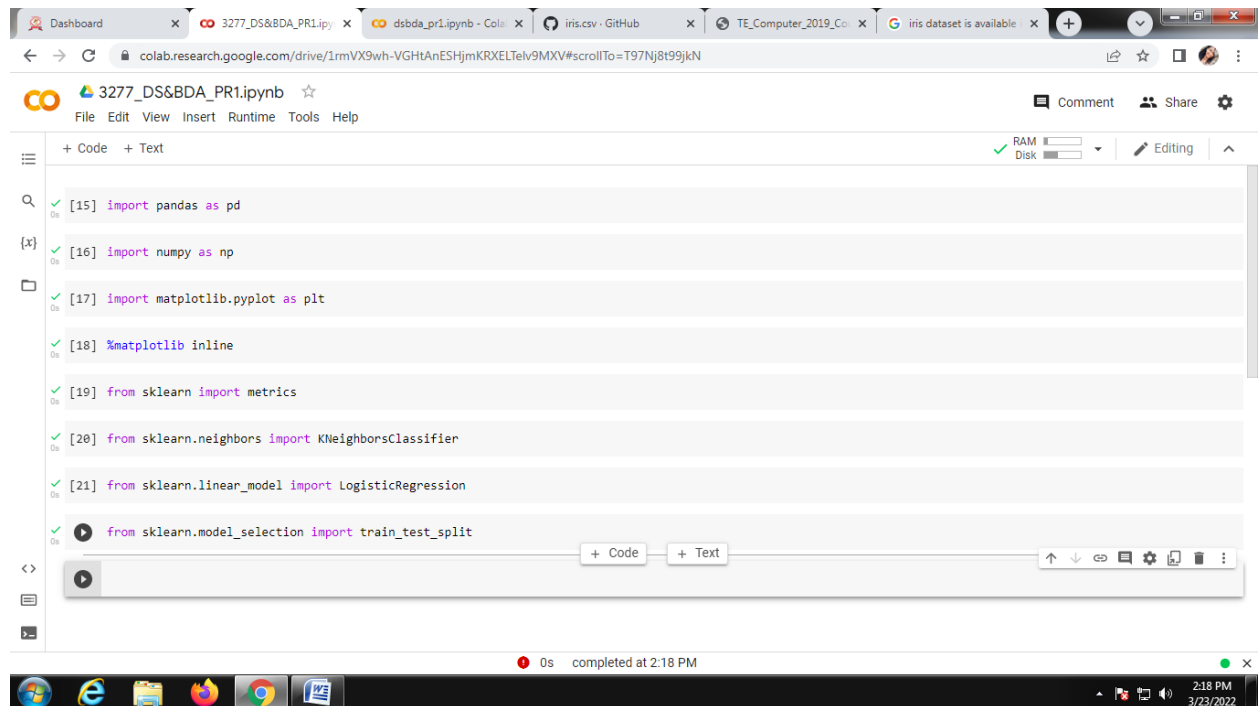
Hence, we have implemented data wrangling practical.

Practical 1

Data Wrangling I

Perform the following operations using Python on any open source dataset (e.g., data.csv)

1. Import all the required Python Libraries



The screenshot shows a Google Colab notebook interface. The browser tabs at the top include 'Dashboard', '3277_DS&BDA_PR1.ipynb', 'dsbda_pr1.ipynb - Colab', 'iris.csv - GitHub', 'TE_Computer_2019_Col', and 'iris dataset is available'. The notebook's address bar shows a Google Drive link. The notebook title is '3277_DS&BDA_PR1.ipynb'. The menu bar includes 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. On the right, there are buttons for 'Comment', 'Share', and 'Settings', along with RAM and Disk usage indicators. The code editor shows the following imports:

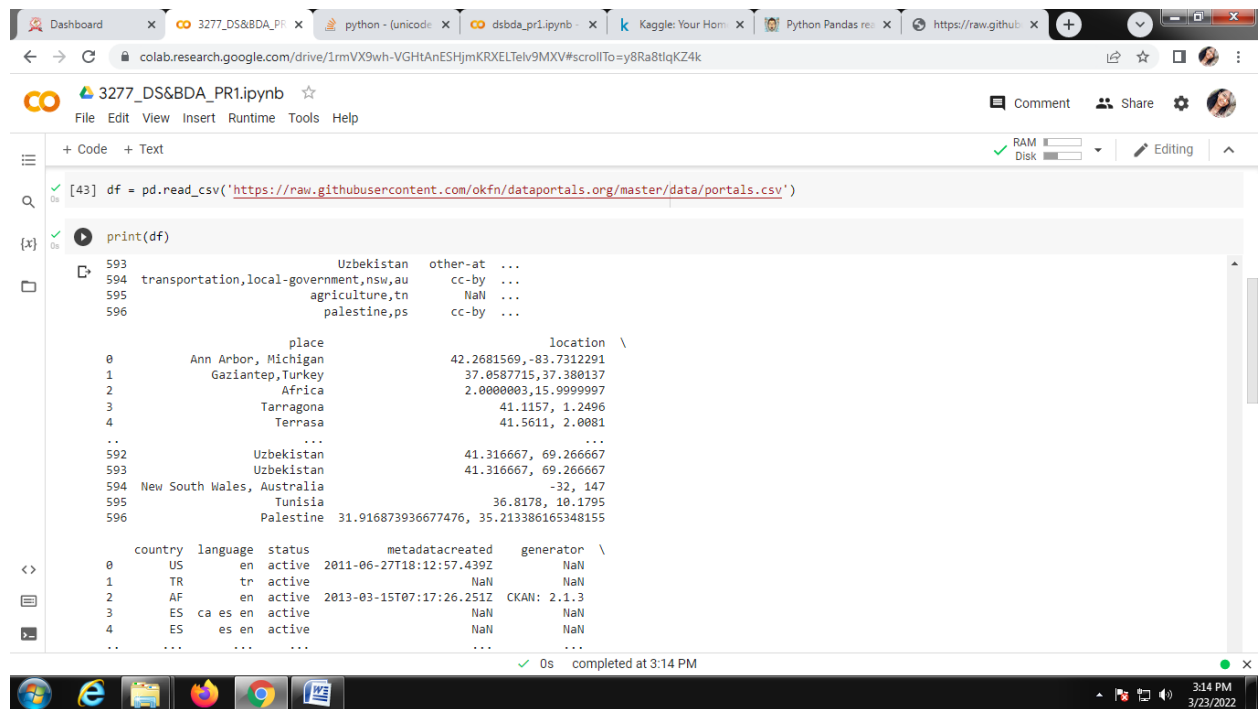
```
[15] import pandas as pd
[16] import numpy as np
[17] import matplotlib.pyplot as plt
[18] %matplotlib inline
[19] from sklearn import metrics
[20] from sklearn.neighbors import KNeighborsClassifier
[21] from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
```

At the bottom, a status bar indicates '0s completed at 2:18 PM' and the system clock shows '2:18 PM 3/23/2022'.

2. Locate an open source data from the web (e.g. <https://www.kaggle.com>). Provide a clear description of the data and its source (i.e. URL of the web site).

Description: I've used portals dataset for this practical which is freely available on internet on data portals site.

3. Load the Dataset into Pandas DataFrame.



The screenshot shows a Google Colab notebook titled '3277_DS&BDA_PR1.ipynb'. The code cell contains the following Python code:

```
[43] df = pd.read_csv('https://raw.githubusercontent.com/okfn/dataportals.org/master/data/portals.csv')

print(df)
```

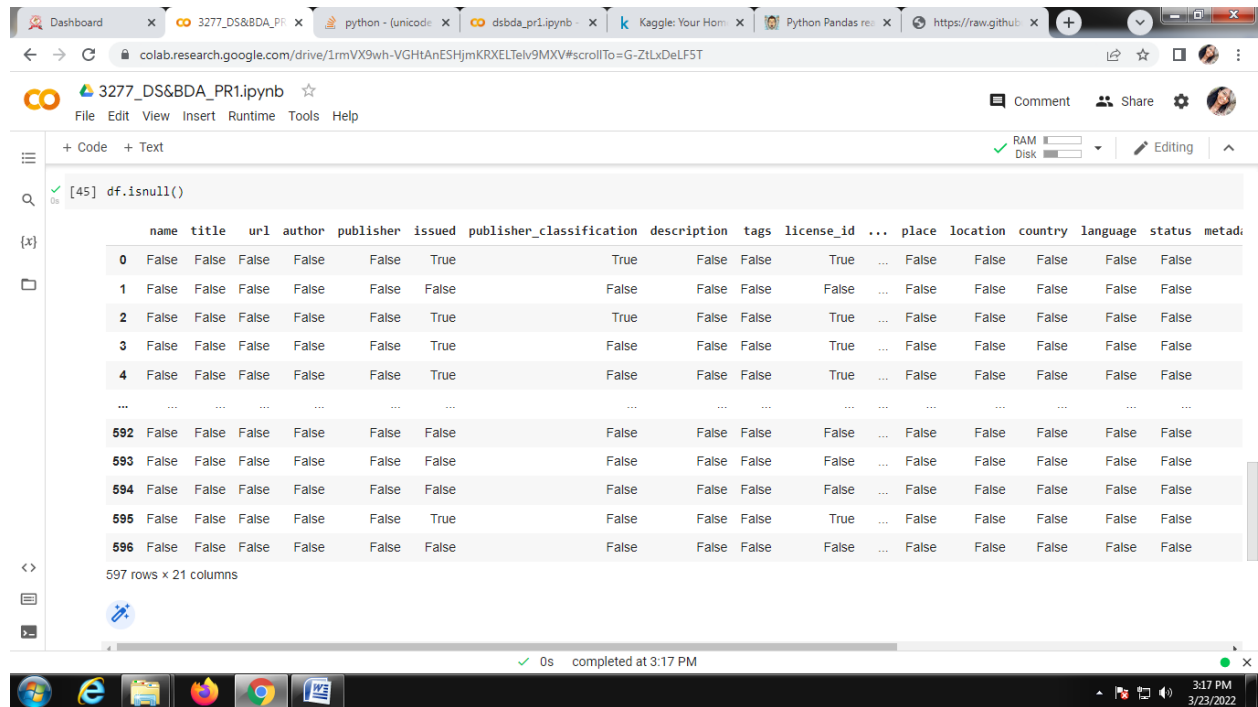
The output of the code is a Pandas DataFrame with 5 rows and 6 columns. The columns are: 'country', 'language', 'status', 'metadatatcreated', and 'generator'. The rows represent different portals, including 'Uzbekistan', 'transportation,local-government,nsw,au', 'agriculture,tn', 'palestine,ps', 'Ann Arbor, Michigan', 'Gaziantep,Turkey', 'Africa', 'Tarragona', 'Terrasa', 'Uzbekistan', 'Uzbekistan', 'New South Wales, Australia', 'Tunisia', and 'Palestine'.

	country	language	status	metadatatcreated	generator
0	US	en	active	2011-06-27T18:12:57.439Z	NaN
1	TR	tr	active	NaN	NaN
2	AF	en	active	2013-03-15T07:17:26.251Z	CKAN: 2.1.3
3	ES	ca es en	active	NaN	NaN
4	ES	es en	active	NaN	NaN

The notebook interface shows the code cell is executed, and the output is displayed. The status bar at the bottom indicates 'completed at 3:14 PM'.

4. Data Preprocessing: check for missing values in the data using pandas `isnull()`, `describe()` function to get some initial statistics. Provide variable descriptions. Types of variables etc. Check the dimensions of the data frame.

isnull() : to check the missing values



3277_DS&BDA_PR1.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

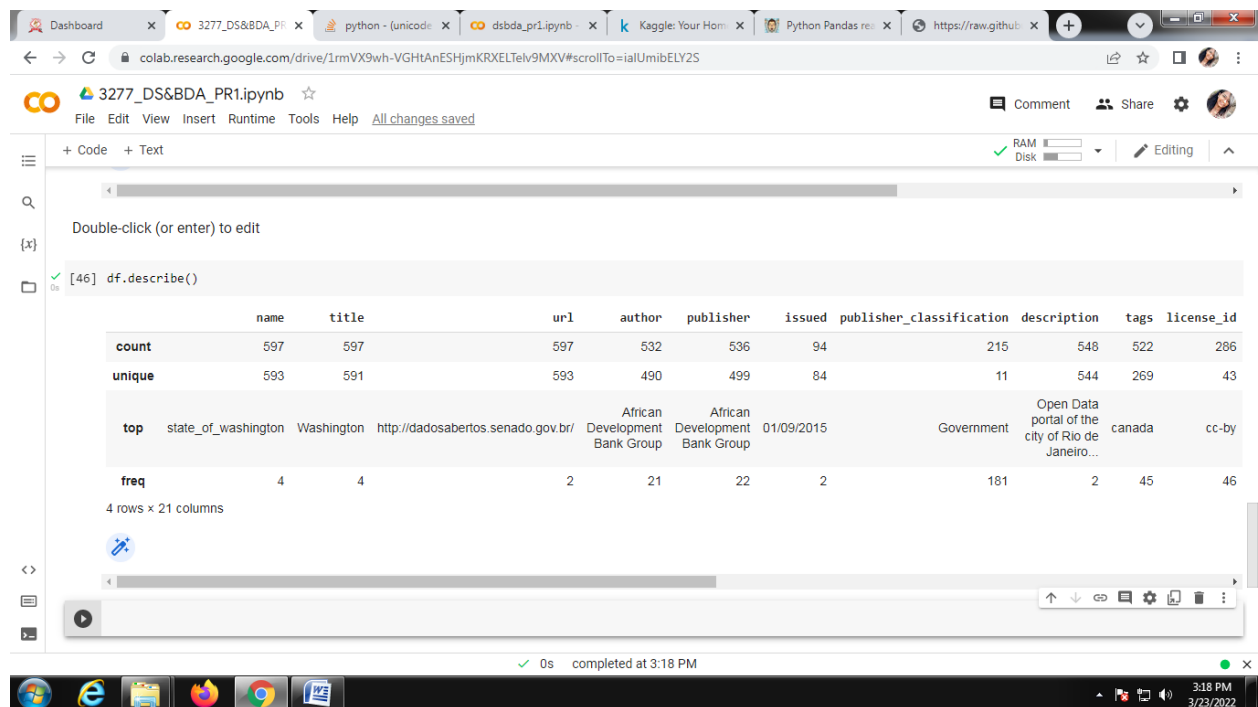
[45] df.isnull()

	name	title	url	author	publisher	issued	publisher_classification	description	tags	license_id	...	place	location	country	language	status	metad
0	False	False	False	False	False	True	True	False	False	True	...	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False
2	False	False	False	False	False	True	True	False	False	True	...	False	False	False	False	False	False
3	False	False	False	False	False	True	False	False	False	True	...	False	False	False	False	False	False
4	False	False	False	False	False	True	False	False	False	True	...	False	False	False	False	False	False
...
592	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False
593	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False
594	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False
595	False	False	False	False	False	True	False	False	False	True	...	False	False	False	False	False	False
596	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False

597 rows x 21 columns

0s completed at 3:17 PM

describe() : returns the statistical summary of dataframe or series.



3277_DS&BDA_PR1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

Double-click (or enter) to edit

[46] df.describe()

	name	title	url	author	publisher	issued	publisher_classification	description	tags	license_id
count	597	597	597	532	536	94	215	548	522	286
unique	593	591	593	490	499	84	11	544	269	43
top	state_of_washington	Washington	http://dadosabertos.senado.gov.br/	African Development Bank Group	African Development Bank Group	01/09/2015	Government	Open Data portal of the city of Rio de Janeiro...	canada	cc-by
freq	4	4	2	21	22	2	181	2	45	46

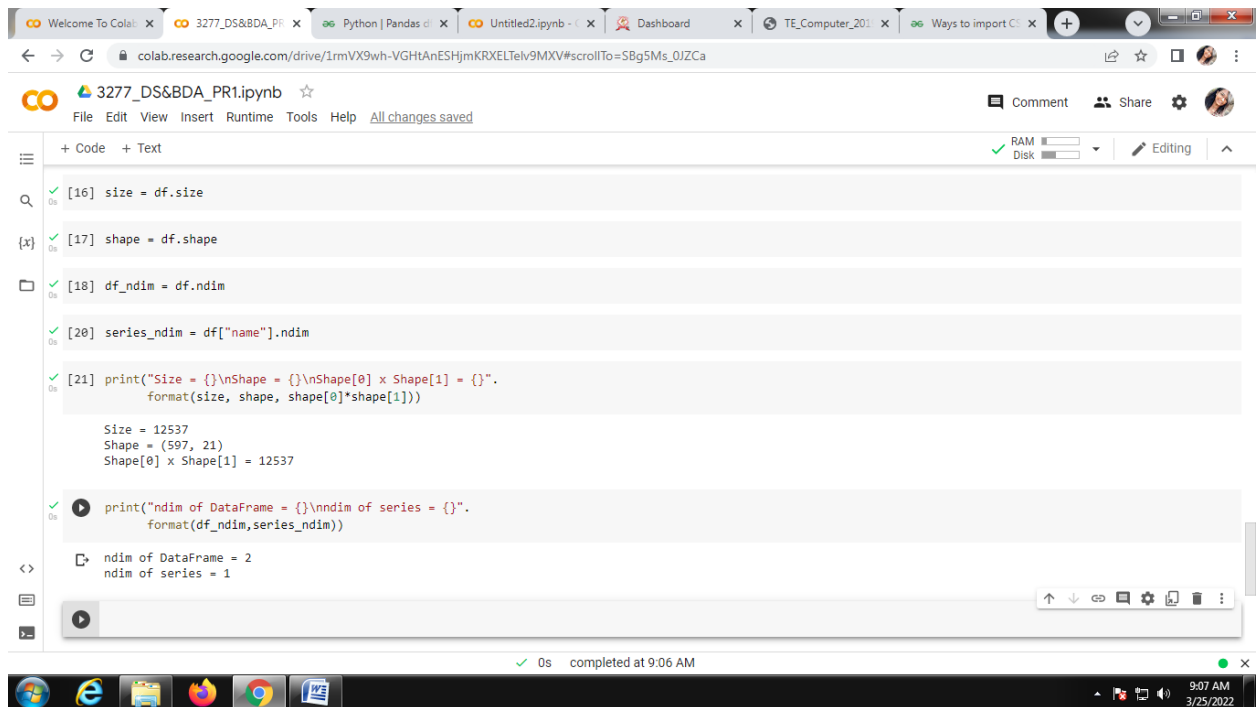
4 rows x 21 columns

0s completed at 3:18 PM

`size()` : count the number of element along given axis.

`shape()` : gives the number of elements in each dimension of an array.

`ndim()` : return the number of dimensions of an array.



The screenshot shows a Jupyter Notebook interface with the following code cells:

```
[16] size = df.size
```

```
[17] shape = df.shape
```

```
[18] df.ndim = df.ndim
```

```
[20] series_ndim = df["name"].ndim
```

```
[21] print("Size = {}\nShape = {}\nShape[0] x Shape[1] = {}".format(size, shape, shape[0]*shape[1]))
```

Size = 12537
Shape = (597, 21)
Shape[0] x Shape[1] = 12537

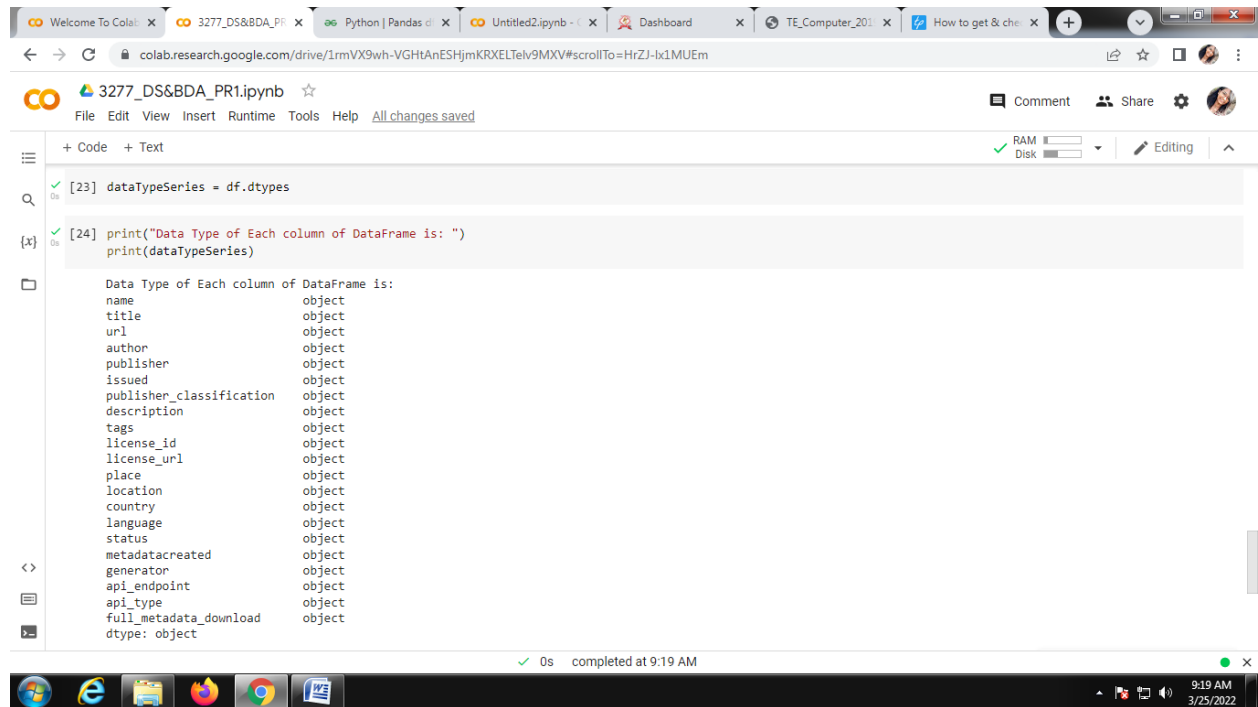
```
print("ndim of DataFrame = {}\nndim of series = {}".format(df.ndim, series_ndim))
```

ndim of DataFrame = 2
ndim of series = 1

The interface also shows a toolbar with options like + Code, + Text, RAM, Disk, and Editing. The bottom status bar indicates "0s completed at 9:06 AM" and the system clock shows "9:07 AM 3/25/2022".

5. Data Formatting and Data Normalization: Summarize the types of variables by checking the data types (i.e., character, numeric, integer, factor, and logical) of the variables in the data set. If variables are not in the correct data type, apply proper type conversions.

dtypes: to check the data types of columns in a DataFrame.



The screenshot shows a Jupyter Notebook interface with the following code and output:

```
[23] dataTypeSeries = df.dtypes
```

```
[24] print("Data Type of Each column of DataFrame is: ")
      print(dataTypeSeries)
```

Data Type of Each column of DataFrame is:

name	object
title	object
url	object
author	object
publisher	object
issued	object
publisher_classification	object
description	object
tags	object
license_id	object
license_url	object
place	object
location	object
country	object
language	object
status	object
metadata_created	object
generator	object
api_endpoint	object
api_type	object
full_metadata_download	object
dtype:	object

0s completed at 9:19 AM

6. Turn categorical variables into quantitative variables in Python.

get_dummies() : this method will return dummy variable columns.

concat() : to concatenate dummy columns into DataFrames

3277_DS&BDA_PR1.ipynb

```
[27] dummies = pd.get_dummies(df.author)

[29] merged = pd.concat([df, dummies], axis = 'columns')

[30] merged.drop(['author'], axis = 'columns')
```

	name	title	url	publisher	issued	publisher_classification	description	tags	license
0	a2gov_org	Ann Arbor, Michigan	http://www.a2gov.org/services/data/Pages/default.aspx	City of Ann Arbor	NaN	NaN	City of Ann Arbor's Open Data Catalog (USA)	ctic unitedstates	
1	acikveri-sahinbey-bel-tr	Açık Veri Portali - Test Yayını	http://acikveri.sahinbey.bel.tr/dataset	Sahinbey Belediyesi	31/01/2015	Government	The first official open data portal of Turkey	turkey national	Unk
2	africa_open_data	Africa Open Data	http://africaopendata.org/	Africa Open Data	NaN	NaN	Africa's largest central repository for Govern...	ckan africa	
3	ajuntament-de-tarragona	Open Data Tarragona	http://opendata.tarragona.cat/	Ajuntament de	NaN	Government	Open Data Tarragona	city spain	

0s completed at 9:43 AM

3277_DS&BDA_PR1.ipynb

```
print(merged)
```

```
0      a2gov_org
1  acikveri-sahinbey-bel-tr
2      africa_open_data
3  ajuntament-de-tarragona
4  ajuntament-de-terassa
..
592      stat-uz
593      data-gov-uz
594      transport-for-nsu
595      agridata-tn
596      open_data_ps

      title \
0      Ann Arbor, Michigan
1      Açık Veri Portali - Test Yayını
2      Africa Open Data
3      Open Data Tarragona
4      Open Data Terasa
..
592  The State Committee of the Republic of Uzbekis...
593      Open Data Portal of Uzbekistan
594      Transport for NSW Open Data Hub
595      Agriculture Portal in Tunisia
596      Palestine Open Data Portal

      url \
0      http://www.a2gov.org/services/data/Pages/default.aspx
```

0s completed at 9:43 AM