Comparative Presidential Executive Orders

(From President Truman to President Obama)

Emmanuel Olawuyi, Adote A, and Mohammed Shams

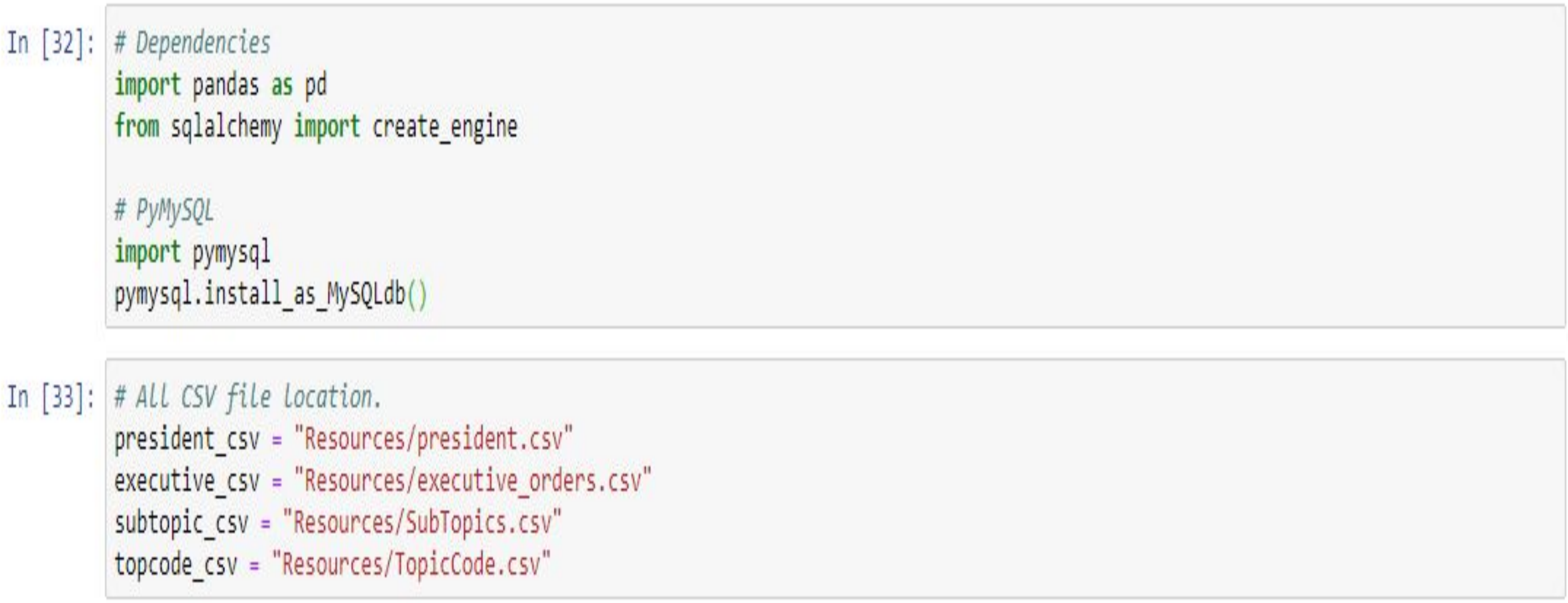
Mo

Introduction

- Dataset for the Project was derived from https://data.world/brandtcowan/executive-orders
- Datasets included csv files for Executive Orders, Topic of Orders, Subtopic of Orders, and Presidential Orders
- Data was Extracted, Transformed using Jupyter Notebook and loaded into MySQL for analysis.

Mo

Extractions of Data



In [32]:

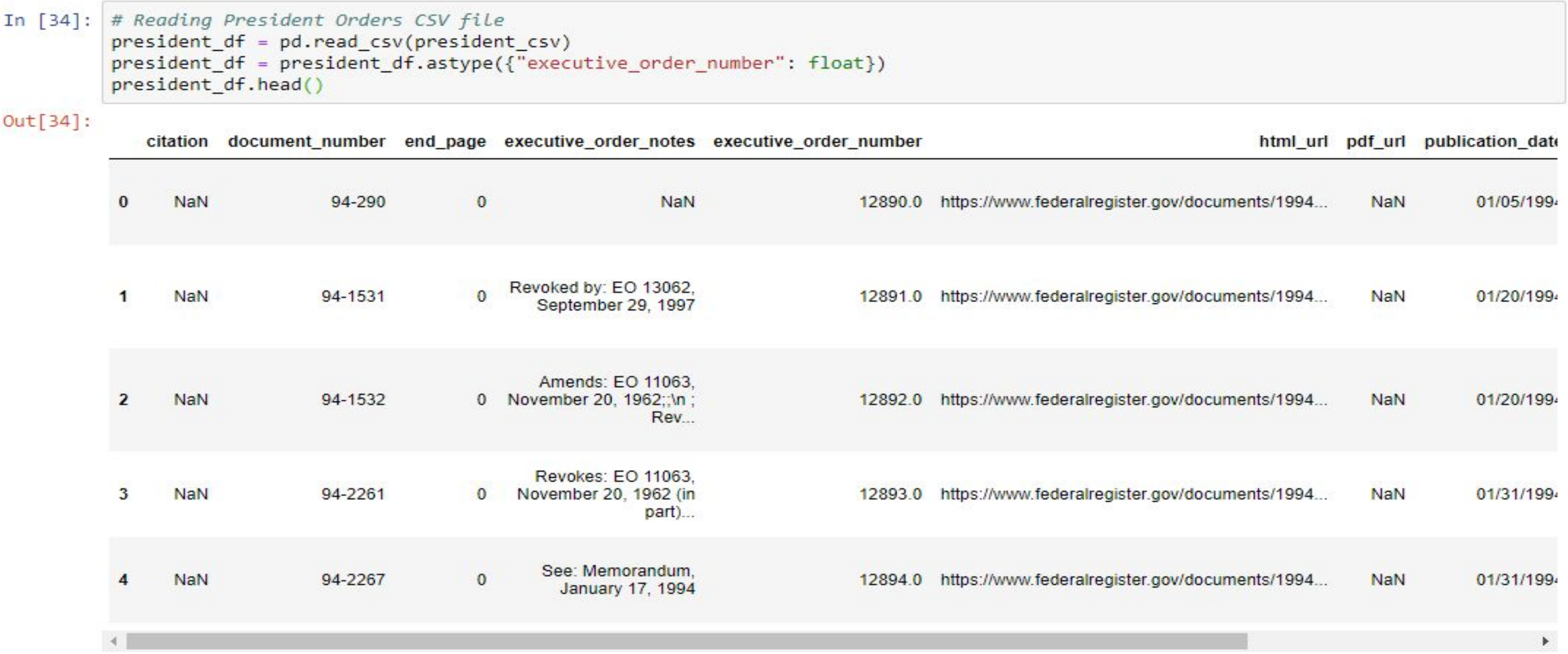
```
# Dependencies
import pandas as pd
from sqlalchemy import create_engine

# PyMySQL
import pymysql
pymysql.install_as_MySQLdb()
```

In [33]:

```
# All CSV file location.
president_csv = "Resources/president.csv"
executive_csv = "Resources/executive_orders.csv"
subtopic_csv = "Resources/SubTopics.csv"
topcode_csv = "Resources/TopicCode.csv"
```

Mo

Extractions of Data



In [34]:

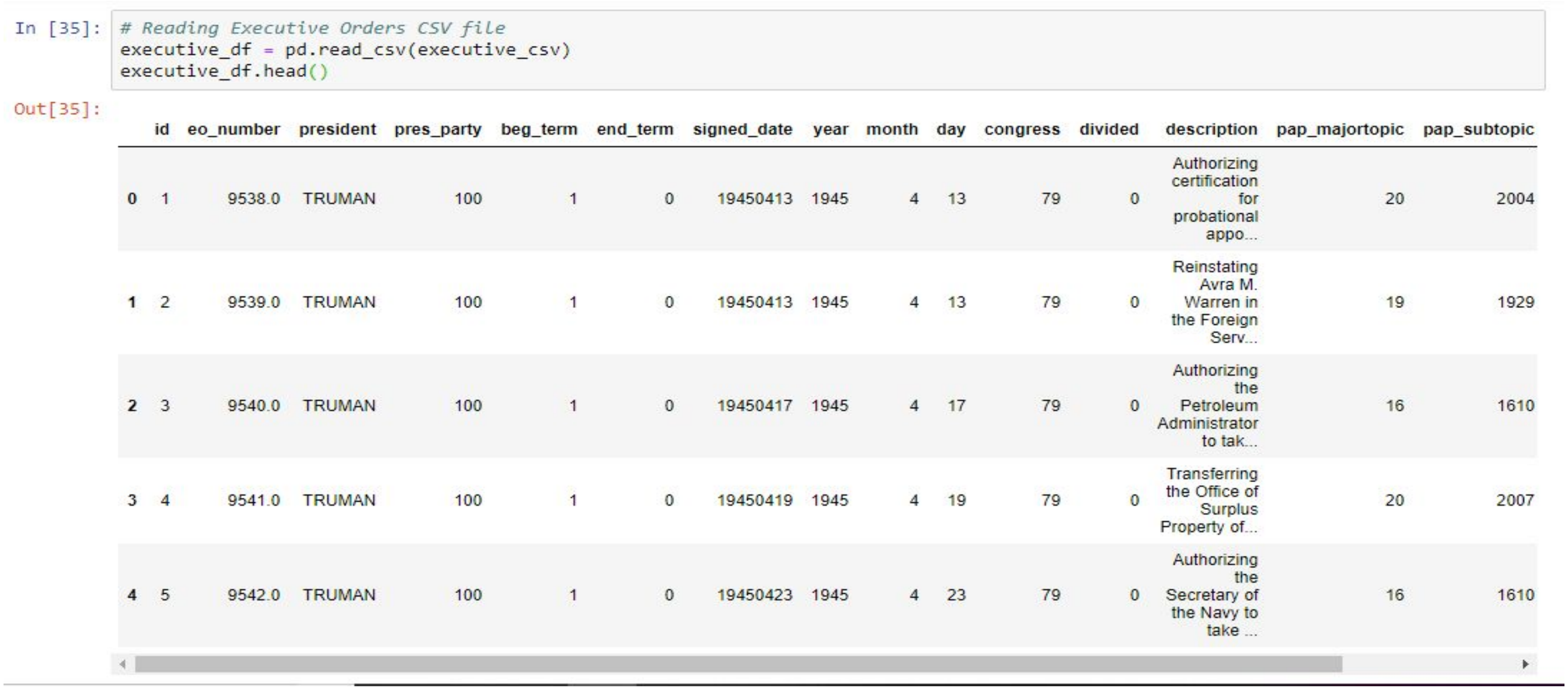
```
# Reading President Orders CSV file
president_df = pd.read_csv(president_csv)
president_df = president_df.astype({"executive_order_number": float})
president_df.head()
```

Out[34]:

	citation	document_number	end_page	executive_order_notes	executive_order_number	html_url	pdf_url	publication_dat
0	NaN	94-290	0	NaN	12890.0	https://www.federalregister.gov/documents/1994...	NaN	01/05/199
1	NaN	94-1531	0	Revoked by: EO 13062, September 29, 1997	12891.0	https://www.federalregister.gov/documents/1994...	NaN	01/20/199
2	NaN	94-1532	0	Amends: EO 11063, November 20, 1962;;\n ; Rev...	12892.0	https://www.federalregister.gov/documents/1994...	NaN	01/20/199
3	NaN	94-2261	0	Revokes: EO 11063, November 20, 1962 (in part)...	12893.0	https://www.federalregister.gov/documents/1994...	NaN	01/31/199
4	NaN	94-2267	0	See: Memorandum, January 17, 1994	12894.0	https://www.federalregister.gov/documents/1994...	NaN	01/31/199

Mo

Extractions of Data



In [35]:

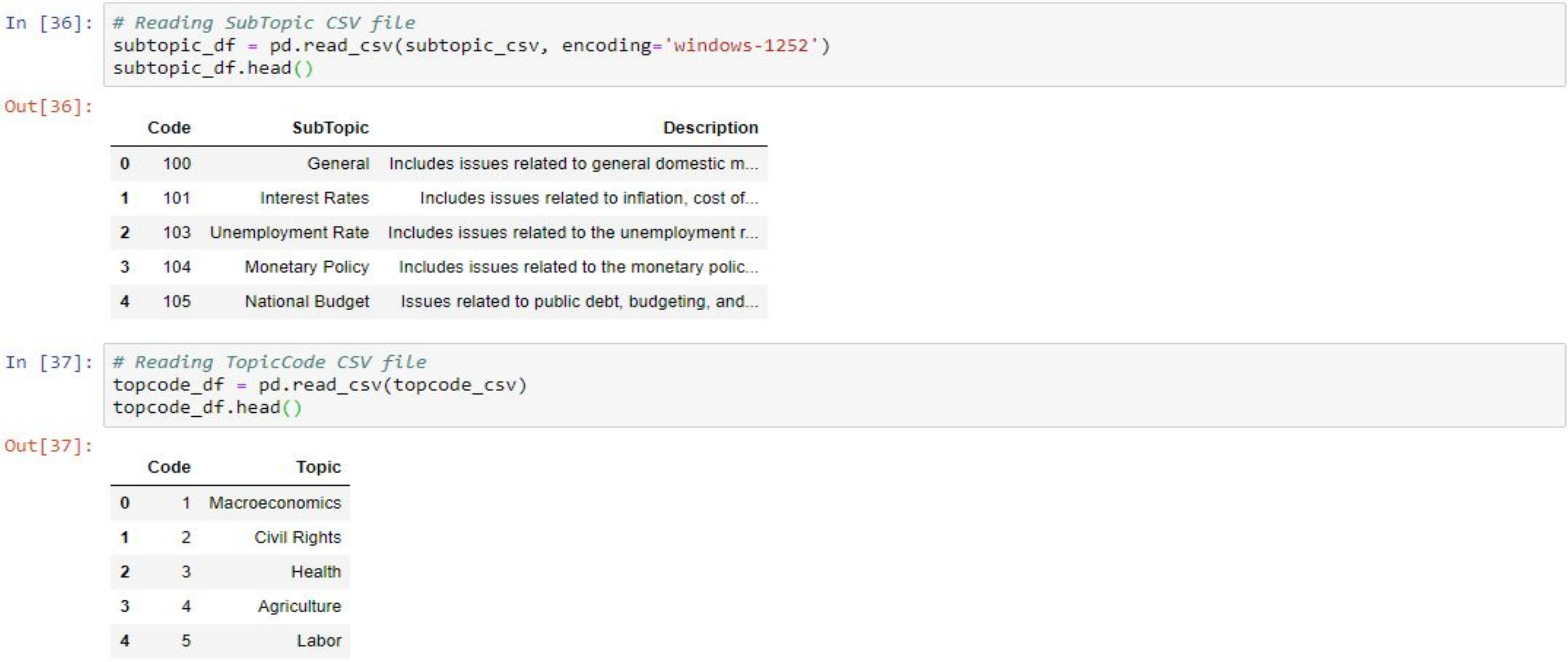
```
# Reading Executive Orders CSV file
executive_df = pd.read_csv(executive_csv)
executive_df.head()
```

Out[35]:

	id	eo_number	president	pres_party	beg_term	end_term	signed_date	year	month	day	congress	divided	description	pap_majortopic	pap_subtopic
0	1	9538.0	TRUMAN	100	1	0	19450413	1945	4	13	79	0	Authorizing certification for probational appo...	20	2004
1	2	9539.0	TRUMAN	100	1	0	19450413	1945	4	13	79	0	Reinstating Avra M. Warren in the Foreign Serv...	19	1929
2	3	9540.0	TRUMAN	100	1	0	19450417	1945	4	17	79	0	Authorizing the Petroleum Administrator to tak...	16	1610
3	4	9541.0	TRUMAN	100	1	0	19450419	1945	4	19	79	0	Transferring the Office of Surplus Property of...	20	2007
4	5	9542.0	TRUMAN	100	1	0	19450423	1945	4	23	79	0	Authorizing the Secretary of the Navy to take ...	16	1610

Mo

Extractions of Data



In [36]:

```
# Reading SubTopic CSV file
subtopic_df = pd.read_csv(subtopic_csv, encoding='windows-1252')
subtopic_df.head()
```

Out[36]:

	Code	SubTopic	Description
0	100	General	Includes issues related to general domestic m...
1	101	Interest Rates	Includes issues related to inflation, cost of...
2	103	Unemployment Rate	Includes issues related to the unemployment r...
3	104	Monetary Policy	Includes issues related to the monetary polic...
4	105	National Budget	Issues related to public debt, budgeting, and...

In [37]:

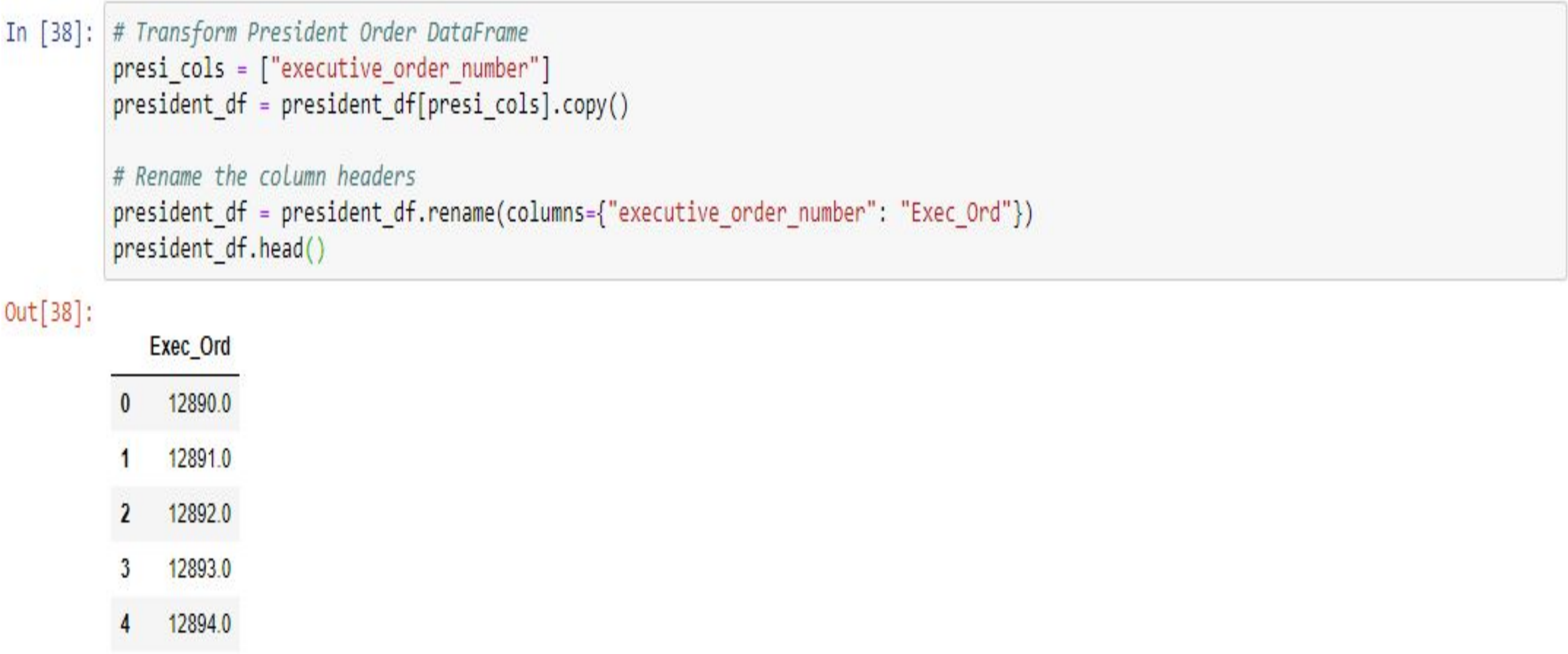
```
# Reading TopicCode CSV file
topcode_df = pd.read_csv(topcode_csv)
topcode_df.head()
```

Out[37]:

	Code	Topic
0	1	Macroeconomics
1	2	Civil Rights
2	3	Health
3	4	Agriculture
4	5	Labor

Mo

Transforming Data

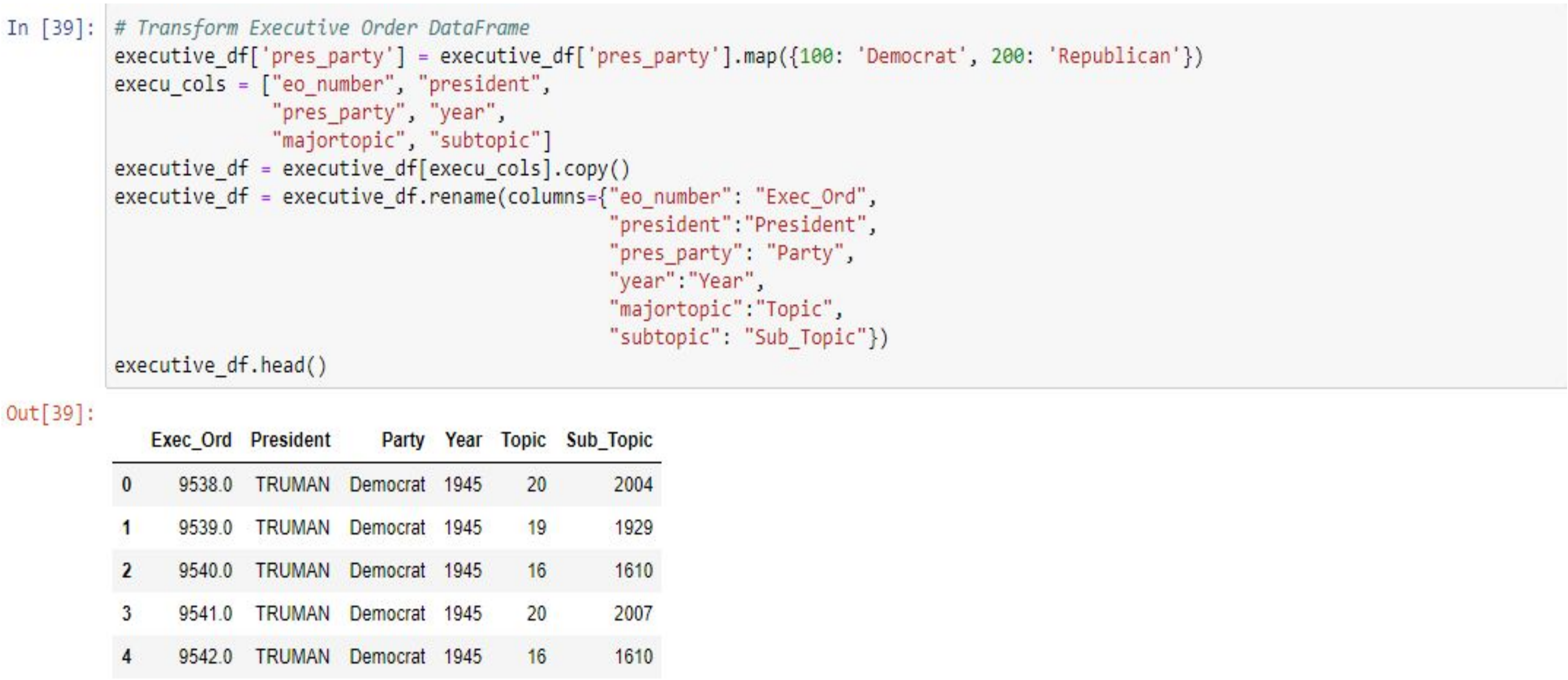


```
In [38]: # Transform President Order DataFrame
         presi_cols = ["executive_order_number"]
         president_df = president_df[presi_cols].copy()

         # Rename the column headers
         president_df = president_df.rename(columns={"executive_order_number": "Exec_Ord"})
         president_df.head()
```

```
Out[38]:
     Exec_Ord
  0   12890.0
  1   12891.0
  2   12892.0
  3   12893.0
  4   12894.0
```

Emmanuel

Transforming Data



In [39]:

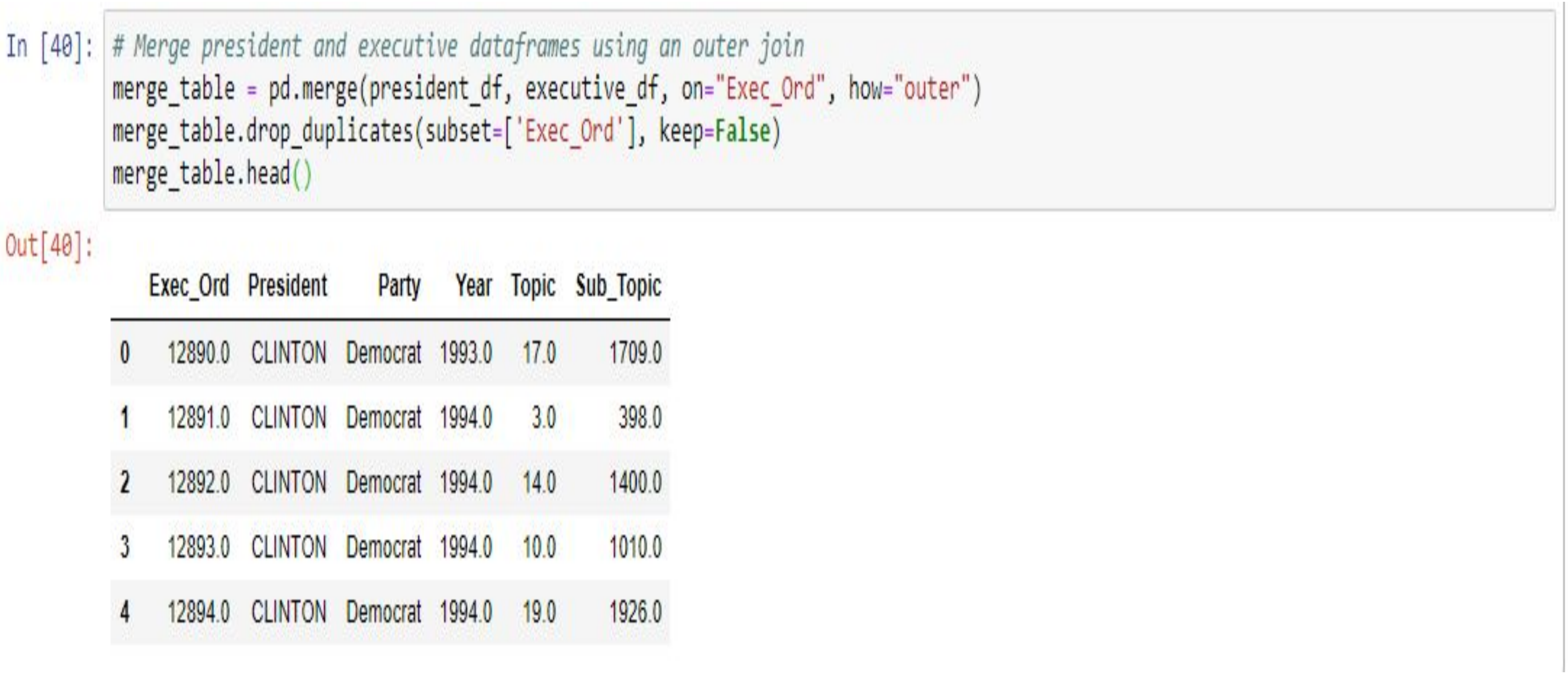
```
# Transform Executive Order DataFrame
executive_df['pres_party'] = executive_df['pres_party'].map({100: 'Democrat', 200: 'Republican'})
execu_cols = ["eo_number", "president",
              "pres_party", "year",
              "majortopic", "subtopic"]
executive_df = executive_df[execu_cols].copy()
executive_df = executive_df.rename(columns={"eo_number": "Exec_Ord",
                                            "president":"President",
                                            "pres_party": "Party",
                                            "year":"Year",
                                            "majortopic":"Topic",
                                            "subtopic": "Sub_Topic"})

executive_df.head()
```

Out[39]:

	Exec_Ord	President	Party	Year	Topic	Sub_Topic
0	9538.0	TRUMAN	Democrat	1945	20	2004
1	9539.0	TRUMAN	Democrat	1945	19	1929
2	9540.0	TRUMAN	Democrat	1945	16	1610
3	9541.0	TRUMAN	Democrat	1945	20	2007
4	9542.0	TRUMAN	Democrat	1945	16	1610

Emmanuel

Transforming Data

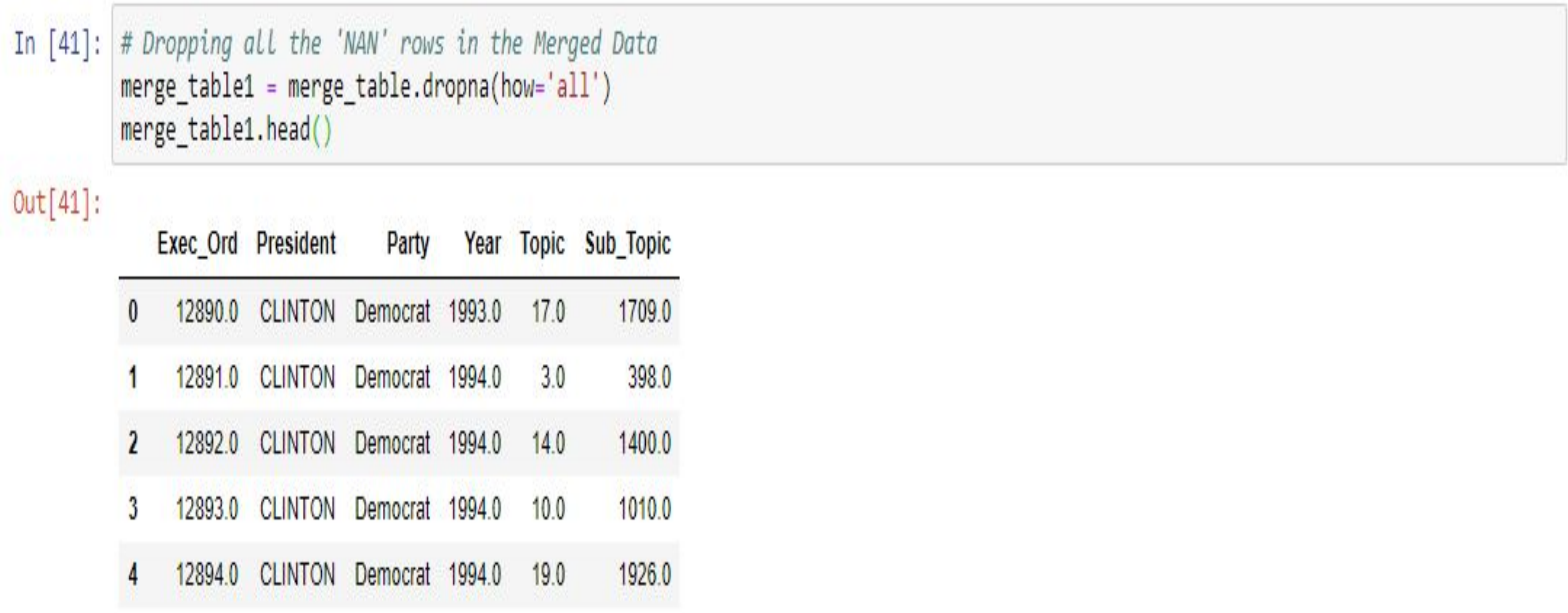


```
In [40]: # Merge president and executive dataframes using an outer join
         merge_table = pd.merge(president_df, executive_df, on="Exec_Ord", how="outer")
         merge_table.drop_duplicates(subset=['Exec_Ord'], keep=False)
         merge_table.head()
```

```
Out[40]:
     Exec_Ord  President     Party    Year  Topic  Sub_Topic
  0   12890.0    CLINTON  Democrat  1993.0   17.0     1709.0
  1   12891.0    CLINTON  Democrat  1994.0    3.0      398.0
  2   12892.0    CLINTON  Democrat  1994.0   14.0     1400.0
  3   12893.0    CLINTON  Democrat  1994.0   10.0     1010.0
  4   12894.0    CLINTON  Democrat  1994.0   19.0     1926.0
```

Emmanuel

Transforming Data



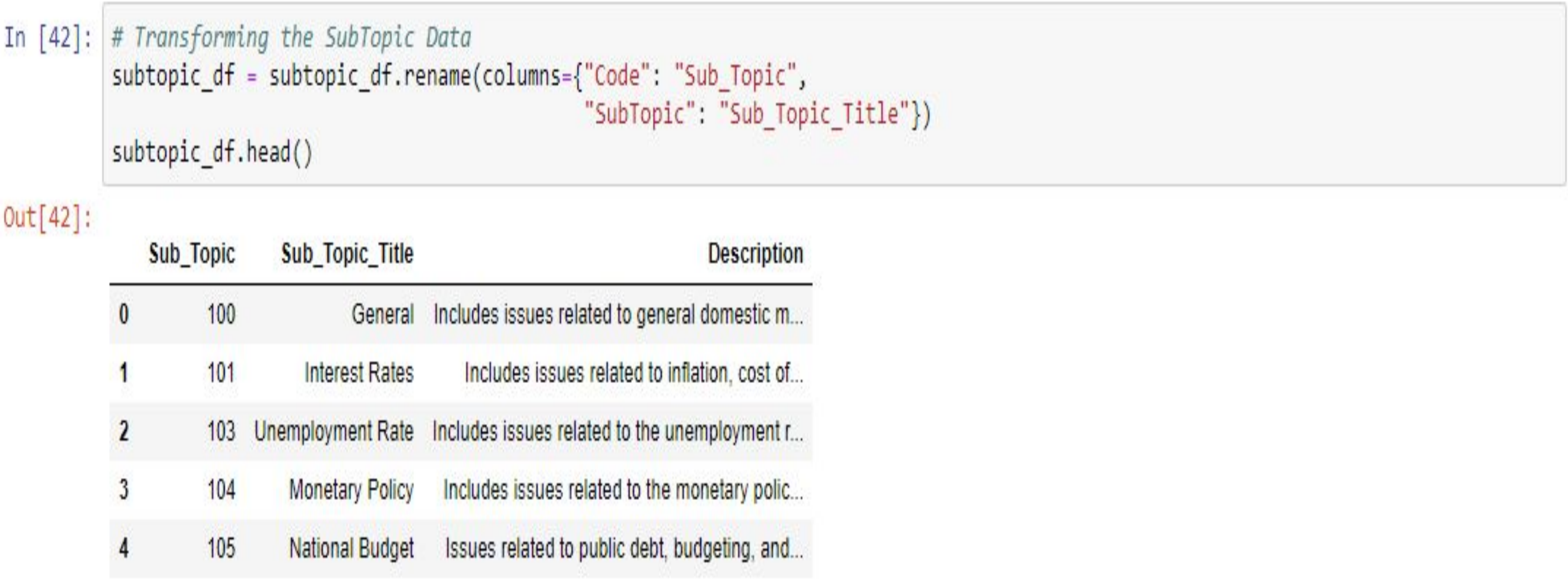
```
In [41]: # Dropping all the 'NAN' rows in the Merged Data
         merge_table1 = merge_table.dropna(how='all')
         merge_table1.head()
```

Out[41]:

	Exec_Ord	President	Party	Year	Topic	Sub_Topic
0	12890.0	CLINTON	Democrat	1993.0	17.0	1709.0
1	12891.0	CLINTON	Democrat	1994.0	3.0	398.0
2	12892.0	CLINTON	Democrat	1994.0	14.0	1400.0
3	12893.0	CLINTON	Democrat	1994.0	10.0	1010.0
4	12894.0	CLINTON	Democrat	1994.0	19.0	1926.0

Emmanuel

Transforming Data



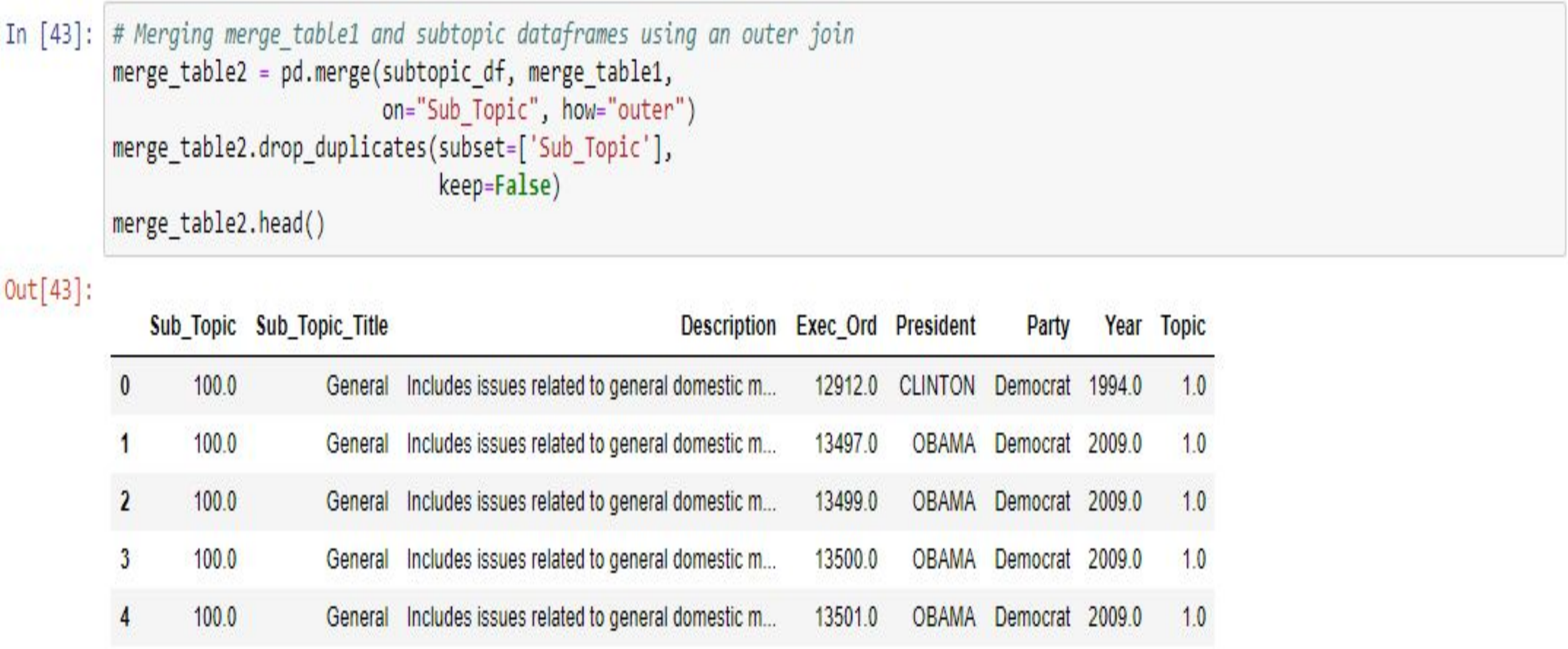
In [42]:
```
# Transforming the SubTopic Data
subtopic_df = subtopic_df.rename(columns={"Code": "Sub_Topic",
                                          "SubTopic": "Sub_Topic_Title"})

subtopic_df.head()
```

Out[42]:

	Sub_Topic	Sub_Topic_Title	Description
0	100	General	Includes issues related to general domestic m...
1	101	Interest Rates	Includes issues related to inflation, cost of...
2	103	Unemployment Rate	Includes issues related to the unemployment r...
3	104	Monetary Policy	Includes issues related to the monetary polic...
4	105	National Budget	Issues related to public debt, budgeting, and...

Emmanuel

Transforming Data



In [43]:

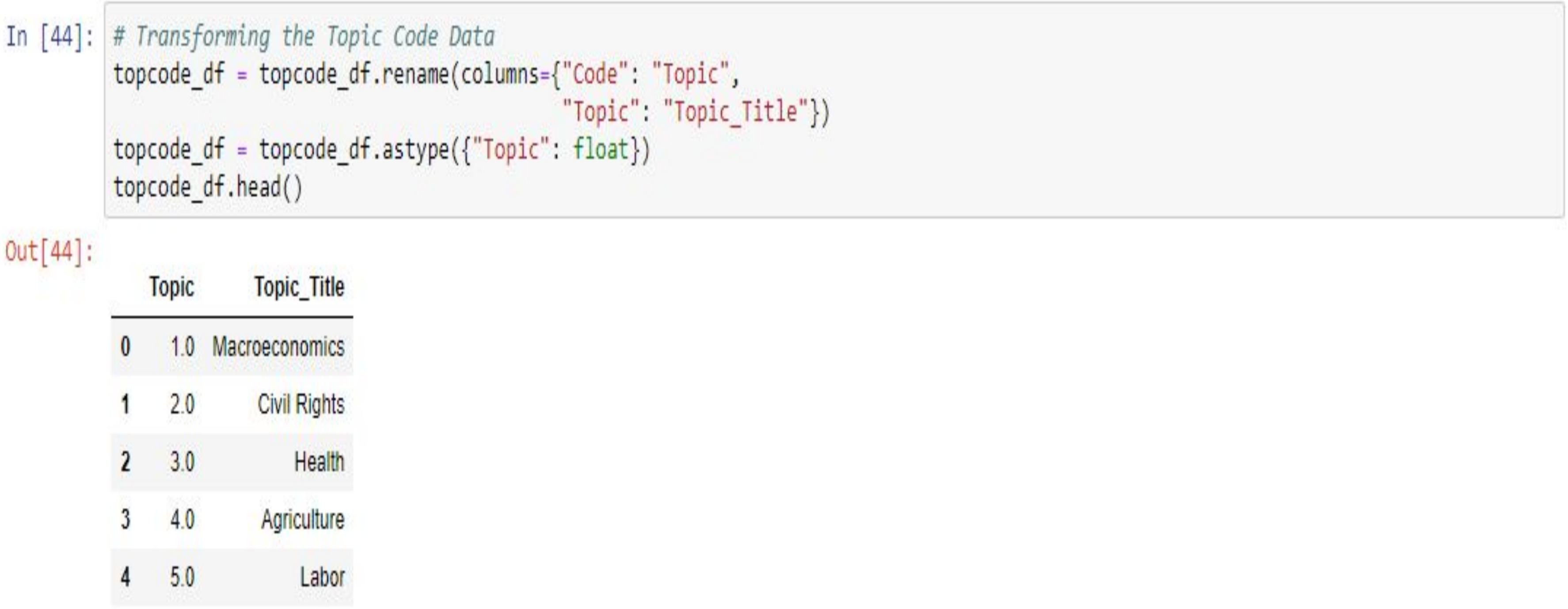
```
# Merging merge_table1 and subtopic dataframes using an outer join
merge_table2 = pd.merge(subtopic_df, merge_table1,
                        on="Sub_Topic", how="outer")
merge_table2.drop_duplicates(subset=['Sub_Topic'],
                             keep=False)
merge_table2.head()
```

Out[43]:

	Sub_Topic	Sub_Topic_Title	Description	Exec_Ord	President	Party	Year	Topic
0	100.0	General	Includes issues related to general domestic m...	12912.0	CLINTON	Democrat	1994.0	1.0
1	100.0	General	Includes issues related to general domestic m...	13497.0	OBAMA	Democrat	2009.0	1.0
2	100.0	General	Includes issues related to general domestic m...	13499.0	OBAMA	Democrat	2009.0	1.0
3	100.0	General	Includes issues related to general domestic m...	13500.0	OBAMA	Democrat	2009.0	1.0
4	100.0	General	Includes issues related to general domestic m...	13501.0	OBAMA	Democrat	2009.0	1.0

Emmanuel

Transforming Data

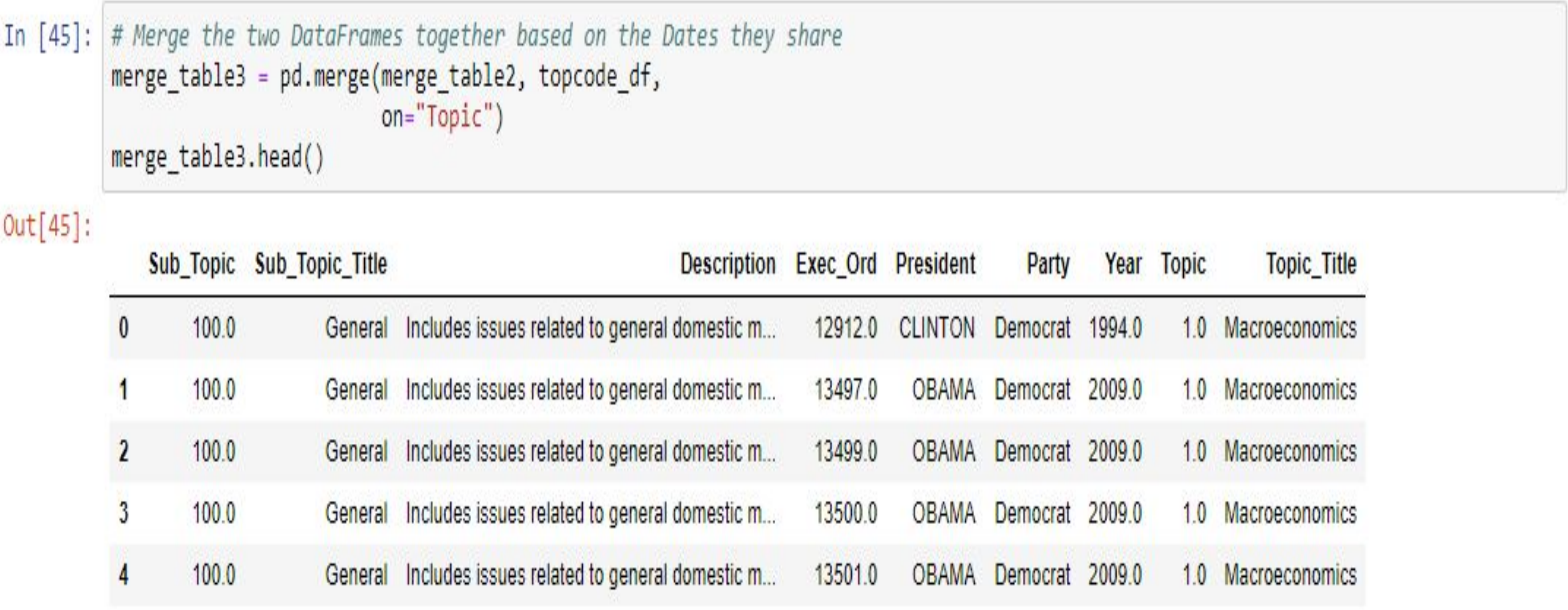


```
In [44]: # Transforming the Topic Code Data
         topcode_df = topcode_df.rename(columns={"Code": "Topic",
                                                 "Topic": "Topic_Title"})
         topcode_df = topcode_df.astype({"Topic": float})
         topcode_df.head()
```

```
Out[44]:
     Topic     Topic_Title
  0    1.0  Macroeconomics
  1    2.0    Civil Rights
  2    3.0          Health
  3    4.0     Agriculture
  4    5.0           Labor
```

Emmanuel

Transforming Data



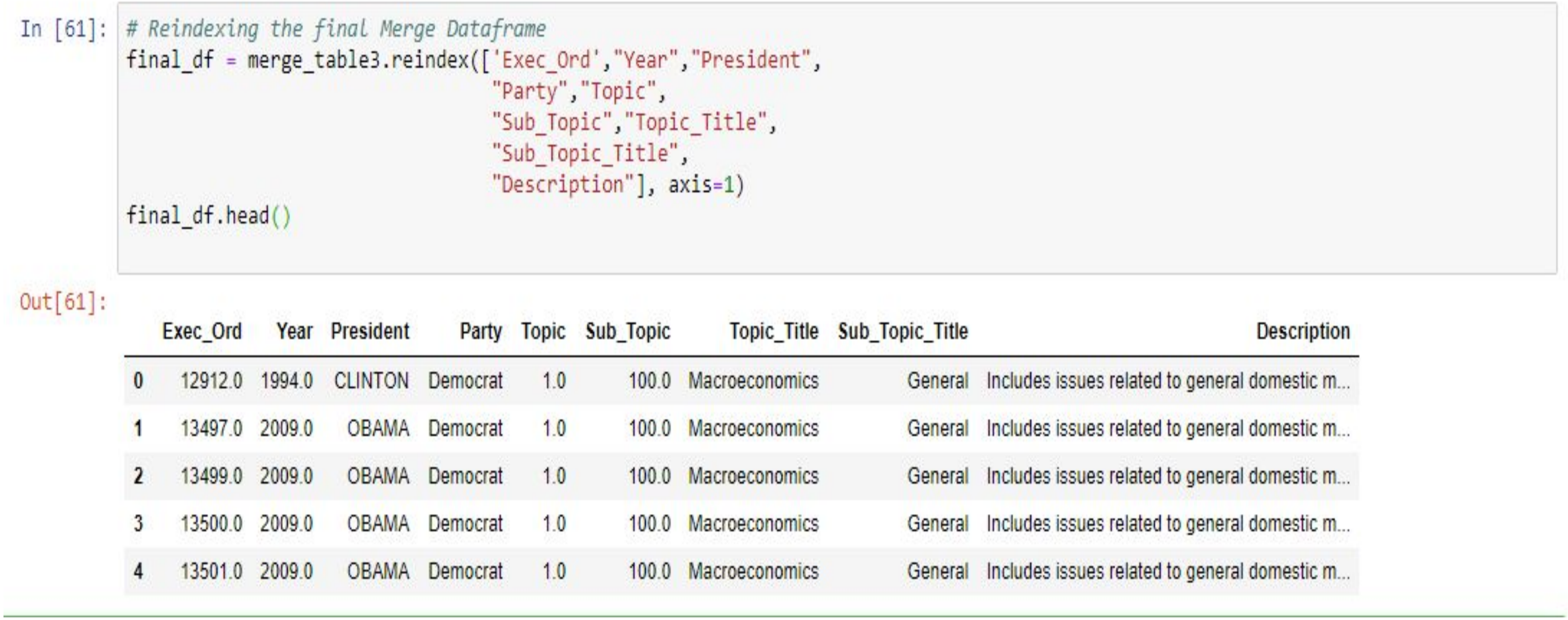
```
In [45]: # Merge the two DataFrames together based on the Dates they share
         merge_table3 = pd.merge(merge_table2, topcode_df,
                                 on="Topic")
         merge_table3.head()
```

Out[45]:

	Sub_Topic	Sub_Topic_Title	Description	Exec_Ord	President	Party	Year	Topic	Topic_Title
0	100.0	General	Includes issues related to general domestic m...	12912.0	CLINTON	Democrat	1994.0	1.0	Macroeconomics
1	100.0	General	Includes issues related to general domestic m...	13497.0	OBAMA	Democrat	2009.0	1.0	Macroeconomics
2	100.0	General	Includes issues related to general domestic m...	13499.0	OBAMA	Democrat	2009.0	1.0	Macroeconomics
3	100.0	General	Includes issues related to general domestic m...	13500.0	OBAMA	Democrat	2009.0	1.0	Macroeconomics
4	100.0	General	Includes issues related to general domestic m...	13501.0	OBAMA	Democrat	2009.0	1.0	Macroeconomics

Emmanuel

Transforming Data



In [61]:

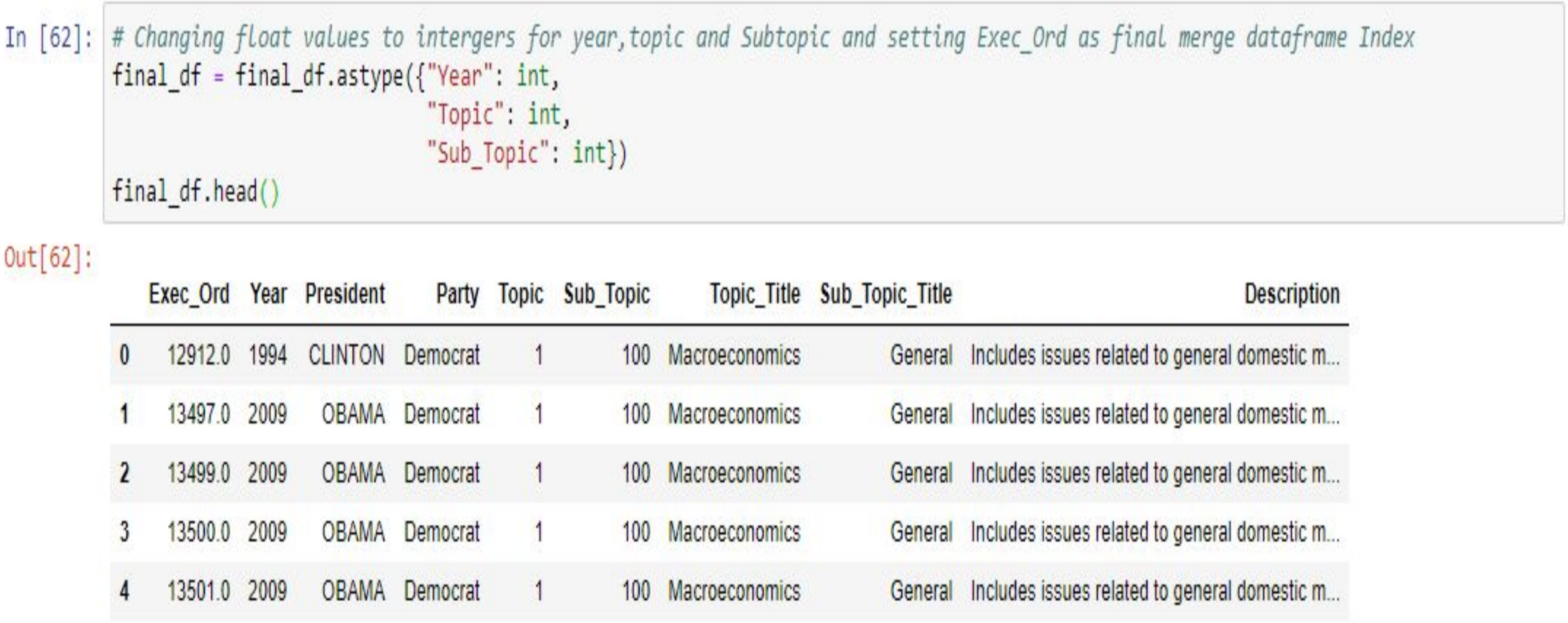
```
# Reindexing the final Merge Dataframe
final_df = merge_table3.reindex(['Exec_Ord',"Year","President",
                                 "Party","Topic",
                                 "Sub_Topic","Topic_Title",
                                 "Sub_Topic_Title",
                                 "Description"], axis=1)
final_df.head()
```

Out[61]:

	Exec_Ord	Year	President	Party	Topic	Sub_Topic	Topic_Title	Sub_Topic_Title	Description
0	12912.0	1994.0	CLINTON	Democrat	1.0	100.0	Macroeconomics	General	Includes issues related to general domestic m...
1	13497.0	2009.0	OBAMA	Democrat	1.0	100.0	Macroeconomics	General	Includes issues related to general domestic m...
2	13499.0	2009.0	OBAMA	Democrat	1.0	100.0	Macroeconomics	General	Includes issues related to general domestic m...
3	13500.0	2009.0	OBAMA	Democrat	1.0	100.0	Macroeconomics	General	Includes issues related to general domestic m...
4	13501.0	2009.0	OBAMA	Democrat	1.0	100.0	Macroeconomics	General	Includes issues related to general domestic m...

Emmanuel

Transforming Data



In [62]:

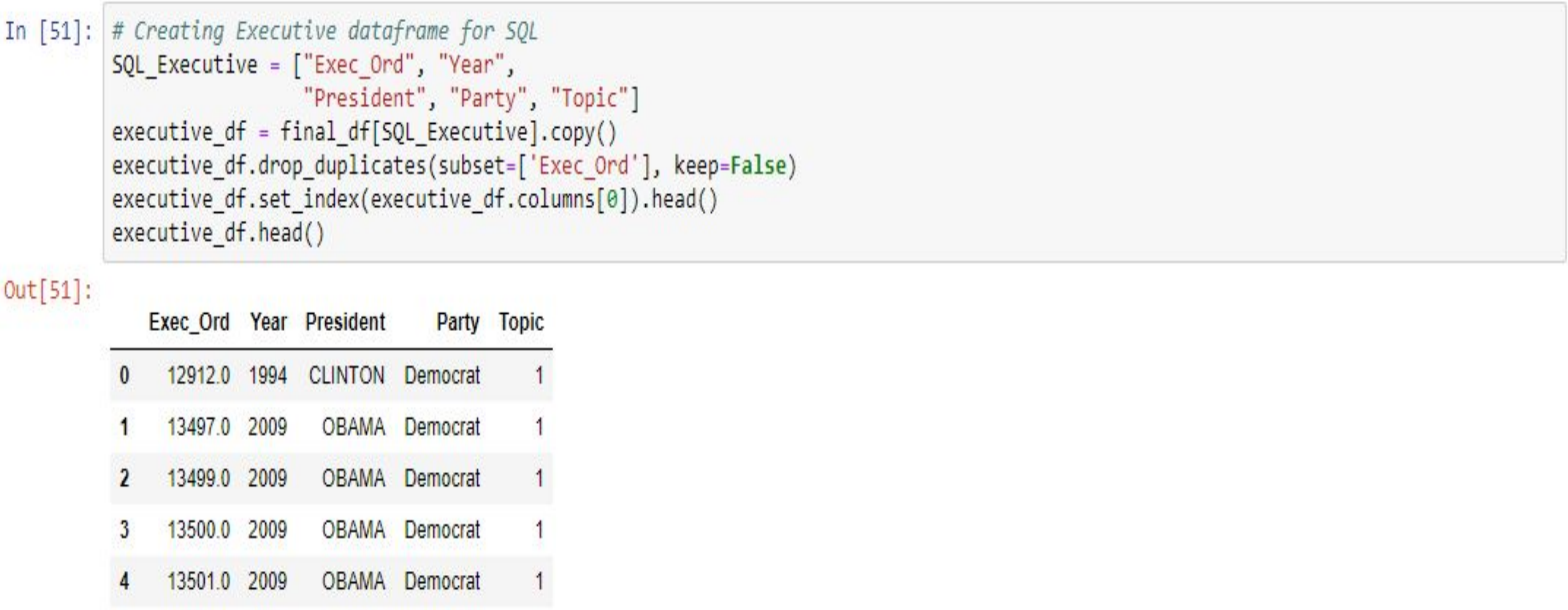
```
# Changing float values to intergers for year,topic and Subtopic and setting Exec_Ord as final merge dataframe Index
final_df = final_df.astype({"Year": int,
                            "Topic": int,
                            "Sub_Topic": int})
final_df.head()
```

Out[62]:

	Exec_Ord	Year	President	Party	Topic	Sub_Topic	Topic_Title	Sub_Topic_Title	Description
0	12912.0	1994	CLINTON	Democrat	1	100	Macroeconomics	General	Includes issues related to general domestic m...
1	13497.0	2009	OBAMA	Democrat	1	100	Macroeconomics	General	Includes issues related to general domestic m...
2	13499.0	2009	OBAMA	Democrat	1	100	Macroeconomics	General	Includes issues related to general domestic m...
3	13500.0	2009	OBAMA	Democrat	1	100	Macroeconomics	General	Includes issues related to general domestic m...
4	13501.0	2009	OBAMA	Democrat	1	100	Macroeconomics	General	Includes issues related to general domestic m...

Emmanuel

Transforming Data

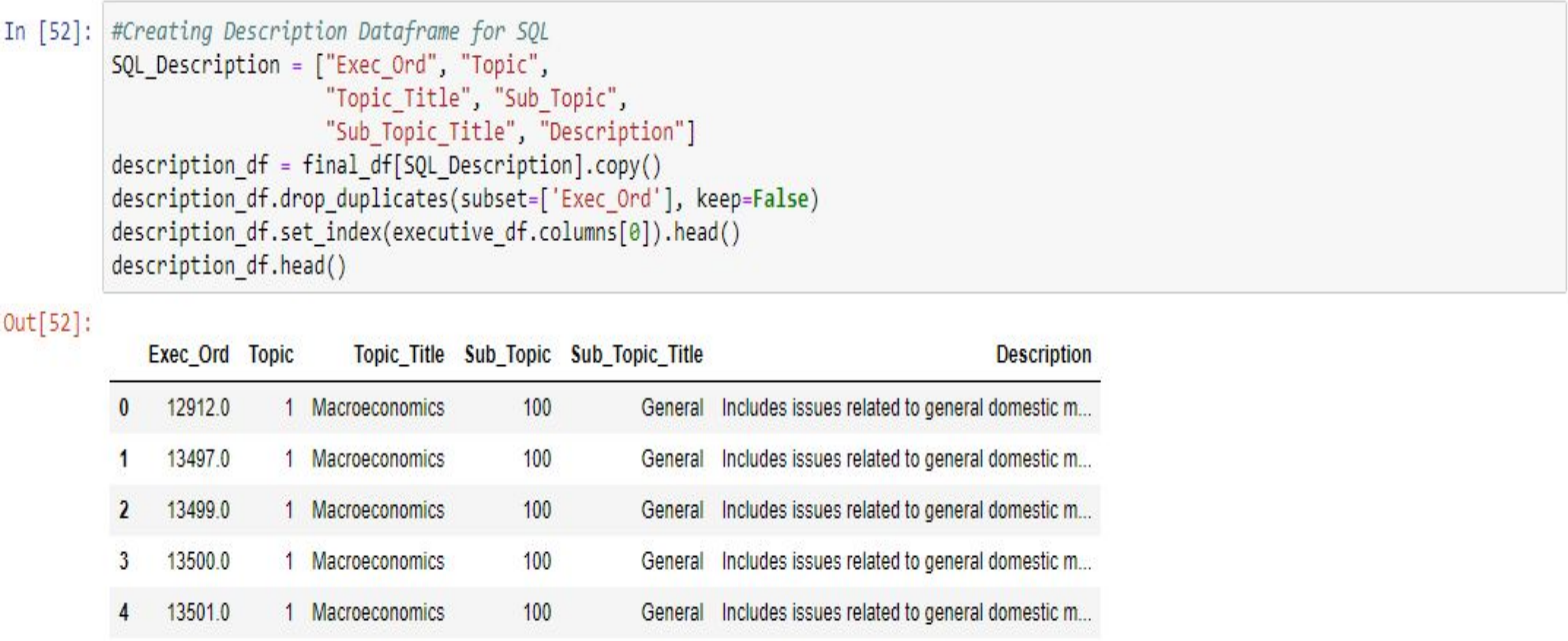


```
In [51]: # Creating Executive dataframe for SQL
         SQL_Executive = ["Exec_Ord", "Year",
                          "President", "Party", "Topic"]
         executive_df = final_df[SQL_Executive].copy()
         executive_df.drop_duplicates(subset=['Exec_Ord'], keep=False)
         executive_df.set_index(executive_df.columns[0]).head()
         executive_df.head()
```

```
Out[51]:
      Exec_Ord  Year  President     Party  Topic
   0   12912.0  1994    CLINTON  Democrat      1
   1   13497.0  2009      OBAMA  Democrat      1
   2   13499.0  2009      OBAMA  Democrat      1
   3   13500.0  2009      OBAMA  Democrat      1
   4   13501.0  2009      OBAMA  Democrat      1
```

Adote

Transforming Data



In [52]:

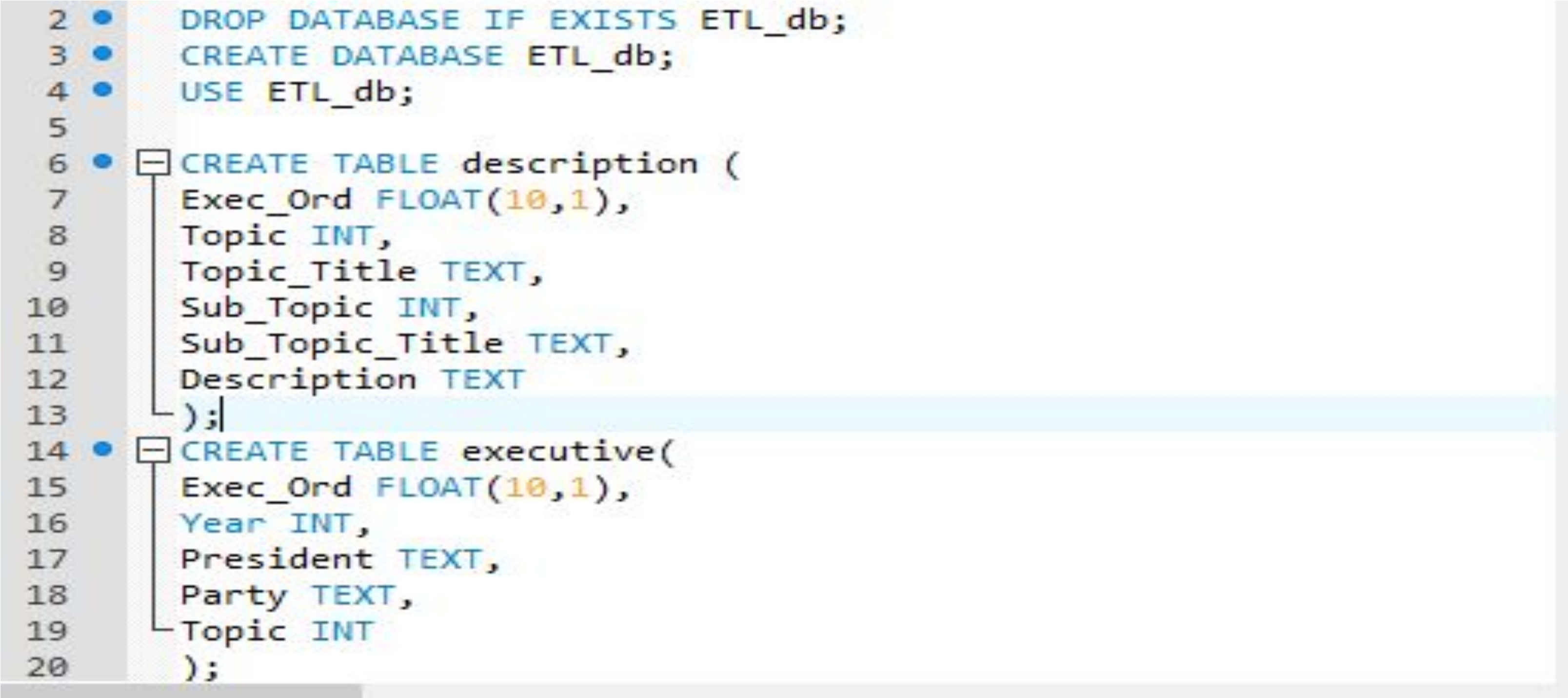
```
#Creating Description Dataframe for SQL
SQL_Description = ["Exec_Ord", "Topic",
                   "Topic_Title", "Sub_Topic",
                   "Sub_Topic_Title", "Description"]
description_df = final_df[SQL_Description].copy()
description_df.drop_duplicates(subset=['Exec_Ord'], keep=False)
description_df.set_index(executive_df.columns[0]).head()
description_df.head()
```

Out[52]:

	Exec_Ord	Topic	Topic_Title	Sub_Topic	Sub_Topic_Title	Description
0	12912.0	1	Macroeconomics	100	General	Includes issues related to general domestic m...
1	13497.0	1	Macroeconomics	100	General	Includes issues related to general domestic m...
2	13499.0	1	Macroeconomics	100	General	Includes issues related to general domestic m...
3	13500.0	1	Macroeconomics	100	General	Includes issues related to general domestic m...
4	13501.0	1	Macroeconomics	100	General	Includes issues related to general domestic m...

Adote

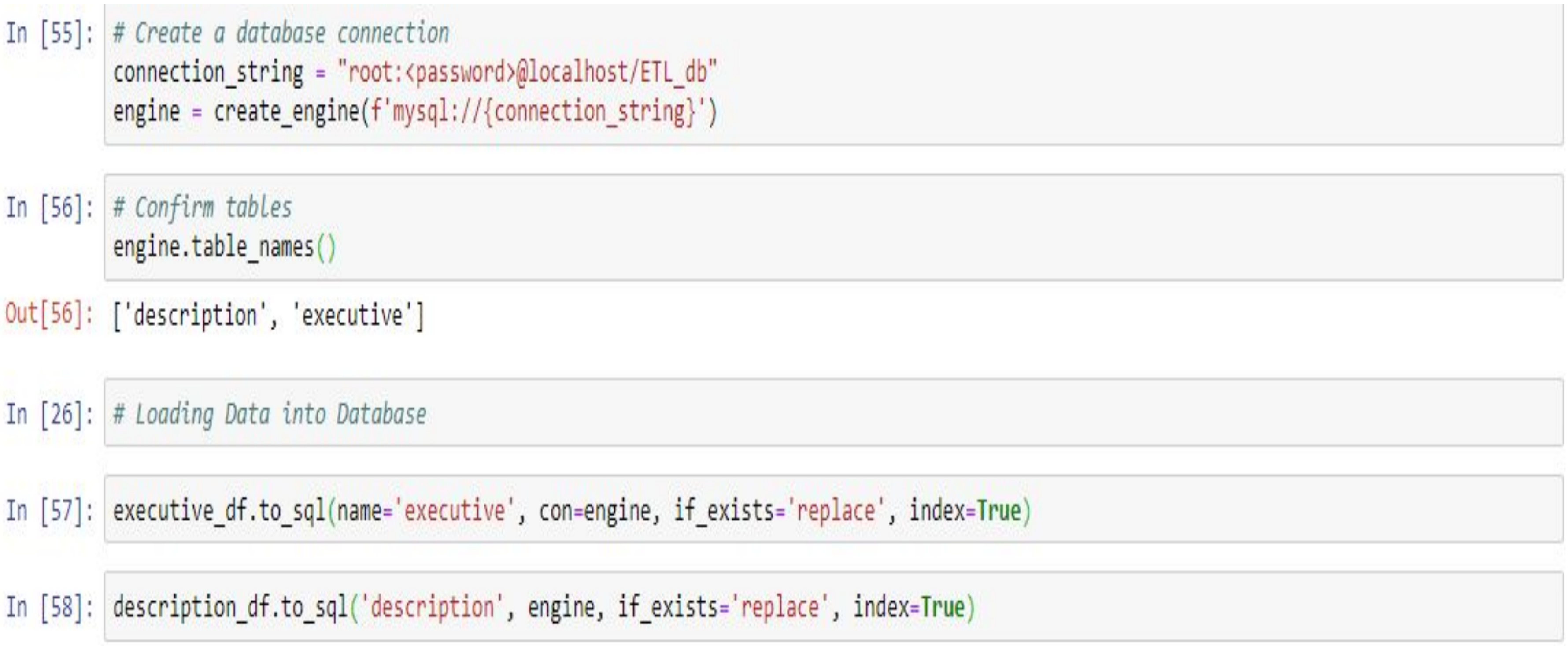
SQL CODE



```
DROP DATABASE IF EXISTS ETL_db;
CREATE DATABASE ETL_db;
USE ETL_db;

CREATE TABLE description (
Exec_Ord FLOAT(10,1),
Topic INT,
Topic_Title TEXT,
Sub_Topic INT,
Sub_Topic_Title TEXT,
Description TEXT
);
CREATE TABLE executive(
Exec_Ord FLOAT(10,1),
Year INT,
President TEXT,
Party TEXT,
Topic INT
);
```

Adote

Loading Data



```
In [55]: # Create a database connection
         connection_string = "root:<password>@localhost/ETL_db"
         engine = create_engine(f'mysql://{connection_string}')

In [56]: # Confirm tables
         engine.table_names()

Out[56]: ['description', 'executive']

In [26]: # Loading Data into Database

In [57]: executive_df.to_sql(name='executive', con=engine, if_exists='replace', index=True)

In [58]: description_df.to_sql('description', engine, if_exists='replace', index=True)
```

Adote