

```
In [1]: #Generate a 3X4 NumPy array after seeding the random generator
#Save the array as a CSV named np.csv
#View the np field with the cat command (doesn't need to be included in code, just so
#Create a DataFrame from the file and print the results
#Write the DataFrame to a CSV file

#Load np, pd
import numpy as np
import pandas as pd

#random seed
np.random.seed(0)

#3x4 numpy array
array = np.random.random((3, 4))

#save the array as a CSV named np.csv
np.savetxt('np.csv', array, delimiter=',')

#csv file into dataframe
data = pd.read_csv('np.csv', header=None)
print("dataFrame from the csv file:")
print(data)

#write dataframe into csv file
data.to_csv('output.csv', index=False)

dataFrame from the csv file:
      0      1      2      3
0  0.548814  0.715189  0.602763  0.544883
1  0.423655  0.645894  0.437587  0.891773
2  0.963663  0.383442  0.791725  0.528895
```

```
In [2]: #Generate a 365X4 NumPy array with random values
#Store the array in a CSV file and check its size
#Save the array in the NumPy format, load the array, check its shape and the size of t
#Create a DataFrame from this array you have created and write it to a pickle, then re
#Print the size of the pickle

#Load np, pd
import numpy as np
import pandas as pd

#365x4 numpy array with random values
np_array = np.random.random((365, 4))

#store array in a CSV file
np.savetxt('numpy_array.csv', np_array, delimiter=',')

#check size of csv file (bytes)
import os
csv_size = os.path.getsize('numpy_array.csv')
print(f"size of the csv file: {csv_size} bytes")

#save and load array
```

```

np.save('numpy_array.npy', np_array)
loaded_array = np.load('numpy_array.npy')

#check shape and size of the file (bytes)
print(f"shape of the loaded array: {loaded_array.shape}")
np_size = os.path.getsize('numpy_array.npy')
print(f"size of the numpy file: {np_size} bytes")

#make df and write it to a pickle
df = pd.DataFrame(loaded_array)
df.to_pickle('numpy_dataframe.pkl')

#retrieve df from pickle
retrieved_df = pd.read_pickle('numpy_dataframe.pkl')

#print
pkl_size = os.path.getsize('numpy_dataframe.pkl')
print(f"size of the pickle file: {pkl_size} bytes")

```

size of the csv file: 36865 bytes
 shape of the loaded array: (365, 4)
 size of the numpy file: 11808 bytes
 size of the pickle file: 12239 bytes

In [3]: *#Using the array created in #2, create an excel file with this data
 #After that is complete, create a dataframe from the excel
 #Print your results*

```

#Load np ,pd
import numpy as np
import pandas as pd

#creating an excel file from the numpy array
df = pd.DataFrame(np_array)
df.to_excel('numpy_data.xlsx', index=False)

#make df from the excel file
df_from_excel = pd.read_excel('numpy_data.xlsx')
print("dataFrame from the excel file:")
print(df_from_excel)

```

dataFrame from the excel file:

	0	1	2	3
0	0.568045	0.925597	0.071036	0.087129
1	0.020218	0.832620	0.778157	0.870012
2	0.978618	0.799159	0.461479	0.780529
3	0.118274	0.639921	0.143353	0.944669
4	0.521848	0.414662	0.264556	0.774234
..
360	0.881188	0.916419	0.271551	0.607545
361	0.526584	0.537946	0.937663	0.305189
362	0.983434	0.902131	0.458723	0.817453
363	0.769047	0.677895	0.319834	0.196451
364	0.671528	0.842973	0.016253	0.642803

[365 rows x 4 columns]

In [5]: *#Using this JSON string, parse a JSON string with the Loads() function
 #'{"country": "Netherlands", "dma_code": "0", "timezone": "Europe/Amsterdam"}*

```

#, "area_code": "0", "ip": "46.19.37.108", "asn": "AS196752", "continent_code":
#"EU", "isp": "Tilaa V.O.F.", "longitude": 5.75, "latitude": 52.5, "country_code":
#"NL", "country_code3": "NLD"}'
#Print the values for the "Country" column
#Overwrite the value for Netherlands with a value of your choice
#Print your results

#Load json
import json

#json string
json_string = '{"country": "Netherlands", "dma_code": "0", "timezone": "Europe/Amsterdam", "
data = json.loads(json_string)

#print the value for the "country" column
print("value for the 'country' column:", data['country'])

#overwrite the value for "netherlands" with a new value
data['country'] = "United States"

#print results
print("Updated value for the 'country' column:", data['country'])

```

value for the 'country' column: Netherlands
Updated value for the 'country' column: United States

In [6]: #Using the Pandas read_json() function, we can either create a pandas Series or DataFr
#Change the country value again to your choice and convert the Pandas Series to a JSON

```

#Load pd
import pandas as pd

#json string
json_string = '{"country": "Netherlands", "dma_code": "0", "timezone": "Europe/Amsterdam", "
#create pandas series from jsno
series_data = pd.read_json(json_string, typ='series')

#change country value again
series_data['country'] = "Australia"

#convert pandas series to json string
json_string_updated = series_data.to_json()
print("updated JSON string:")
print(json_string_updated)

```

updated JSON string:
{"country": "Australia", "dma_code": "0", "timezone": "Europe\Amsterdam", "area_cod
e": "0", "ip": "46.19.37.108", "asn": "AS196752", "continent_code": "EU", "isp": "Tilaa V.O.
F.", "longitude": 5.75, "latitude": 52.5, "country_code": "NL", "country_code3": "NLD"}