



School of IT & Business Technologies
Graduate Diploma in Data Analytics (Level 7)
Cover Sheet and Student Declaration

This sheet must be signed by the student and attached to the submitted assessment.

Course Title:	Data Transformation and Management	Course code:	GDDA612
Student Name:	Mira Torrit	Student ID:	764707793
Assessment No & Type:	Assessment 1 - Project 1	Cohort:	GDDA7123C
Due Date:	02/02/24	Date Submitted:	02/02/24
Tutor's Name:	Harsh Tiwari		
Assessment Weighting:	60%		
Total Marks:	100		

Student Declaration:

I declare that:

- I have read the New Zealand School of Education Ltd policies and regulations on assessments and understand what plagiarism is.
- I am aware of the penalties for cheating and plagiarism as laid down by the New Zealand School of Education Ltd.
- This is an original assessment and is entirely my own work.
- Where I have quoted or made use of the ideas of other writers, I have acknowledged the source.
- This assessment has been prepared exclusively for this course and has not been or will not be submitted as assessed work in any other course.
- It has been explained to me that this assessment may be used by NZSE Ltd, for internal and/or external moderation.

Student signature:

Date: 02/02/24

Tutor only to complete			
	Part A (max. 20 marks)	Part B (max. 35 marks)	Part C (max. 45 marks)
Assessment result:	Marks: /100	Grade:	

Part A

Task 1: Data Sources Data Source 1: ("https://en.wikipedia.org/wiki/Gold_reserve" (https://en.wikipedia.org/wiki/Gold_reserve")) Description: A gold reserve is being held by country's central bank for various purposes such as guarantee to pay depositors, note holders or trading peers during the eras of the gold standard. It also preserves the value and uphold the stability of the national currency. This data show the top 50 countries with gold reserves as of January 2024. Data Source 2: ("<https://www.visualcapitalist.com/200-years-of-global-gold-production-by-country/>" (<https://www.visualcapitalist.com/200-years-of-global-gold-production-by-country/>")) Description: This website exhibits the evolution of gold mining and its industry growth over the years. As of 2022, the top 3 countries producers are China, Russia and Australia. Data Source 3: ("<https://tradingeconomics.com/country-list/gold-reserves>" (<https://tradingeconomics.com/country-list/gold-reserves>")) Description: This page shows the world gold reserves as of September 2023.

Task 2: a) I used the HTTP, User-Agent to simulate a common web browser to avoid being blocked by the website. This code checks the HTTP on whether the response status code 200 allows data extraction. The python library Beautiful soup used for locating the data table. b) The scraping process follows ethical standards by using the User-agent header. This allows the website administrator to contact you if there are issues with the web scraping activities, checking the website's terms of service and privacy considerations. No sensitive data was taken during the web scraping. c) It is important to consider responsible and ethical practices during web scraping by incorporating the principles of Tikanga (cultural sensitivity), Whakapapa (respect) and Pepeha (consent). The data collection should obtain consent before it is done in consideration with the cultural sensitivity. These cultural protocols ensure positive impact on the community. It also maintains the trust between the web scraper and web administrators. And lastly, it avoids errors or risks in legal compliance on collecting data.

Part B

Task 3 : Data Preparation and Cleansing

Dataset 1 - Top 50 World Gold Holdings as of January 2024

```
In [1]: import requests
import pandas as pd
from bs4 import BeautifulSoup

def get_datasource_1(url):

    headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.6010.129 Safari/537.36'}
    response = requests.get(url, headers=headers)
    if response.status_code == 200:
        soup = BeautifulSoup(response.text, "html.parser")
        table = soup.find('table', class_="wikitable")

        if table:
            header_rows = table.find("tbody")
            headers = [header.text.strip() for header in header_rows.find_all("th")]
            data = []
            data_rows = table.find_all("tr")
            for row in data_rows[1:]:
                row_data = [cell.text.strip() for cell in row.find_all("td")]
                data.append(row_data)
            df = pd.DataFrame(data, columns=headers)
            return df
        else:
            return None

gold_holdings_Jan2024_df = get_datasource_1("https://en.wikipedia.org/wiki/Gold_reserves_by_country_and_type")
gold_holdings_Jan2024_df.head(100)
```

Out[1]:

Rank		Country/Organization	Gold holdings(in metric tons)	Gold's share offorex reserves
0	1	United States	8,133.5	69.6%
1	2	Germany	3,352.6	68.7%
2	—	International Monetary Fund	2,814.0	—[a]
3	3	Italy	2,451.8	65.5%
4	4	France	2,436.9	67.1%
5	5	Russia	2,332.7	25.7%
6	6	China	2,226.4	4.3%
7	7	Switzerland	1,040.0	8.4%
8	8	Japan	846.0	4.4%
9	9	India	803.6	8.6%
10	10	Netherlands	612.5	57.9%
11	11	Turkey	522.5[b]	30.8%
12	—	European Central Bank	506.5	40.5%
13	12	Taiwan	422.4	4.7%
14	13	Portugal	382.6	72.9%
15	14	Uzbekistan	362.0	72.1%
16	15	Poland	358.7	12.6%
17	16	Saudi Arabia	323.1	4.7%
18	17	United Kingdom	310.3	11.6%
19	18	Kazakhstan	304.3	58.5%
20	19	Lebanon	286.8	56.5%
21	20	Spain	281.6	18.2%
22	21	Austria	280.0	59.4%
23	22	Thailand	244.2	7.3%
24	23	Singapore	230.3	4.3%
25	24	Belgium	227.4	36.8%
26	25	Algeria	173.6	13.9%
27	26	Philippines	165.6	10.7%
28	27	Venezuela	161.2	84.5%
29	28	Iraq	132.7	7.9%
30	29	Brazil	129.7	2.4%
31	30	Egypt	126.0	24.6%
32	31	Sweden	125.7	13.9%
33	32	South Africa	125.4	13.3%
34	33	Mexico	120.0	3.7%
35	34	Libya	116.6	9.4%
36	35	Greece	114.4	55.9%
37	36	South Korea	104.4	1.7%
38	37	Romania	103.6	9.5%
39	—	Bank for International Settlements	102.0[c]	—[a]
40	38	Qatar	99.2	12.7%

Rank	Country/Organization	Gold holdings(in metric tons)	Gold's share offorex reserves
41	39	Hungary	94.5
42	40	Australia	79.8
43	41	Kuwait	79.0
44	42	Indonesia	78.6
45	43	United Arab Emirates	74.6
46	44	Denmark	66.5
47	45	Pakistan	64.7
48	46	Argentina	61.7
49	47	Belarus	53.8
50	48	Finland	49.0
51	49	Jordan	43.5
52	50	Bolivia	42.5
53	—	World	35,809.3[d]
54	—	Euro Area (including the ECB)	10,771.4

In [2]: gold_holdings_Jan2024_df.shape

Out[2]: (55, 4)

In [3]: gold_holdings_Jan2024_df.isnull().sum()

Out[3]: Rank 0
 Country/Organization 0
 Gold holdings(in metric tons) 0
 Gold's share offorex reserves 0
 dtype: int64

In [4]: #1) Removing outliers: removed entries with no rank

```
gold_holdings_Jan2024_df[ "Rank" ] = pd.to_numeric(gold_holdings_Jan2024_df[ "Rank" ].rep  
gold_holdings_Jan2024_df = gold_holdings_Jan2024_df.dropna(subset=[ "Rank" ])  
gold_holdings_Jan2024_df.head(100)
```

Out[4]:

Rank	Country/Organization	Gold holdings(in metric tons)	Gold's share offorex reserves
0	1.0	United States	8,133.5
1	2.0	Germany	3,352.6
3	3.0	Italy	2,451.8
4	4.0	France	2,436.9
5	5.0	Russia	2,332.7
6	6.0	China	2,226.4
7	7.0	Switzerland	1,040.0
8	8.0	Japan	846.0
9	9.0	India	803.6
10	10.0	Netherlands	612.5
11	11.0	Turkey	522.5[b]
13	12.0	Taiwan	422.4
14	13.0	Portugal	382.6
15	14.0	Uzbekistan	362.0
16	15.0	Poland	358.7
17	16.0	Saudi Arabia	323.1
18	17.0	United Kingdom	310.3
19	18.0	Kazakhstan	304.3
20	19.0	Lebanon	286.8
21	20.0	Spain	281.6
22	21.0	Austria	280.0
23	22.0	Thailand	244.2
24	23.0	Singapore	230.3
25	24.0	Belgium	227.4
26	25.0	Algeria	173.6
27	26.0	Philippines	165.6
28	27.0	Venezuela	161.2
29	28.0	Iraq	132.7
30	29.0	Brazil	129.7
31	30.0	Egypt	126.0
32	31.0	Sweden	125.7
33	32.0	South Africa	125.4
34	33.0	Mexico	120.0
35	34.0	Libya	116.6
36	35.0	Greece	114.4
37	36.0	South Korea	104.4
38	37.0	Romania	103.6
40	38.0	Qatar	99.2
41	39.0	Hungary	94.5
42	40.0	Australia	79.8
43	41.0	Kuwait	79.0

Rank	Country/Organization	Gold holdings(in metric tons)	Gold's share offorex reserves
44	42.0	Indonesia	78.6
45	43.0	United Arab Emirates	74.6
46	44.0	Denmark	66.5
47	45.0	Pakistan	64.7
48	46.0	Argentina	61.7
49	47.0	Belarus	53.8
50	48.0	Finland	49.0
51	49.0	Jordan	43.5
52	50.0	Bolivia	42.5

In [5]: #2 Removing irrelevant data: retain only numeric values in gold holdings column
gold_holdings_Jan2024_df['Gold holdings(in metric tons)'] = gold_holdings_Jan2024_df[
gold_holdings_Jan2024_df.head(100)

Out[5]:

Rank	Country/Organization	Gold holdings(in metric tons)	Gold's share offorex reserves
0	1.0	United States	8,133.5
1	2.0	Germany	3,352.6
3	3.0	Italy	2,451.8
4	4.0	France	2,436.9
5	5.0	Russia	2,332.7
6	6.0	China	2,226.4
7	7.0	Switzerland	1,040.0
8	8.0	Japan	846.0
9	9.0	India	803.6
10	10.0	Netherlands	612.5
11	11.0	Turkey	522.5
13	12.0	Taiwan	422.4
14	13.0	Portugal	382.6
15	14.0	Uzbekistan	362.0
16	15.0	Poland	358.7
17	16.0	Saudi Arabia	323.1
18	17.0	United Kingdom	310.3
19	18.0	Kazakhstan	304.3
20	19.0	Lebanon	286.8
21	20.0	Spain	281.6
22	21.0	Austria	280.0
23	22.0	Thailand	244.2
24	23.0	Singapore	230.3
25	24.0	Belgium	227.4
26	25.0	Algeria	173.6
27	26.0	Philippines	165.6
28	27.0	Venezuela	161.2
29	28.0	Iraq	132.7
30	29.0	Brazil	129.7
31	30.0	Egypt	126.0
32	31.0	Sweden	125.7
33	32.0	South Africa	125.4
34	33.0	Mexico	120.0
35	34.0	Libya	116.6
36	35.0	Greece	114.4
37	36.0	South Korea	104.4
38	37.0	Romania	103.6
40	38.0	Qatar	99.2
41	39.0	Hungary	94.5
42	40.0	Australia	79.8
43	41.0	Kuwait	79.0

Rank	Country/Organization	Gold holdings(in metric tons)	Gold's share offorex reserves
44	42.0	Indonesia	78.6
45	43.0	United Arab Emirates	74.6
46	44.0	Denmark	66.5
47	45.0	Pakistan	64.7
48	46.0	Argentina	61.7
49	47.0	Belarus	53.8
50	48.0	Finland	49.0
51	49.0	Jordan	43.5
52	50.0	Bolivia	42.5

In [6]: #3 Converting Units: Convert from Metric Tones to Tonnes.

```
exchange_rate = 1.1023122100918887
gold_holdings_Jan2024_df['Gold holdings(in tons)'] = pd.to_numeric(gold_holdings_Jan2024_df['Gold holdings(in tons)'], errors='coerce')
gold_holdings_Jan2024_df.head(100)
```

Out[6]:

Rank	Country/Organization	Gold holdings(in metric tons)	Gold's share offorex reserves	Gold holdings(in tons)
0	1.0 United States	8,133.5	69.6%	8965.656361
1	2.0 Germany	3,352.6	68.7%	3695.611916
3	3.0 Italy	2,451.8	65.5%	2702.649077
4	4.0 France	2,436.9	67.1%	2686.224625
5	5.0 Russia	2,332.7	25.7%	2571.363692
6	6.0 China	2,226.4	4.3%	2454.187905
7	7.0 Switzerland	1,040.0	8.4%	1146.404698
8	8.0 Japan	846.0	4.4%	932.556130
9	9.0 India	803.6	8.6%	885.818092
10	10.0 Netherlands	612.5	57.9%	675.166229
11	11.0 Turkey	522.5	30.8%	575.958130
13	12.0 Taiwan	422.4	4.7%	465.616678
14	13.0 Portugal	382.6	72.9%	421.744652
15	14.0 Uzbekistan	362.0	72.1%	399.037020
16	15.0 Poland	358.7	12.6%	395.399390
17	16.0 Saudi Arabia	323.1	4.7%	356.157075
18	17.0 United Kingdom	310.3	11.6%	342.047479
19	18.0 Kazakhstan	304.3	58.5%	335.433606
20	19.0 Lebanon	286.8	56.5%	316.143142
21	20.0 Spain	281.6	18.2%	310.411118
22	21.0 Austria	280.0	59.4%	308.647419
23	22.0 Thailand	244.2	7.3%	269.184642
24	23.0 Singapore	230.3	4.3%	253.862502
25	24.0 Belgium	227.4	36.8%	250.665797
26	25.0 Algeria	173.6	13.9%	191.361400
27	26.0 Philippines	165.6	10.7%	182.542902
28	27.0 Venezuela	161.2	84.5%	177.692728
29	28.0 Iraq	132.7	7.9%	146.276830
30	29.0 Brazil	129.7	2.4%	142.969894
31	30.0 Egypt	126.0	24.6%	138.891338
32	31.0 Sweden	125.7	13.9%	138.560645
33	32.0 South Africa	125.4	13.3%	138.229951
34	33.0 Mexico	120.0	3.7%	132.277465
35	34.0 Libya	116.6	9.4%	128.529604
36	35.0 Greece	114.4	55.9%	126.104517
37	36.0 South Korea	104.4	1.7%	115.081395
38	37.0 Romania	103.6	9.5%	114.199545
40	38.0 Qatar	99.2	12.7%	109.349371
41	39.0 Hungary	94.5	13.8%	104.168504
42	40.0 Australia	79.8	8.9%	87.964514

Rank	Country/Organization	Gold holdings(in metric tons)	Gold's share of forex reserves	Gold holdings(in tons)	
43	41.0	Kuwait	79.0	10.3%	87.082665
44	42.0	Indonesia	78.6	3.9%	86.641740
45	43.0	United Arab Emirates	74.6	2.9%	82.232491
46	44.0	Denmark	66.5	4.3%	73.303762
47	45.0	Pakistan	64.7	32.8%	71.319600
48	46.0	Argentina	61.7	18.8%	68.012663
49	47.0	Belarus	53.8	42.9%	59.304397
50	48.0	Finland	49.0	18.9%	54.013298
51	49.0	Jordan	43.5	18.0%	47.950581
52	50.0	Bolivia	42.5	87.5%	46.848269

```
In [7]: # Removing irrelevant data: Removed Gold holdings(in metric tons) as it was already covered by the total assets
gold_holdings_Jan2024_df = gold_holdings_Jan2024_df.drop('Gold holdings(in metric tons)', axis=1)
gold_holdings_Jan2024_df.head(100)
```

Out[7]:

Rank	Country/Organization	Gold's share offorex reserves	Gold holdings(in tons)
0	1.0	United States	69.6% 8965.656361
1	2.0	Germany	68.7% 3695.611916
3	3.0	Italy	65.5% 2702.649077
4	4.0	France	67.1% 2686.224625
5	5.0	Russia	25.7% 2571.363692
6	6.0	China	4.3% 2454.187905
7	7.0	Switzerland	8.4% 1146.404698
8	8.0	Japan	4.4% 932.556130
9	9.0	India	8.6% 885.818092
10	10.0	Netherlands	57.9% 675.166229
11	11.0	Turkey	30.8% 575.958130
13	12.0	Taiwan	4.7% 465.616678
14	13.0	Portugal	72.9% 421.744652
15	14.0	Uzbekistan	72.1% 399.037020
16	15.0	Poland	12.6% 395.399390
17	16.0	Saudi Arabia	4.7% 356.157075
18	17.0	United Kingdom	11.6% 342.047479
19	18.0	Kazakhstan	58.5% 335.433606
20	19.0	Lebanon	56.5% 316.143142
21	20.0	Spain	18.2% 310.411118
22	21.0	Austria	59.4% 308.647419
23	22.0	Thailand	7.3% 269.184642
24	23.0	Singapore	4.3% 253.862502
25	24.0	Belgium	36.8% 250.665797
26	25.0	Algeria	13.9% 191.361400
27	26.0	Philippines	10.7% 182.542902
28	27.0	Venezuela	84.5% 177.692728
29	28.0	Iraq	7.9% 146.276830
30	29.0	Brazil	2.4% 142.969894
31	30.0	Egypt	24.6% 138.891338
32	31.0	Sweden	13.9% 138.560645
33	32.0	South Africa	13.3% 138.229951
34	33.0	Mexico	3.7% 132.277465
35	34.0	Libya	9.4% 128.529604
36	35.0	Greece	55.9% 126.104517
37	36.0	South Korea	1.7% 115.081395
38	37.0	Romania	9.5% 114.199545
40	38.0	Qatar	12.7% 109.349371
41	39.0	Hungary	13.8% 104.168504
42	40.0	Australia	8.9% 87.964514
43	41.0	Kuwait	10.3% 87.082665

Rank	Country/Organization	Gold's share offorex reserves	Gold holdings(in tons)
44	42.0	Indonesia	3.9% 86.641740
45	43.0	United Arab Emirates	2.9% 82.232491
46	44.0	Denmark	4.3% 73.303762
47	45.0	Pakistan	32.8% 71.319600
48	46.0	Argentina	18.8% 68.012663
49	47.0	Belarus	42.9% 59.304397
50	48.0	Finland	18.9% 54.013298
51	49.0	Jordan	18.0% 47.950581
52	50.0	Bolivia	87.5% 46.848269

In [8]: #4 Data type conversion: Convert Rank to integer

```
gold_holdings_Jan2024_df['Rank'] = gold_holdings_Jan2024_df['Rank'].astype(int)
gold_holdings_Jan2024_df.head(100)
```

Out[8]:

Rank	Country/Organization	Gold's share offorex reserves	Gold holdings(in tons)
0	1	United States	69.6% 8965.656361
1	2	Germany	68.7% 3695.611916
3	3	Italy	65.5% 2702.649077
4	4	France	67.1% 2686.224625
5	5	Russia	25.7% 2571.363692
6	6	China	4.3% 2454.187905
7	7	Switzerland	8.4% 1146.404698
8	8	Japan	4.4% 932.556130
9	9	India	8.6% 885.818092
10	10	Netherlands	57.9% 675.166229
11	11	Turkey	30.8% 575.958130
13	12	Taiwan	4.7% 465.616678
14	13	Portugal	72.9% 421.744652
15	14	Uzbekistan	72.1% 399.037020
16	15	Poland	12.6% 395.399390
17	16	Saudi Arabia	4.7% 356.157075
18	17	United Kingdom	11.6% 342.047479
19	18	Kazakhstan	58.5% 335.433606
20	19	Lebanon	56.5% 316.143142
21	20	Spain	18.2% 310.411118
22	21	Austria	59.4% 308.647419
23	22	Thailand	7.3% 269.184642
24	23	Singapore	4.3% 253.862502
25	24	Belgium	36.8% 250.665797
26	25	Algeria	13.9% 191.361400
27	26	Philippines	10.7% 182.542902
28	27	Venezuela	84.5% 177.692728
29	28	Iraq	7.9% 146.276830
30	29	Brazil	2.4% 142.969894
31	30	Egypt	24.6% 138.891338
32	31	Sweden	13.9% 138.560645
33	32	South Africa	13.3% 138.229951
34	33	Mexico	3.7% 132.277465
35	34	Libya	9.4% 128.529604
36	35	Greece	55.9% 126.104517
37	36	South Korea	1.7% 115.081395
38	37	Romania	9.5% 114.199545
40	38	Qatar	12.7% 109.349371
41	39	Hungary	13.8% 104.168504
42	40	Australia	8.9% 87.964514
43	41	Kuwait	10.3% 87.082665

Rank	Country/Organization	Gold's share offorex reserves	Gold holdings(in tons)
44	42	Indonesia	3.9% 86.641740
45	43	United Arab Emirates	2.9% 82.232491
46	44	Denmark	4.3% 73.303762
47	45	Pakistan	32.8% 71.319600
48	46	Argentina	18.8% 68.012663
49	47	Belarus	42.9% 59.304397
50	48	Finland	18.9% 54.013298
51	49	Jordan	18.0% 47.950581
52	50	Bolivia	87.5% 46.848269

In [9]: #5 Encoding/decoding and re-coding variables

```
columns_to_rename = { 'Country/Organization' : 'Country', 'Gold holdings(in tons)' :  
gold_holdings_Jan2024_df.rename(columns=columns_to_rename, inplace=True)  
gold_holdings_Jan2024_df.head(100)
```

Out[9]:

Rank		Country	Gold's share offorex reserves	Gold holdings(in tons) as of Jan2024
0	1	United States	69.6%	8965.656361
1	2	Germany	68.7%	3695.611916
3	3	Italy	65.5%	2702.649077
4	4	France	67.1%	2686.224625
5	5	Russia	25.7%	2571.363692
6	6	China	4.3%	2454.187905
7	7	Switzerland	8.4%	1146.404698
8	8	Japan	4.4%	932.556130
9	9	India	8.6%	885.818092
10	10	Netherlands	57.9%	675.166229
11	11	Turkey	30.8%	575.958130
13	12	Taiwan	4.7%	465.616678
14	13	Portugal	72.9%	421.744652
15	14	Uzbekistan	72.1%	399.037020
16	15	Poland	12.6%	395.399390
17	16	Saudi Arabia	4.7%	356.157075
18	17	United Kingdom	11.6%	342.047479
19	18	Kazakhstan	58.5%	335.433606
20	19	Lebanon	56.5%	316.143142
21	20	Spain	18.2%	310.411118
22	21	Austria	59.4%	308.647419
23	22	Thailand	7.3%	269.184642
24	23	Singapore	4.3%	253.862502
25	24	Belgium	36.8%	250.665797
26	25	Algeria	13.9%	191.361400
27	26	Philippines	10.7%	182.542902
28	27	Venezuela	84.5%	177.692728
29	28	Iraq	7.9%	146.276830
30	29	Brazil	2.4%	142.969894
31	30	Egypt	24.6%	138.891338
32	31	Sweden	13.9%	138.560645
33	32	South Africa	13.3%	138.229951
34	33	Mexico	3.7%	132.277465
35	34	Libya	9.4%	128.529604
36	35	Greece	55.9%	126.104517
37	36	South Korea	1.7%	115.081395
38	37	Romania	9.5%	114.199545
40	38	Qatar	12.7%	109.349371
41	39	Hungary	13.8%	104.168504
42	40	Australia	8.9%	87.964514
43	41	Kuwait	10.3%	87.082665

Rank	Country	Gold's share offorex reserves	Gold holdings(in tons) as of Jan2024
44	42	Indonesia	3.9%
45	43	United Arab Emirates	2.9%
46	44	Denmark	4.3%
47	45	Pakistan	32.8%
48	46	Argentina	18.8%
49	47	Belarus	42.9%
50	48	Finland	18.9%
51	49	Jordan	18.0%
52	50	Bolivia	87.5%

Dataset 2: 200 Years of Global Gold Production, by Country

```
In [10]: def get_datasource_2(url):

    headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4453.102 Safari/537.36'}
    response = requests.get(url, headers=headers)

    if response.status_code == 200:

        soup = BeautifulSoup(response.text, "html.parser")
        table = soup.find('table', id="tablepress-3185")
        if table:
            header_rows = table.find("thead")
            headers = [header.text.strip() for header in header_rows.find_all("th")]
            data = []
            data_rows = table.find_all("tr")
            for row in data_rows[1:]:
                row_data = [cell.text.strip() for cell in row.find_all("td")]
                data.append(row_data)
            df = pd.DataFrame(data, columns=headers)
            return df
        else:
            return None

gold_production_2022_df = get_datasource_2("https://www.visualcapitalist.com/200-years-of-global-gold-production-by-country/")
print(gold_production_2022_df)
```

Rank	Country	2022E Gold Production, tonnes	% of Total
0	#1 CN China	330	11%
1	#2 RU Russia	320	10%
2	#3 AU Australia	320	10%
3	#4 CA Canada	220	7%
4	#5 us United States	170	5%
5	#6 MX Mexico	120	4%
6	#7 KZ Kazakhstan	120	4%
7	#8 ZA South Africa	110	4%
8	#9 PE Peru	100	3%
9	#10 UZ Uzbekistan	100	3%
10	#11 GH Ghana	90	3%
11	#12 ID Indonesia	70	2%
12	- Rest of the World	1,030	33%
13	None	None	None

```
In [11]: gold_production_2022_df.shape
```

```
Out[11]: (14, 4)
```

```
In [12]: gold_production_2022_df.isnull().sum()
```

```
Out[12]: Rank          1  
Country        1  
2022E Gold Production, tonnes 1  
% of Total     1  
dtype: int64
```

```
In [13]: #1) Data type conversion of rank from string to float
```

```
gold_production_2022_df["Rank"] = pd.to_numeric(gold_production_2022_df["Rank"].replace
```

```
<
```

```
>
```

```
In [14]: #2) Removing outliers
```

```
gold_production_2022_df = gold_production_2022_df.dropna(subset=["Rank"])  
gold_production_2022_df.head(100)
```

```
Out[14]:
```

	Rank	Country	2022E Gold Production, tonnes	% of Total
0	1.0	cn China	330	11%
1	2.0	ru Russia	320	10%
2	3.0	AU Australia	320	10%
3	4.0	CA Canada	220	7%
4	5.0	us United States	170	5%
5	6.0	mx Mexico	120	4%
6	7.0	kz Kazakhstan	120	4%
7	8.0	za South Africa	110	4%
8	9.0	PE Peru	100	3%
9	10.0	uz Uzbekistan	100	3%
10	11.0	gh Ghana	90	3%
11	12.0	ID Indonesia	70	2%

```
In [15]: gold_production_2022_df.shape
```

```
Out[15]: (12, 4)
```

```
In [16]: gold_production_2022_df.isnull().sum()
```

```
Out[16]: Rank          0  
Country        0  
2022E Gold Production, tonnes 0  
% of Total     0  
dtype: int64
```

In [17]: #3) Removing unwanted observations: Update Country column and retain only Country name
gold_production_2022_df["Country"] = gold_production_2022_df["Country"].str.split(".").str[0]
gold_production_2022_df.head(100)

Out[17]:

	Rank	Country	2022E Gold Production, tonnes	% of Total
0	1.0	China	330	11%
1	2.0	Russia	320	10%
2	3.0	Australia	320	10%
3	4.0	Canada	220	7%
4	5.0	United	170	5%
5	6.0	Mexico	120	4%
6	7.0	Kazakhstan	120	4%
7	8.0	South	110	4%
8	9.0	Peru	100	3%
9	10.0	Uzbekistan	100	3%
10	11.0	Ghana	90	3%
11	12.0	Indonesia	70	2%

In [18]: #4) Removing irrelevant data: Remove % of Total as it is not needed
gold_production_2022_df = gold_production_2022_df.drop("% of Total", axis=1)
gold_production_2022_df.head(100)

Out[18]:

	Rank	Country	2022E Gold Production, tonnes
0	1.0	China	330
1	2.0	Russia	320
2	3.0	Australia	320
3	4.0	Canada	220
4	5.0	United	170
5	6.0	Mexico	120
6	7.0	Kazakhstan	120
7	8.0	South	110
8	9.0	Peru	100
9	10.0	Uzbekistan	100
10	11.0	Ghana	90
11	12.0	Indonesia	70

```
In [19]: #Convert Rank to Integer instead of float  
gold_production_2022_df[ "Rank" ] = gold_production_2022_df[ "Rank" ].astype(int)  
gold_production_2022_df.head(100)
```

Out[19]:

Rank	Country	2022E Gold Production, tonnes
0	1	China
1	2	Russia
2	3	Australia
3	4	Canada
4	5	United
5	6	Mexico
6	7	Kazakhstan
7	8	South
8	9	Peru
9	10	Uzbekistan
10	11	Ghana
11	12	Indonesia

```
In [20]: #5) Encoding/decoding and re-coding variable:
```

```
#Rename 2022E Gold Production, Tonnes to Gold Production 2022 as dataframe will only have one column  
#Rename Rank to Rank, 2022 to show that the ranking is for the year 2022  
gold_production_2022_df.rename(columns={'2022E Gold Production, tonnes' : 'Gold Production 2022'}, inplace=True)  
gold_production_2022_df.head(100)
```

Out[20]:

Rank, 2022	Country	Gold Production, 2022
0	1	China
1	2	Russia
2	3	Australia
3	4	Canada
4	5	United
5	6	Mexico
6	7	Kazakhstan
7	8	South
8	9	Peru
9	10	Uzbekistan
10	11	Ghana
11	12	Indonesia

Dataset 3: Gold Reserves

```
In [21]: def get_datasource_3(url):

    headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/100.0.4896.121 Safari/537.36'}
    response = requests.get(url, headers=headers)

    if response.status_code == 200:

        soup = BeautifulSoup(response.text, "html.parser")
        table = soup.find('table', class_="table-heatmap")

        if table:
            header_rows = table.find("thead")
            headers = [header.text.strip() for header in header_rows.find_all("th")]
            data = []
            data_rows = table.find_all("tr")
            for row in data_rows[1:]:
                row_data = [cell.text.strip() for cell in row.find_all("td")]
                data.append(row_data)
            df = pd.DataFrame(data, columns=headers)
            return df

    else:
        return None
```

```
gold_reserves_df = get_datasource_3("https://tradingeconomics.com/country-list/gold-reserves")
print(gold_reserves_df)
```

	Country	Last	Previous	Reference	Unit
0	United States	8133	8133	Sep/23	Tonnes
1	Germany	3353	3353	Sep/23	Tonnes
2	Italy	2452	2452	Sep/23	Tonnes
3	France	2437	2437	Sep/23	Tonnes
4	Russia	2333	2330	Sep/23	Tonnes
..
108	Armenia	0	0	Sep/23	Tonnes
109	Azerbaijan	0	0	Sep/23	Tonnes
110	Cameroon	0	0	Sep/23	Tonnes
111	Canada	0	0	Sep/23	Tonnes
112	Nicaragua	0	0	Sep/23	Tonnes

```
[113 rows x 5 columns]
```

```
In [22]: gold_reserves_df.shape
```

```
Out[22]: (113, 5)
```

```
In [23]: gold_reserves_df.isnull().sum()
```

```
Out[23]: Country      0
          Last       0
          Previous   0
          Reference  0
          Unit       0
          dtype: int64
```

In [24]: #1) Removing irrelevant data

```
#remove unit column as everything will be in Tonnes
#remove previous and reference as it is not needed
gold_reserves_df = gold_reserves_df.drop(columns=["Unit", "Previous", "Reference"])
gold_reserves_df.head(200)
```

Out[24]:

	Country	Last
0	United States	8133
1	Germany	3353
2	Italy	2452
3	France	2437
4	Russia	2333
...
108	Armenia	0
109	Azerbaijan	0
110	Cameroon	0
111	Canada	0
112	Nicaragua	0

113 rows × 2 columns

In [25]: #2) Data Type conversion: Convert 'Last' column from string to float

```
print(gold_reserves_df["Last"].dtype)
gold_reserves_df["Last"] = pd.to_numeric(gold_reserves_df["Last"], errors="coerce")
gold_reserves_df.head(200)
```

object

Out[25]:

	Country	Last
0	United States	8133.0
1	Germany	3353.0
2	Italy	2452.0
3	France	2437.0
4	Russia	2333.0
...
108	Armenia	0.0
109	Azerbaijan	0.0
110	Cameroon	0.0
111	Canada	0.0
112	Nicaragua	0.0

113 rows × 2 columns

In [26]: gold_reserves_df.isnull().sum()

Out[26]: Country 0
Last 0
dtype: int64

```
In [27]: #3) Removing outliers: #Remove Countries with 0 Gold Reserves  
gold_reserves_df = gold_reserves_df[gold_reserves_df["Last"] > 0]  
gold_reserves_df.head(200)
```

Out[27]:

	Country	Last
0	United States	8133.00
1	Germany	3353.00
2	Italy	2452.00
3	France	2437.00
4	Russia	2333.00
...
103	Estonia	0.25
104	Uruguay	0.10
105	Burundi	0.03
106	Fiji	0.03
107	Kenya	0.02

108 rows × 2 columns

```
In [28]: #4) Encoding/Decoding and re-coding variables  
#Rename Last column to Gold Reserves, 2023  
column_to_rename = {'Last' : 'Gold Reserves as of September 2023'}  
gold_reserves_df.rename(columns=column_to_rename, inplace=True)  
gold_reserves_df.head(200)
```

Out[28]:

	Country	Gold Reserves as of September 2023
0	United States	8133.00
1	Germany	3353.00
2	Italy	2452.00
3	France	2437.00
4	Russia	2333.00
...
103	Estonia	0.25
104	Uruguay	0.10
105	Burundi	0.03
106	Fiji	0.03
107	Kenya	0.02

108 rows × 2 columns

Task 4 : Documentation

- The most common data cleaning techniques I used are: 1) Encoding.decoding and re-coding variables such as renaming the column names to make it easy to understand; 2) Data type conversion such as converting the variables from string to float or float to integer for standardization; 3) Removing irrelevant data to enhance the quality and simplification of the accurate analysis; 4) removing unwanted observations such as the availability of both country name and its flag. This causes redundancy in the data making it complicated to analyse; and 5) removing outliers with unspecified country names. Since the common column is the country name, it is more accurate if the data specified the actual value of the

variable to make it more accurate. The cleaned dataset provides high-quality information for decision-making and strategic planning. It improves the accuracy of data, helping marketers to target the right audience and reducing wasted resources. It also facilitates analysis and customer relationships for an effective marketing strategy. (Please take a look at the individual cleaning application for the documentation)

b) The challenges I experienced in data gathering are the following: 1) It is hard to find uncleaned data on websites and APIs, social media platforms, etc. Due to the public display of data, the website admins make sure that the platform is comprehensible. To resolve this, I chose data with kind of unusual data structure or format like having a flag and a country (redundancy). 2) The preparation and cleaning of data can be tricky, especially when dealing with tools like Python with different types of data. To facilitate, it's important to define / figure out your expected outcome. The idea you came up with has made the research process more organized. 3) Web scraping is hard when you are not familiar with the HTML file. Although there are very useful libraries for scraping, not all can be done the same. To facilitate, additional researches were done such as using the "User-Agent" which "tells" the website which application is accessing the website such as mozilla firefox or any kind of browsers. I believe, an effective web scraper should have at least a basic knowledge on web development. 4) The diversity of the data sets makes it hard to merge it. Since it came from 3 sources, there are uniqueness on the format of each. To resolve this, data cleaning was done for standardization and data efficiency. 5) Not all data are accessible. It should be done responsibly by considering the values of respect with the privacy of the website owner. To make sure, always check the terms and conditions/ service of the website to avoid legal sanctions.

Part C

Task 5: Store Datasets - To save a dataframe to a csv file, use the pandas method 'to_csv'. The dataframe "gold_holdings_Jan2024_df" is saved to the file name "gold_production_2022.csv". The parameter "index=False" excludes the index column from the csv file.

```
In [29]: #save Dataframe to CSV  
gold_holdings_Jan2024_df.to_csv('gold_holdings_jan2024.csv', index=False)  
print('gold_holdings_jan2024.csv created')  
  
gold_holdings_jan2024.csv created
```

```
In [30]: #save Dataframe to CSV  
gold_production_2022_df.to_csv('gold_production_2022.csv', index=False)  
print('gold_production_2022.csv created')  
  
gold_production_2022.csv created
```

```
In [31]: #save Dataframe to CSV  
gold_reserves_df.to_csv('gold_reserves_2023.csv', index=False)  
print('gold_reserves_2023.csv created')  
  
gold_reserves_2023.csv created
```

Task 6: Merge Data - Using pandas 'read_csv', I separated the data sources per dataframe to avoid the mix up of records.

```
In [32]: gold_production_2022 = pd.read_csv('gold_production_2022.csv')  
gold_holdings_Jan2024 = pd.read_csv('gold_holdings_jan2024.csv')  
gold_reserves_2023 = pd.read_csv('gold_reserves_2023.csv')
```

In [33]: gold_production_2022

Out[33]:

	Rank, 2022	Country	Gold Production, 2022
0	1	China	330
1	2	Russia	320
2	3	Australia	320
3	4	Canada	220
4	5	United	170
5	6	Mexico	120
6	7	Kazakhstan	120
7	8	South	110
8	9	Peru	100
9	10	Uzbekistan	100
10	11	Ghana	90
11	12	Indonesia	70

In [34]: gold_holdings_Jan2024

Out[34]:

Rank		Country	Gold's share of forex reserves	Gold holdings(in tons) as of Jan2024
0	1	United States	69.6%	8965.656361
1	2	Germany	68.7%	3695.611916
2	3	Italy	65.5%	2702.649077
3	4	France	67.1%	2686.224625
4	5	Russia	25.7%	2571.363692
5	6	China	4.3%	2454.187905
6	7	Switzerland	8.4%	1146.404698
7	8	Japan	4.4%	932.556130
8	9	India	8.6%	885.818092
9	10	Netherlands	57.9%	675.166229
10	11	Turkey	30.8%	575.958130
11	12	Taiwan	4.7%	465.616678
12	13	Portugal	72.9%	421.744652
13	14	Uzbekistan	72.1%	399.037020
14	15	Poland	12.6%	395.399390
15	16	Saudi Arabia	4.7%	356.157075
16	17	United Kingdom	11.6%	342.047479
17	18	Kazakhstan	58.5%	335.433606
18	19	Lebanon	56.5%	316.143142
19	20	Spain	18.2%	310.411118
20	21	Austria	59.4%	308.647419
21	22	Thailand	7.3%	269.184642
22	23	Singapore	4.3%	253.862502
23	24	Belgium	36.8%	250.665797
24	25	Algeria	13.9%	191.361400
25	26	Philippines	10.7%	182.542902
26	27	Venezuela	84.5%	177.692728
27	28	Iraq	7.9%	146.276830
28	29	Brazil	2.4%	142.969894
29	30	Egypt	24.6%	138.891338
30	31	Sweden	13.9%	138.560645
31	32	South Africa	13.3%	138.229951
32	33	Mexico	3.7%	132.277465
33	34	Libya	9.4%	128.529604
34	35	Greece	55.9%	126.104517
35	36	South Korea	1.7%	115.081395
36	37	Romania	9.5%	114.199545
37	38	Qatar	12.7%	109.349371
38	39	Hungary	13.8%	104.168504
39	40	Australia	8.9%	87.964514
40	41	Kuwait	10.3%	87.082665

Rank	Country	Gold's share offorex reserves	Gold holdings(in tons) as of Jan2024
41	42	Indonesia	3.9%
42	43	United Arab Emirates	2.9%
43	44	Denmark	4.3%
44	45	Pakistan	32.8%
45	46	Argentina	18.8%
46	47	Belarus	42.9%
47	48	Finland	18.9%
48	49	Jordan	18.0%
49	50	Bolivia	87.5%

In [35]: gold_reserves_2023

Out[35]:

Country	Gold Reserves as of September 2023
0 United States	8133.00
1 Germany	3353.00
2 Italy	2452.00
3 France	2437.00
4 Russia	2333.00
...	...
103 Estonia	0.25
104 Uruguay	0.10
105 Burundi	0.03
106 Fiji	0.03
107 Kenya	0.02

108 rows × 2 columns

In [36]: #Merging via inner

#used inner join to show only common data present to all 3 data frames

```
gold_data_df = pd.merge(gold_production_2022, gold_holdings_Jan2024, on='Country', how='inner')
gold_data_df = pd.merge(gold_data_df, gold_reserves_2023, on='Country', how='inner')
gold_data_df.fillna(' ', inplace=True)
gold_data_df.head(100)
```

Out[36]:

Rank, 2022	Country	Gold Production, 2022	Rank	Gold's share offorex reserves	Gold holdings(in tons) as of Jan2024	Gold Reserves as of September 2023
0	1	China	330	6	4.3%	2454.187905
1	2	Russia	320	5	25.7%	2571.363692
2	3	Australia	320	40	8.9%	87.964514
3	6	Mexico	120	33	3.7%	132.277465
4	7	Kazakhstan	120	18	58.5%	335.433606
5	10	Uzbekistan	100	14	72.1%	399.037020
6	12	Indonesia	70	42	3.9%	86.641740

In [37]: #Merging via outer

```
#used outer join to show all common data present to all data frames without losing data
gold_data_df = pd.merge(gold_production_2022, gold_holdings_Jan2024, on='Country', how='outer')
gold_data_df = pd.merge(gold_data_df, gold_reserves_2023, on='Country', how='outer')
gold_data_df.fillna(' ', inplace=True)

gold_data_df.head(100)
```

Out[37]:

	Rank, 2022	Country	Gold Production, 2022	Rank	Gold's share offorex reserves	Gold holdings(in tons) as of Jan2024	Gold Reserves as of September 2023
0	1.0	China	330.0	6.0	4.3%	2454.187905	2192.0
1	2.0	Russia	320.0	5.0	25.7%	2571.363692	2333.0
2	3.0	Australia	320.0	40.0	8.9%	87.964514	79.85
3	4.0	Canada	220.0				
4	5.0	United	170.0				
...
95		Yemen					1.56
96		Bosnia and Herzegovina					1.49
97		Suriname					1.46
98		El Salvador					1.37
99		Papua New Guinea					1.33

100 rows × 7 columns

In [38]: #Merging via left

```
#used left join to include information from the left DataFrame even when there are no
gold_data_df = pd.merge(gold_production_2022, gold_holdings_Jan2024, on='Country', how='left')
gold_data_df = pd.merge(gold_data_df, gold_reserves_2023, on='Country', how='left')
gold_data_df.fillna(' ', inplace=True)

gold_data_df.head(100)
```

Out[38]:

	Rank, 2022	Country	Gold Production, 2022	Rank	Gold's share offorex reserves	Gold holdings(in tons) as of Jan2024	Gold Reserves as of September 2023
0	1	China	330	6.0	4.3%	2454.187905	2192.0
1	2	Russia	320	5.0	25.7%	2571.363692	2333.0
2	3	Australia	320	40.0	8.9%	87.964514	79.85
3	4	Canada	220				
4	5	United	170				
5	6	Mexico	120	33.0	3.7%	132.277465	120.0
6	7	Kazakhstan	120	18.0	58.5%	335.433606	309.0
7	8	South	110				
8	9	Peru	100				34.68
9	10	Uzbekistan	100	14.0	72.1%	399.03702	384.0
10	11	Ghana	90				8.74
11	12	Indonesia	70	42.0	3.9%	86.64174	78.57

In [39]: #Merging via right

```
#used right join to include information from the right DataFrame even when there are missing values
gold_data_df = pd.merge(gold_production_2022, gold_holdings_Jan2024, on='Country', how='right')
gold_data_df = pd.merge(gold_data_df, gold_reserves_2023, on='Country', how='right')
gold_data_df.fillna(' ', inplace=True)
gold_data_df.head(100)
```

Out[39]:

	Rank, 2022	Country	Gold Production, 2022	Rank	Gold's share of forex reserves	Gold holdings(in tons) as of Jan2024	Gold Reserves as of September 2023
0		United States		1.0	69.6%	8965.656361	8133.00
1		Germany		2.0	68.7%	3695.611916	3353.00
2		Italy		3.0	65.5%	2702.649077	2452.00
3		France		4.0	67.1%	2686.224625	2437.00
4	2.0	Russia	320.0	5.0	25.7%	2571.363692	2333.00
...
95		El Salvador					1.37
96		Papua New Guinea					1.33
97		Honduras					0.69
98		Dominican Republic					0.57
99		Sri Lanka					0.47

100 rows × 7 columns

Task 7 : Indexing - using the inner joined merged data, I used the Country column since this is the common among the three datasets.

In [40]: # Indexing using the inner join merged data

```
# Setting Country column as specific index.
gold_data_df = pd.merge(gold_production_2022, gold_holdings_Jan2024, on='Country', how='right')
gold_data_df = pd.merge(gold_data_df, gold_reserves_2023, on='Country', how='inner')

#Set a specific column as the index
gold_data_df.set_index('Country', inplace=True)
gold_data_df.head(100)
```

Out[40]:

	Rank, 2022	Gold Production, 2022	Rank	Gold's share of forex reserves	Gold holdings(in tons) as of Jan2024	Gold Reserves as of September 2023
Country						
China	1	330	6	4.3%	2454.187905	2192.00
Russia	2	320	5	25.7%	2571.363692	2333.00
Australia	3	320	40	8.9%	87.964514	79.85
Mexico	6	120	33	3.7%	132.277465	120.00
Kazakhstan	7	120	18	58.5%	335.433606	309.00
Uzbekistan	10	100	14	72.1%	399.037020	384.00
Indonesia	12	70	42	3.9%	86.641740	78.57

```
In [41]: # Resetting the index to access the column, data alignment and filling of any gaps. Then  
gold_data_df = gold_data_df.reset_index()  
gold_data_df
```

Out[41]:

	Country	Rank, 2022	Gold Production, 2022	Rank	Gold's share offorex reserves	Gold holdings(in tons) as of Jan2024	Gold Reserves as of September 2023
0	China	1	330	6	4.3%	2454.187905	2192.00
1	Russia	2	320	5	25.7%	2571.363692	2333.00
2	Australia	3	320	40	8.9%	87.964514	79.85
3	Mexico	6	120	33	3.7%	132.277465	120.00
4	Kazakhstan	7	120	18	58.5%	335.433606	309.00
5	Uzbekistan	10	100	14	72.1%	399.037020	384.00
6	Indonesia	12	70	42	3.9%	86.641740	78.57

```
In [42]: # Accessing the Country  
gold_data_df['Country']
```

```
Out[42]: 0      China  
1      Russia  
2    Australia  
3     Mexico  
4  Kazakhstan  
5  Uzbekistan  
6  Indonesia  
Name: Country, dtype: object
```

```
In [43]: # Conditional Indexing - this helps to focus on specific subsets of the data that are  
condition = gold_data_df['Gold Production, 2022'] > 200  
filtered_df = gold_data_df[condition]  
print(filtered_df)
```

```
      Country  Rank, 2022  Gold Production, 2022  Rank  \\\n0      China          1                  330      6  
1      Russia          2                  320      5  
2  Australia          3                  320     40  
  
      Gold's share offorex reserves  Gold holdings(in tons) as of Jan2024  \\\n0                      4.3%                2454.187905  
1                      25.7%               2571.363692  
2                      8.9%                87.964514  
  
      Gold Reserves as of September 2023  
0                      2192.00  
1                      2333.00  
2                      79.85
```

```
In [44]: # Multi-Level Indexing - this is useful in dealing with complex and hierachial data

gold_data_df = pd.merge(gold_production_2022, gold_holdings_Jan2024, on='Country', how='left')
gold_data_df = pd.merge(gold_data_df, gold_reserves_2023, on='Country', how='inner')

#Set a specific column as the index
gold_data_df.set_index(['Country', 'Rank, 2022'], inplace=True)
gold_data_df.fillna(' ', inplace=True)
gold_data_df.head(100)
```

Out[44]:

		Gold Production, 2022	Rank	Gold's share offorex reserves	Gold holdings(in tons) as of Jan2024	Gold Reserves as of September 2023
Country	Rank, 2022					
China	1	330	6	4.3%	2454.187905	2192.00
Russia	2	320	5	25.7%	2571.363692	2333.00
Australia	3	320	40	8.9%	87.964514	79.85
Mexico	6	120	33	3.7%	132.277465	120.00
Kazakhstan	7	120	18	58.5%	335.433606	309.00
Uzbekistan	10	100	14	72.1%	399.037020	384.00
Indonesia	12	70	42	3.9%	86.641740	78.57

Task 8 : Sorting - improved readability, easier for exploration and analyze data.

```
In [45]: # Sorting ascending per country
gold_data_df = gold_data_df.sort_values(by='Country', ascending=True)
gold_data_df.head()
```

Out[45]:

		Gold Production, 2022	Rank	Gold's share offorex reserves	Gold holdings(in tons) as of Jan2024	Gold Reserves as of September 2023
Country	Rank, 2022					
Australia	3	320	40	8.9%	87.964514	79.85
China	1	330	6	4.3%	2454.187905	2192.00
Indonesia	12	70	42	3.9%	86.641740	78.57
Kazakhstan	7	120	18	58.5%	335.433606	309.00
Mexico	6	120	33	3.7%	132.277465	120.00

In [46]: # Descending per country

```
gold_data_df = gold_data_df.sort_values(by='Country', ascending=False)
gold_data_df.head()
```

Out[46]:

Country	Rank, 2022	Gold Production, 2022	Rank	Gold's share offorex reserves	Gold holdings(in tons) as of Jan2024	Gold Reserves as of September 2023
Uzbekistan	10	100	14	72.1%	399.037020	384.00
Russia	2	320	5	25.7%	2571.363692	2333.00
Mexico	6	120	33	3.7%	132.277465	120.00
Kazakhstan	7	120	18	58.5%	335.433606	309.00
Indonesia	12	70	42	3.9%	86.641740	78.57

Task 9: Summary Statistics - to provide an overview of the characteristics of the dataset for data exploration and decision making using the "describe" function.

In [47]: # Summary statistics of inner join table -

```
gold_data_df = pd.merge(gold_production_2022, gold_holdings_Jan2024, on='Country', how='inner')
gold_data_df = pd.merge(gold_data_df, gold_reserves_2023, on='Country', how='inner')

summary_stats = gold_data_df.describe()
summary_stats.head()
```

Out[47]:

	Rank, 2022	Gold Production, 2022	Rank	Gold holdings(in tons) as of Jan2024	Gold Reserves as of September 2023
count	7.000000	7.000000	7.000000	7.000000	7.000000
mean	5.857143	197.142857	22.571429	866.700849	785.202857
std	4.140393	119.263614	15.640302	1131.462557	1016.680430
min	1.000000	70.000000	5.000000	86.641740	78.570000
25%	2.500000	110.000000	10.000000	110.120990	99.925000

Task 10 - to easily see the performance of the top gold producing countries. When dealing with large data, slicing will enable you to work on a small portion

In [48]: # Slicing by getting the top10

```
gold_data_df['Rank, 2022'] = pd.to_numeric(gold_data_df['Rank, 2022'], errors='coerce')
gold_data_df = gold_data_df.sort_values(by='Rank, 2022', ascending=True)
#Get first 10 from the list
first_three = gold_data_df[0:10]
first_three.head(10)
```

Out[48]:

	Rank, 2022	Country	Gold Production, 2022	Rank	Gold's share offorex reserves	Gold holdings(in tons) as of Jan2024	Gold Reserves as of September 2023
0	1	China	330	6	4.3%	2454.187905	2192.00
1	2	Russia	320	5	25.7%	2571.363692	2333.00
2	3	Australia	320	40	8.9%	87.964514	79.85
3	6	Mexico	120	33	3.7%	132.277465	120.00
4	7	Kazakhstan	120	18	58.5%	335.433606	309.00
5	10	Uzbekistan	100	14	72.1%	399.037020	384.00
6	12	Indonesia	70	42	3.9%	86.641740	78.57

In [49]: #used inner join to show only common data present to all 3 data frames

```
gold_data_df = pd.merge(gold_production_2022, gold_holdings_Jan2024, on='Country', how='inner')
gold_data_df = pd.merge(gold_data_df, gold_reserves_2023, on='Country', how='inner')
#Sort via Rank, 2022 descending to get the top 3 highest Gold Production, 2022
gold_data_df = gold_data_df.sort_values(by='Gold Production, 2022', ascending=False)
top = gold_data_df.iloc[0:3]
top.head(10)
```

Out[49]:

	Rank, 2022	Country	Gold Production, 2022	Rank	Gold's share offorex reserves	Gold holdings(in tons) as of Jan2024	Gold Reserves as of September 2023
0	1	China	330	6	4.3%	2454.187905	2192.00
1	2	Russia	320	5	25.7%	2571.363692	2333.00
2	3	Australia	320	40	8.9%	87.964514	79.85

In [50]: # Storing merged dataframe to csv

```
csv_file_path = 'gold_data_df.csv'
gold_data_df.to_csv(csv_file_path, index=False)
print(f'DataFrame has been saved to {csv_file_path}')
```

DataFrame has been saved to gold_data_df.csv

Task 11

In [51]: #importing data to MongoDB (database)

```
import pandas as pd
from pymongo import MongoClient

gold_data_df = pd.merge(gold_production_2022, gold_holdings_Jan2024, on='Country', how='inner')
gold_data_df = pd.merge(gold_data_df, gold_reserves_2023, on='Country', how='outer')
gold_data_df.fillna(' ', inplace=True)

client = MongoClient('mongodb://localhost:27017/')
db = client['Assessment1_612_DB']
collection = db['Outer_join']

data_dict = gold_data_df.to_dict(orient='records')

# Insert data into MongoDB
collection.insert_many(data_dict)
```



```
In [52]: gold_data_df = pd.merge(gold_production_2022, gold_holdings_Jan2024, on='Country', how='inner')
gold_data_df = pd.merge(gold_data_df, gold_reserves_2023, on='Country', how='inner')
gold_data_df.fillna(' ', inplace=True)

client = MongoClient('mongodb://localhost:27017/')
db = client['Assessment1_612_DB']
collection = db['Inner_join']

data_dict = gold_data_df.to_dict(orient='records')

# Insert data into MongoDB
collection.insert_many(data_dict)
```

```
Out[52]: InsertManyResult([ObjectId('65bc9513e2568c3f98d98dba'), ObjectId('65bc9513e2568c3f98d98dbb'), ObjectId('65bc9513e2568c3f98d98dbc'), ObjectId('65bc9513e2568c3f98d98dbd'), ObjectId('65bc9513e2568c3f98d98dbe'), ObjectId('65bc9513e2568c3f98d98dbf'), ObjectId('65bc9513e2568c3f98d98dc0')], acknowledged=True)
```

```
In [53]: import pandas as pd
from pymongo import MongoClient

gold_data_df = pd.merge(gold_production_2022, gold_holdings_Jan2024, on='Country', how='left')
gold_data_df = pd.merge(gold_data_df, gold_reserves_2023, on='Country', how='left')
gold_data_df.fillna(' ', inplace=True)

client = MongoClient('mongodb://localhost:27017/')
db = client['Assessment1_612_DB']
collection = db['Left_join']

data_dict = gold_data_df.to_dict(orient='records')

# Insert data into MongoDB
collection.insert_many(data_dict)
```

```
Out[53]: InsertManyResult([ObjectId('65bc9513e2568c3f98d98dc2'), ObjectId('65bc9513e2568c3f98d98dc3'), ObjectId('65bc9513e2568c3f98d98dc4'), ObjectId('65bc9513e2568c3f98d98dc5'), ObjectId('65bc9513e2568c3f98d98dc6'), ObjectId('65bc9513e2568c3f98d98dc7'), ObjectId('65bc9513e2568c3f98d98dc8'), ObjectId('65bc9513e2568c3f98d98dc9'), ObjectId('65bc9513e2568c3f98d98dca'), ObjectId('65bc9513e2568c3f98d98dcb'), ObjectId('65bc9513e2568c3f98d98dcc'), ObjectId('65bc9513e2568c3f98d98dcf')], acknowledged=True)
```

```
In [54]: import pandas as pd
from pymongo import MongoClient

gold_data_df = pd.merge(gold_production_2022, gold_holdings_Jan2024, on='Country', how='left')
gold_data_df = pd.merge(gold_data_df, gold_reserves_2023, on='Country', how='right')
gold_data_df.fillna(' ', inplace=True)

client = MongoClient('mongodb://localhost:27017/')
db = client['Assessment1_612_DB']
collection = db['Right_join']

data_dict = gold_data_df.to_dict(orient='records')

# Insert data into MongoDB
collection.insert_many(data_dict)
```

In []:

gold_data_df.csv

Rank, 2022	Country	Gold Productio n, 2022	Rank	Gold's share offorex reserves	Gold holdings(in tons) as of Jan2024	Gold Reserves as of Septembe r 2023
1 China		330	6	4.30%	2454.187904548581	2192
2 Russia		320	5	25.70%	2571.3636924813486	2333
3 Australia		320	40	8.90%	87.96451436533272	79.85
6 Mexico		120	33	3.70%	132.27746521102665	120
7 Kazakhstan		120	18	58.50%	335.43360553096176	309
10 Uzbekistan		100	14	72.10%	399.03702005326375	384
12 Indonesia		70	42	3.90%	86.64173971322245	78.57

gold_holdings_jan2024

Rank	Country	Gold's share offorex reserves	Gold holdings(in tons) as of Jan2024
1	United States	69.60%	8965.656360782377
2	Germany	68.70%	3695.611915554066
3	Italy	65.50%	2702.649076703293
4	France	67.10%	2686.2246247729236
5	Russia	25.70%	2571.3636924813486
6	China	4.30%	2454.187904548581
7	Switzerland	8.40%	1146.4046984955644
8	Japan	4.40%	932.5561297377378
9	India	8.60%	885.8180920298419
10	Netherlands	57.90%	675.1662286812818
11	Turkey	30.80%	575.9581297730118
12	Taiwan	4.70%	465.61667754281376
13	Portugal	72.90%	421.74465158115663
14	Uzbekistan	72.10%	399.03702005326375
15	Poland	12.60%	395.39938975996046
16	Saudi Arabia	4.70%	356.15707508068925
17	United Kingdom	11.60%	342.0474787915131
18	Kazakhstan	58.50%	335.43360553096176
19	Lebanon	56.50%	316.1431418543537
20	Spain	18.20%	310.4111183618759
21	Austria	59.40%	308.64741882572883
22	Thailand	7.30%	269.18464170443923
23	Singapore	4.30%	253.86250198416198
24	Belgium	36.80%	250.6657965748955
25	Algeria	13.90%	191.3613996719519
26	Philippines	10.70%	182.54290199121678
27	Venezuela	84.50%	177.69272826681245
28	Iraq	7.90%	146.27683027919363
29	Brazil	2.40%	142.96989364891795
30	Egypt	24.60%	138.89133847157797
31	Sweden	13.90%	138.56064480855042
32	South Africa	13.30%	138.22995114552285
33	Mexico	3.70%	132.27746521102665
34	Libya	9.40%	128.52960369671422
35	Greece	55.90%	126.10451683451208
36	South Korea	1.70%	115.08139473359319
37	Romania	9.50%	114.19954496551966
38	Qatar	12.70%	109.34937124111536
39	Hungary	13.80%	104.16850385368349
40	Australia	8.90%	87.96451436533272
41	Kuwait	10.30%	87.08266459725921
42	Indonesia	3.90%	86.64173971322245
43	United Arab Emirates	2.90%	82.23249087
44	Denmark	4.30%	73.30376197

45	Pakistan	32.80%	71.31959999
46	Argentina	18.80%	68.01266336266954
47	Belarus	42.90%	59.30439690294361
48	Finland	18.90%	54.01329829450255
49	Jordan	18.00%	47.95058113899716
50	Bolivia	87.50%	46.84826892890527

gold_production_2022

Rank, 2022	Country	Gold Production, 2022
1	China	330
2	Russia	320
3	Australia	320
4	Canada	220
5	United	170
6	Mexico	120
7	Kazakhstan	120
8	South	110
9	Peru	100
10	Uzbekistan	100
11	Ghana	90
12	Indonesia	70

gold_reserves_2023.csv

Country	Gold Reserves as of September 2023
United States	8133
Germany	3353
Italy	2452
France	2437
Russia	2333
China	2192
Switzerland	1040
Japan	846
India	801
Netherlands	612
Euro Area	507
Turkey	479
Taiwan	424
Uzbekistan	384
Portugal	383
Poland	334
Saudi Arabia	323
United Kingdom	310
Kazakhstan	309
Lebanon	287
Spain	282
Austria	280
Thailand	244
Singapore	230
Belgium	227
Algeria	174
Philippines	165
Venezuela	161
Libya	147
Iraq	133
Brazil	130
Egypt	126
Sweden	126
South Africa	125
Mexico	120
Greece	114
South Korea	104
Romania	104
Qatar	98.11
Hungary	94.49
Australia	79.85
Kuwait	78.97
Indonesia	78.57
United Arab Emirates	74.26

Denmark	66.55
Pakistan	64.66
Argentina	61.74
Belarus	53.85
Finland	49.02
Bolivia	42.51
Cambodia	42.49
Bulgaria	40.86
Malaysia	38.88
Serbia	38.68
Jordan	37.32
Peru	34.68
Ecuador	33.78
Slovakia	31.69
Ukraine	27.06
Syria	25.82
Czech Republic	25.07
Morocco	22.12
Afghanistan	21.87
Nigeria	21.37
Kyrgyzstan	21.02
Bangladesh	14.03
Cyprus	13.9
Mauritius	12.42
Ireland	12.04
Ghana	8.74
Tajikistan	8.46
Paraguay	8.19
Nepal	7.99
Myanmar	7.27
Guatemala	6.89
Macedonia	6.89
Tunisia	6.84
Latvia	6.66
Mongolia	6.61
Lithuania	5.82
Colombia	4.68
Bahrain	4.67
Mozambique	3.94
Albania	3.42
Slovenia	3.17
Aruba	3.11
Luxembourg	2.24
Hong Kong	2.08
Iceland	1.98
Trinidad And Tobago	1.95
Oman	1.88

Haiti	1.81
Yemen	1.56
Bosnia and Herzegovina	1.49
Suriname	1.46
El Salvador	1.37
Papua New Guinea	1.33
Honduras	0.69
Dominican Republic	0.57
Sri Lanka	0.47
Malawi	0.4
Malta	0.26
Chile	0.25
Estonia	0.25
Uruguay	0.1
Burundi	0.03
Fiji	0.03
Kenya	0.02

MongoDB Compass - localhost:27017/Assessment1_612_DB

Connect Edit View Help

localhost:27017 ...

My Queries Inner_join Assessment1_612_DB +

+ Create collection Refresh View Sort by Collection Name

Inner_join

Storage size: 4.10 kB Documents: 7 Avg. document size: 227.00 B Indexes: 1 Total index size: 4.10 kB

Left_join

Storage size: 4.10 kB Documents: 12 Avg. document size: 225.00 B Indexes: 1 Total index size: 4.10 kB

Outer_join

Storage size: 4.10 kB Documents: 111 Avg. document size: 228.00 B Indexes: 1 Total index size: 4.10 kB

Right_join

Storage size: 4.10 kB Documents: 108 Avg. document size: 228.00 B Indexes: 1 Total index size: 4.10 kB

Databases Assessment1_612_DB + Search

- Inner_join
- Left_join
- Outer_join
- Right_join
- Bijj_data
- Task_1_datasource
- admin
- config
- gold_db
- local
- sample_db

MONGOSH

MongoDB Compass - localhost:27017/Assessment1_612_DB.Right_join

Connect Edit View Collection Help

localhost:27017 ...

My Queries Right_join Assessment1_612_DB +

Assessment1_612_DB.Right_join

108 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or [Generate query](#) Explain Reset Find Options

+ ADD DATA EXPORT DATA 1 - 20 of 108

Document 1

```
_id: ObjectId('65bc9513e2568c3f98d98dcf')
Rank, 2022: ""
Country: "United States"
Gold Production, 2022: ""
Rank: 1
Gold's share of forex reserves: "69.6%"
Gold holdings(in tons) as of Jan2024: 8965.656360782377
Gold Reserves as of September 2023: 8133
```

Document 2

```
_id: ObjectId('65bc9513e2568c3f98d98dd0')
Rank, 2022: ""
Country: "Germany"
Gold Production, 2022: ""
Rank: 2
Gold's share of forex reserves: "68.7%"
Gold holdings(in tons) as of Jan2024: 8965.656360782377
Gold Reserves as of September 2023: 8133
```

Databases Assessment1_612_DB + Search

- Inner_join
- Left_join
- Outer_join
- Right_join
- Bijj_data
- Task_1_datasource
- admin
- config
- gold_db
- local

MongoDB Compass - localhost:27017/Assessment1_612_DB.Outer_join

Connect Edit View Collection Help

localhost:27017 ...

My Queries Outer_join Assessment1_612_DB +

Assessment1_612_DB.Outer_join

111 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or [Generate query](#) Explain Reset Find Options

ADD DATA EXPORT DATA 1 - 20 of 111

```
_id: ObjectId('65bc9513e2568c3f98d98d4a')
Rank, 2022: 1
Country: "China"
Gold Production, 2022: 330
Rank: 6
Gold's share offorex reserves: "4.3%"
Gold holdings(in tons) as of Jan2024: 2454.187904548581
Gold Reserves as of September 2023: 2192
```

```
_id: ObjectId('65bc9513e2568c3f98d98d4b')
Rank, 2022: 2
Country: "Russia"
Gold Production, 2022: 320
Rank: 5
Gold's share offorex reserves: "25.7%"
Gold holdings(in tons) as of Jan2024: 2571.3636924813486
```

My Queries Performance Databases Search Assessment1_612_DB Inner_join Left_join Outer_join Right_join Bijj_data Task_1_datasource admin config gold_db local sample_db

MongoDB Compass - localhost:27017/Assessment1_612_DB.Left_join

Connect Edit View Collection Help

localhost:27017 ...

My Queries Left_join Assessment1_612_DB +

Assessment1_612_DB.Left_join

12 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or [Generate query](#) Explain Reset Find Options

ADD DATA EXPORT DATA 1 - 12 of 12

```
_id: ObjectId('65bc9513e2568c3f98d98dc2')
Rank, 2022: 1
Country: "China"
Gold Production, 2022: 330
Rank: 6
Gold's share offorex reserves: "4.3%"
Gold holdings(in tons) as of Jan2024: 2454.187904548581
Gold Reserves as of September 2023: 2192
```

```
_id: ObjectId('65bc9513e2568c3f98d98dc3')
Rank, 2022: 2
Country: "Russia"
Gold Production, 2022: 320
Rank: 5
Gold's share offorex reserves: "25.7%"
Gold holdings(in tons) as of Jan2024: 2571.3636924813486
```

My Queries Performance Databases Search Assessment1_612_DB Inner_join Outer_join Right_join Bijj_data Task_1_datasource admin config gold_db local sample_db

MongoDB Compass - localhost:27017/Assessment1_612_DB.Inner_join

Connect Edit View Collection Help

localhost:27017 ...

My Queries Inner_join Assessment1_612_DB +

Assessment1_612_DB.Inner_join

7 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or Generate query Explain Reset Find Options

ADD DATA EXPORT DATA

1-7 of 7 < > ⌂ ⌂ ⌂

Document 1

```
_id: ObjectId('65bc9513e2568c3f98d98db')  
Rank, 2022: 1  
Country: "China"  
Gold Production, 2022: 330  
Rank: 6  
Gold's share offorex reserves: "4.3%"  
Gold holdings(in tons) as of Jan2024: 2454.187904548581  
Gold Reserves as of September 2023: 2192
```

Document 2

```
_id: ObjectId('65bc9513e2568c3f98d98db')  
Rank, 2022: 2  
Country: "Russia"  
Gold Production, 2022: 320  
Rank: 5  
Gold's share offorex reserves: "25.7%"  
Gold holdings(in tons) as of Jan2024: 2571.3636924813486
```

Databases Search Assessment1_612_DB Inner_join Left_join Outer_join Right_join Bjj_data Task_1_datasource admin config gold_db local sample_db

MonggoDB : Inner_join

<u>_id</u>	Rank, 2022	Country	Gold Production, 2022	Rank	Gold's share offorex reserves	Gold holdings(in tons) as of Jan2024	Gold Reserves as of September 2023
65bc2bb6b5d663728fe34c74	1	China	330	6	4.30%	2454.187904548581	2192
65bc2bb6b5d663728fe34c75	2	Russia	320	5	25.70%	2571.3636924813486	2333
65bc2bb6b5d663728fe34c76	3	Australia	320	40	8.90%	87.96451436533272	79.85
65bc2bb6b5d663728fe34c77	6	Mexico	120	33	3.70%	132.27746521102665	120
65bc2bb6b5d663728fe34c78	7	Kazakhstan	120	18	58.50%	335.43360553096176	309
65bc2bb6b5d663728fe34c79	10	Uzbekistan	100	14	72.10%	399.03702005326375	384
65bc2bb6b5d663728fe34c7a	12	Indonesia	70	42	3.90%	86.64173971322245	78.57

MonggoDB Left _join

_id	Rank, 2022	Country	Gold Production, 2022	Rank	Gold's share offorex reserves	Gold holdings(in tons) as of Jan2024	Gold Reserves as of September 2023
65bc2bddb5d663728fe34c7c	1	China	330	6	4.30%	2454.187904548581	2192
65bc2bddb5d663728fe34c7d	2	Russia	320	5	25.70%	2571.3636924813486	2333
65bc2bddb5d663728fe34c7e	3	Australia	320	40	8.90%	87.96451436533272	79.85
65bc2bddb5d663728fe34c7f	4	Canada	220				
65bc2bddb5d663728fe34c80	5	United	170				
65bc2bddb5d663728fe34c81	6	Mexico	120	33	3.70%	132.27746521102665	120
65bc2bddb5d663728fe34c82	7	Kazakhstan	120	18	58.50%	335.43360553096176	309
65bc2bddb5d663728fe34c83	8	South	110				
65bc2bddb5d663728fe34c84	9	Peru	100				34.68
65bc2bddb5d663728fe34c85	10	Uzbekistan	100	14	72.10%	399.03702005326375	384
65bc2bddb5d663728fe34c86	11	Ghana	90				8.74
65bc2bddb5d663728fe34c87	12	Indonesia	70	42	3.90%	86.64173971322245	78.57

MongoDB Outer_join

<u>_id</u>	Rank, 2022	Country	Gold Production, 2022	Rank	Gold's share offorex reserves	Gold holdings(in tons) as of Jan2024	Gold Reserves as of September 2023
65bc2c3ab5d663728fe34cf9	1	China	330	6	4.30%	2454.187904548581	2192
65bc2c3ab5d663728fe34cfa	2	Russia	320	5	25.70%	2571.3636924813486	2333
65bc2c3ab5d663728fe34cfb	3	Australia	320	40	8.90%	87.96451436533272	79.85
65bc2c3ab5d663728fe34fcf	4	Canada	220				
65bc2c3ab5d663728fe34cfd	5	United	170				
65bc2c3ab5d663728fe34fce	6	Mexico	120	33	3.70%	132.27746521102665	120
65bc2c3ab5d663728fe34cff	7	Kazakhstan	120	18	58.50%	335.43360553096176	309
65bc2c3ab5d663728fe34d00	8	South	110				
65bc2c3ab5d663728fe34d01	9	Peru	100				34.68
65bc2c3ab5d663728fe34d02	10	Uzbekistan	100	14	72.10%	399.03702005326375	384
65bc2c3ab5d663728fe34d03	11	Ghana	90				8.74
65bc2c3ab5d663728fe34d04	12	Indonesia	70	42	3.90%	86.64173971322245	78.57
65bc2c3ab5d663728fe34d05		United States		1	69.60%	8965.656360782377	8133
65bc2c3ab5d663728fe34d06		Germany		2	68.70%	3695.611915554066	3353
65bc2c3ab5d663728fe34d07		Italy		3	65.50%	2702.649076703293	2452
65bc2c3ab5d663728fe34d08		France		4	67.10%	2686.224624772924	2437
65bc2c3ab5d663728fe34d09		Switzerland		7	8.40%	1146.4046984955644	1040
65bc2c3ab5d663728fe34d0a		Japan		8	4.40%	932.5561297377378	846
65bc2c3ab5d663728fe34d0b		India		9	8.60%	885.8180920298419	801
65bc2c3ab5d663728fe34d0c		Netherlands		10	57.90%	675.1662286812818	612
65bc2c3ab5d663728fe34d0d		Turkey		11	30.80%	575.9581297730118	479
65bc2c3ab5d663728fe34d0e		Taiwan		12	4.70%	465.61667754281376	424
65bc2c3ab5d663728fe34d0f		Portugal		13	72.90%	421.74465158115663	383
65bc2c3ab5d663728fe34d10		Poland		15	12.60%	395.39938975996046	334
65bc2c3ab5d663728fe34d11		Saudi Arabia		16	4.70%	356.15707508068925	323
65bc2c3ab5d663728fe34d12		United Kingdom		17	11.60%	342.0474787915131	310
65bc2c3ab5d663728fe34d13		Lebanon		19	56.50%	316.1431418543537	287

65bc2c3ab5d663728fe34d14	Spain	20	18.20%	310.4111183618759	282
65bc2c3ab5d663728fe34d15	Austria	21	59.40%	308.64741882572883	280
65bc2c3ab5d663728fe34d16	Thailand	22	7.30%	269.18464170443923	244
65bc2c3ab5d663728fe34d17	Singapore	23	4.30%	253.862502	230
65bc2c3ab5d663728fe34d18	Belgium	24	36.80%	250.6657965748955	227
65bc2c3ab5d663728fe34d19	Algeria	25	13.90%	191.3613996719519	174
65bc2c3ab5d663728fe34d1a	Philippines	26	10.70%	182.5429019912168	165
65bc2c3ab5d663728fe34d1b	Venezuela	27	84.50%	177.69272826681245	161
65bc2c3ab5d663728fe34d1c	Iraq	28	7.90%	146.27683027919363	133
65bc2c3ab5d663728fe34d1d	Brazil	29	2.40%	142.96989364891795	130
65bc2c3ab5d663728fe34d1e	Egypt	30	24.60%	138.89133847157797	126
65bc2c3ab5d663728fe34d1f	Sweden	31	13.90%	138.56064480855042	126
65bc2c3ab5d663728fe34d20	South Africa	32	13.30%	138.22995114552285	125
65bc2c3ab5d663728fe34d21	Libya	34	9.40%	128.52960369671422	147
65bc2c3ab5d663728fe34d22	Greece	35	55.90%	126.10451683451208	114
65bc2c3ab5d663728fe34d23	South Korea	36	1.70%	115.0813947335932	104
65bc2c3ab5d663728fe34d24	Romania	37	9.50%	114.19954496551966	104
65bc2c3ab5d663728fe34d25	Qatar	38	12.70%	109.34937124111536	98.11
65bc2c3ab5d663728fe34d26	Hungary	39	13.80%	104.16850385368348	94.49
65bc2c3ab5d663728fe34d27	Kuwait	41	10.30%	87.08266459725921	78.97
65bc2c3ab5d663728fe34d28	United Arab Emirates	43	2.90%	82.23249087	74.26
65bc2c3ab5d663728fe34d29	Denmark	44	4.30%	73.30376197	66.55
65bc2c3ab5d663728fe34d2a	Pakistan	45	32.80%	71.31959999	64.66
65bc2c3ab5d663728fe34d2b	Argentina	46	18.80%	68.01266336266954	61.74
65bc2c3ab5d663728fe34d2c	Belarus	47	42.90%	59.30439690294361	53.85
65bc2c3ab5d663728fe34d2d	Finland	48	18.90%	54.01329829450255	49.02
65bc2c3ab5d663728fe34d2e	Jordan	49	18.00%	47.95058113899716	37.32
65bc2c3ab5d663728fe34d2f	Bolivia	50	87.50%	46.84826892890527	42.51
65bc2c3ab5d663728fe34d30	Euro Area				507
65bc2c3ab5d663728fe34d31	Cambodia				42.49
65bc2c3ab5d663728fe34d32	Bulgaria				40.86
65bc2c3ab5d663728fe34d33	Malaysia				38.88
65bc2c3ab5d663728fe34d34	Serbia				38.68
65bc2c3ab5d663728fe34d35	Ecuador				33.78

65bc2c3ab5d663728fe34d36	Slovakia	31.69
65bc2c3ab5d663728fe34d37	Ukraine	27.06
65bc2c3ab5d663728fe34d38	Syria	25.82
65bc2c3ab5d663728fe34d39	Czech Republic	25.07
65bc2c3ab5d663728fe34d3a	Morocco	22.12
65bc2c3ab5d663728fe34d3b	Afghanistan	21.87
65bc2c3ab5d663728fe34d3c	Nigeria	21.37
65bc2c3ab5d663728fe34d3d	Kyrgyzstan	21.02
65bc2c3ab5d663728fe34d3e	Bangladesh	14.03
65bc2c3ab5d663728fe34d3f	Cyprus	13.9
65bc2c3ab5d663728fe34d40	Mauritius	12.42
65bc2c3ab5d663728fe34d41	Ireland	12.04
65bc2c3ab5d663728fe34d42	Tajikistan	8.46
65bc2c3ab5d663728fe34d43	Paraguay	8.19
65bc2c3ab5d663728fe34d44	Nepal	7.99
65bc2c3ab5d663728fe34d45	Myanmar	7.27
65bc2c3ab5d663728fe34d46	Guatemala	6.89
65bc2c3ab5d663728fe34d47	Macedonia	6.89
65bc2c3ab5d663728fe34d48	Tunisia	6.84
65bc2c3ab5d663728fe34d49	Latvia	6.66
65bc2c3ab5d663728fe34d4a	Mongolia	6.61
65bc2c3ab5d663728fe34d4b	Lithuania	5.82
65bc2c3ab5d663728fe34d4c	Colombia	4.68
65bc2c3ab5d663728fe34d4d	Bahrain	4.67
65bc2c3ab5d663728fe34d4e	Mozambique	3.94
65bc2c3ab5d663728fe34d4f	Albania	3.42
65bc2c3ab5d663728fe34d50	Slovenia	3.17
65bc2c3ab5d663728fe34d51	Aruba	3.11
65bc2c3ab5d663728fe34d52	Luxembourg	2.24
65bc2c3ab5d663728fe34d53	Hong Kong	2.08
65bc2c3ab5d663728fe34d54	Iceland	1.98
65bc2c3ab5d663728fe34d55	Trinidad And Tobago	1.95
65bc2c3ab5d663728fe34d56	Oman	1.88
65bc2c3ab5d663728fe34d57	Haiti	1.81

65bc2c3ab5d663728fe34d58	Yemen	1.56
65bc2c3ab5d663728fe34d59	Bosnia and Herzegovina	1.49
65bc2c3ab5d663728fe34d5a	Suriname	1.46
65bc2c3ab5d663728fe34d5b	El Salvador	1.37
65bc2c3ab5d663728fe34d5c	Papua New Guinea	1.33
65bc2c3ab5d663728fe34d5d	Honduras	0.69
65bc2c3ab5d663728fe34d5e	Dominican Republic	0.57
65bc2c3ab5d663728fe34d5f	Sri Lanka	0.47
65bc2c3ab5d663728fe34d60	Malawi	0.4
65bc2c3ab5d663728fe34d61	Malta	0.26
65bc2c3ab5d663728fe34d62	Chile	0.25
65bc2c3ab5d663728fe34d63	Estonia	0.25
65bc2c3ab5d663728fe34d64	Uruguay	0.1
65bc2c3ab5d663728fe34d65	Burundi	0.03
65bc2c3ab5d663728fe34d66	Fiji	0.03
65bc2c3ab5d663728fe34d67	Kenya	0.02

MongoDB Right_join

_id	Rank, 2022	Country	Gold Productio n, 2022	Rank	Gold's	Gold holdings(in tons) as of Jan2024	Gold Reserves as of September 2023
					share offorex reserves		
65bc2c65b5d663728fe34d69		United States		1	69.60%	8965.656360782377	8133
65bc2c65b5d663728fe34d6a		Germany		2	68.70%	3695.611915554066	3353
65bc2c65b5d663728fe34d6b		Italy		3	65.50%	2702.649076703293	2452
65bc2c65b5d663728fe34d6c		France		4	67.10%	2686.224624772924	2437
65bc2c65b5d663728fe34d6d	2	Russia	320	5	25.70%	2571.3636924813486	2333
65bc2c65b5d663728fe34d6e	1	China	330	6	4.30%	2454.187904548581	2192
65bc2c65b5d663728fe34d6f		Switzerland		7	8.40%	1146.4046984955644	1040
65bc2c65b5d663728fe34d70		Japan		8	4.40%	932.5561297377378	846
65bc2c65b5d663728fe34d71		India		9	8.60%	885.8180920298419	801
65bc2c65b5d663728fe34d72		Netherlands		10	57.90%	675.1662286812818	612
65bc2c65b5d663728fe34d73		Euro Area					507
65bc2c65b5d663728fe34d74		Turkey		11	30.80%	575.9581297730118	479
65bc2c65b5d663728fe34d75		Taiwan		12	4.70%	465.61667754281376	424
65bc2c65b5d663728fe34d76	10	Uzbekistan	100	14	72.10%	399.03702005326375	384
65bc2c65b5d663728fe34d77		Portugal		13	72.90%	421.74465158115663	383
65bc2c65b5d663728fe34d78		Poland		15	12.60%	395.39938975996046	334
65bc2c65b5d663728fe34d79		Saudi Arabia		16	4.70%	356.15707508068925	323
65bc2c65b5d663728fe34d7a		United Kingdom		17	11.60%	342.0474787915131	310
65bc2c65b5d663728fe34d7b	7	Kazakhstan	120	18	58.50%	335.43360553096176	309
65bc2c65b5d663728fe34d7c		Lebanon		19	56.50%	316.1431418543537	287
65bc2c65b5d663728fe34d7d		Spain		20	18.20%	310.4111183618759	282
65bc2c65b5d663728fe34d7e		Austria		21	59.40%	308.64741882572883	280
65bc2c65b5d663728fe34d7f		Thailand		22	7.30%	269.18464170443923	244
65bc2c65b5d663728fe34d80		Singapore		23	4.30%	253.862502	230
65bc2c65b5d663728fe34d81		Belgium		24	36.80%	250.6657965748955	227
65bc2c65b5d663728fe34d82		Algeria		25	13.90%	191.3613996719519	174
65bc2c65b5d663728fe34d83		Philippines		26	10.70%	182.5429019912168	165

65bc2c65b5d663728fe34d84	Venezuela	27	84.50%	177.69272826681245	161
65bc2c65b5d663728fe34d85	Libya	34	9.40%	128.52960369671422	147
65bc2c65b5d663728fe34d86	Iraq	28	7.90%	146.27683027919363	133
65bc2c65b5d663728fe34d87	Brazil	29	2.40%	142.96989364891795	130
65bc2c65b5d663728fe34d88	Egypt	30	24.60%	138.89133847157797	126
65bc2c65b5d663728fe34d89	Sweden	31	13.90%	138.56064480855042	126
65bc2c65b5d663728fe34d8a	South Africa	32	13.30%	138.22995114552285	125
65bc2c65b5d663728fe34d8b	6 Mexico	120	3.70%	132.27746521102665	120
65bc2c65b5d663728fe34d8c	Greece	35	55.90%	126.10451683451208	114
65bc2c65b5d663728fe34d8d	South Korea	36	1.70%	115.0813947335932	104
65bc2c65b5d663728fe34d8e	Romania	37	9.50%	114.19954496551966	104
65bc2c65b5d663728fe34d8f	Qatar	38	12.70%	109.34937124111536	98.11
65bc2c65b5d663728fe34d90	Hungary	39	13.80%	104.16850385368348	94.49
65bc2c65b5d663728fe34d91	3 Australia	320	8.90%	87.96451436533272	79.85
65bc2c65b5d663728fe34d92	Kuwait	41	10.30%	87.08266459725921	78.97
65bc2c65b5d663728fe34d93	12 Indonesia	70	4.90%	86.64173971322245	78.57
65bc2c65b5d663728fe34d94	United Arab Emirates	43	2.90%	82.23249087	74.26
65bc2c65b5d663728fe34d95	Denmark	44	4.30%	73.30376197	66.55
65bc2c65b5d663728fe34d96	Pakistan	45	32.80%	71.31959999	64.66
65bc2c65b5d663728fe34d97	Argentina	46	18.80%	68.01266336266954	61.74
65bc2c65b5d663728fe34d98	Belarus	47	42.90%	59.30439690294361	53.85
65bc2c65b5d663728fe34d99	Finland	48	18.90%	54.01329829450255	49.02
65bc2c65b5d663728fe34d9a	Bolivia	50	87.50%	46.84826892890527	42.51
65bc2c65b5d663728fe34d9b	Cambodia				42.49
65bc2c65b5d663728fe34d9c	Bulgaria				40.86
65bc2c65b5d663728fe34d9d	Malaysia				38.88
65bc2c65b5d663728fe34d9e	Serbia				38.68
65bc2c65b5d663728fe34d9f	Jordan	49	18.00%	47.95058113899716	37.32
65bc2c65b5d663728fe34da0	Peru				34.68
65bc2c65b5d663728fe34da1	Ecuador				33.78
65bc2c65b5d663728fe34da2	Slovakia				31.69
65bc2c65b5d663728fe34da3	Ukraine				27.06
65bc2c65b5d663728fe34da4	Syria				25.82
65bc2c65b5d663728fe34da5	Czech Republic				25.07

65bc2c65b5d663728fe34da6	Morocco	22.12
65bc2c65b5d663728fe34da7	Afghanistan	21.87
65bc2c65b5d663728fe34da8	Nigeria	21.37
65bc2c65b5d663728fe34da9	Kyrgyzstan	21.02
65bc2c65b5d663728fe34daa	Bangladesh	14.03
65bc2c65b5d663728fe34dab	Cyprus	13.9
65bc2c65b5d663728fe34dac	Mauritius	12.42
65bc2c65b5d663728fe34dad	Ireland	12.04
65bc2c65b5d663728fe34dae	Ghana	8.74
65bc2c65b5d663728fe34daf	Tajikistan	8.46
65bc2c65b5d663728fe34db0	Paraguay	8.19
65bc2c65b5d663728fe34db1	Nepal	7.99
65bc2c65b5d663728fe34db2	Myanmar	7.27
65bc2c65b5d663728fe34db3	Guatemala	6.89
65bc2c65b5d663728fe34db4	Macedonia	6.89
65bc2c65b5d663728fe34db5	Tunisia	6.84
65bc2c65b5d663728fe34db6	Latvia	6.66
65bc2c65b5d663728fe34db7	Mongolia	6.61
65bc2c65b5d663728fe34db8	Lithuania	5.82
65bc2c65b5d663728fe34db9	Colombia	4.68
65bc2c65b5d663728fe34dba	Bahrain	4.67
65bc2c65b5d663728fe34dbb	Mozambique	3.94
65bc2c65b5d663728fe34dbc	Albania	3.42
65bc2c65b5d663728fe34dbd	Slovenia	3.17
65bc2c65b5d663728fe34dbe	Aruba	3.11
65bc2c65b5d663728fe34dbf	Luxembourg	2.24
65bc2c65b5d663728fe34dc0	Hong Kong	2.08
65bc2c65b5d663728fe34dc1	Iceland	1.98
65bc2c65b5d663728fe34dc2	Trinidad And Tobago	1.95
65bc2c65b5d663728fe34dc3	Oman	1.88
65bc2c65b5d663728fe34dc4	Haiti	1.81
65bc2c65b5d663728fe34dc5	Yemen	1.56
65bc2c65b5d663728fe34dc6	Bosnia and Herzegovina	1.49
65bc2c65b5d663728fe34dc7	Suriname	1.46

65bc2c65b5d663728fe34dc8	El Salvador	1.37
65bc2c65b5d663728fe34dc9	Papua New Guinea	1.33
65bc2c65b5d663728fe34dca	Honduras	0.69
65bc2c65b5d663728fe34dcb	Dominican Republic	0.57
65bc2c65b5d663728fe34dcc	Sri Lanka	0.47
65bc2c65b5d663728fe34dcd	Malawi	0.4
65bc2c65b5d663728fe34dce	Malta	0.26
65bc2c65b5d663728fe34dcf	Chile	0.25
65bc2c65b5d663728fe34dd0	Estonia	0.25
65bc2c65b5d663728fe34dd1	Uruguay	0.1
65bc2c65b5d663728fe34dd2	Burundi	0.03
65bc2c65b5d663728fe34dd3	Fiji	0.03
65bc2c65b5d663728fe34dd4	Kenya	0.02