In [1]:
```
import matplotlib.pyplot as plt
%matplotlib inline
import pandas as pd
import numpy as np
import seaborn as sns
```

ANALYSIS OF MTN UGANDA'S BALANCE SHEET 2023

In [ ]:
```
df = pd.read_csv('MTN BALANCE SHEET 2024.csv', delimiter=',')
```

In [207…
```
df_cleaned = df[df['Items'].notna()]
```

After exporting the file with the information, the Items column is parsed through to sieve through any commas and special characters that may interfere with the data

In [208…
```
df_cleaned.columns
```

Out[208]:
```
Index(['Items', '2023', '2022'], dtype='object')
```

Checking to confirm the columns in the dataset

In [209…
```
df_cleaned['2022']=pd.to_numeric(df_cleaned['2022'], errors = 'coerce')
invalid = df_cleaned[df_cleaned['2022'].isnull()]
print(invalid)
```

```
                      Items  2023  2022
0                    Assets   NaN   NaN
1        Non-current assets   NaN   NaN
9            Current assets   NaN   NaN
19                   Equity   NaN   NaN
23              Liabilities   NaN   NaN
24  Non-current liabilities   NaN   NaN
31      Current liabilities   NaN   NaN
C:\Users\jbmad\AppData\Local\Temp\ipykernel_12320\1500303887.py:1: SettingWithCopyWar
ning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  df_cleaned['2022']=pd.to_numeric(df_cleaned['2022'], errors = 'coerce')
```

Here on the first line, I converted the '2022' column to an appropriate class so that i can use the values in that colum for calculation as it was recognized as a string. secondly,I printed out the rows within the columns that have no values attached to them, these rows in the dataset serve as headings hence why they dont have any numerical value attached.

In [210…
```
df_cleaned.head(45)
```

Out[210]:

	Items	2023	2022
0	Assets	NaN	NaN
1	Non-current assets	NaN	NaN
2	Property, plant, and equipment	1.086548e+09	9.491893e+08
3	Right-of-use assets	1.091714e+09	9.493578e+08
4	Intangible assets	4.296368e+08	3.577166e+08
5	Deferred Tax assets	2.160931e+07	1.461530e+07
6	Contract assets	2.342408e+07	1.070243e+07
7	Receivables and prepayments	6.655294e+07	5.587604e+07
9	Current assets	NaN	NaN
10	Inventories	1.274521e+07	2.743244e+07
11	Current Investments	1.226500e+07	0.000000e+00
12	Current income tax recoverable	1.976045e+06	4.292700e+05
13	Contract assets	2.171696e+07	1.058507e+07
14	Trade and other receivables	1.872431e+08	1.853633e+08
15	Mobile money deposits	1.488547e+09	1.207758e+09
16	Cash and cash equivalents	2.385629e+08	2.007727e+08
18	Total Assets	4.682540e+09	3.969799e+09
19	Equity	NaN	NaN
20	Ordinary share capital	2.238904e+07	2.238904e+07
21	Retained earnings	9.918299e+08	8.816085e+08
22	Total Equity	1.014219e+09	9.039976e+08
23	Liabilities	NaN	NaN
24	Non-current liabilities	NaN	NaN
25	Borrowings	1.765155e+07	8.289739e+07
26	Lease liabilities	1.107021e+09	9.658918e+08
27	Other financial liability	9.744664e+07	0.000000e+00
28	Contract liabilities	1.239543e+07	1.221504e+07
29	Employee share-based payment liability	1.013507e+07	1.951361e+07
31	Current liabilities	NaN	NaN
32	Trade and other payables	5.100528e+08	4.604305e+08
33	Other financial liability	2.419239e+07	0.000000e+00
34	Contract liabilities	3.196024e+07	1.650762e+07
35	Current income tax payable	2.534440e+06	4.323181e+06
36	Borrowings	1.847363e+08	1.666756e+08

	Items	2023	2022
37	Lease liabilities	1.497282e+08	1.065951e+08
38	Mobile money deposits	1.488547e+09	1.207758e+09
39	Employee share-based payment liability	4.629720e+06	5.446593e+06
40	Provisions	2.729119e+07	1.754629e+07
42	Total liabilities	3.668322e+09	3.065801e+09
43	Total equity and liabilities	4.682540e+09	3.969799e+09

Printed out the whole dataset to check if there are any irregularities so far, which there dont seem to be any.

In [211…
```
df_cleaned.describe()
```

Out[211]:

	2023	2022
count	3.300000e+01	3.300000e+01
mean	7.094758e+08	6.014846e+08
std	1.266793e+09	1.070813e+09
min	1.976045e+06	0.000000e+00
25%	2.160931e+07	1.221504e+07
50%	9.744664e+07	8.289739e+07
75%	1.014219e+09	9.039976e+08
max	4.682540e+09	3.969799e+09

An overview of the dataset in terms of frequency, median etc

In [212…
```
df_cleaned.dtypes
```

Out[212]:
```
Items     object
2023     float64
2022     float64
dtype: object
```

In [213…
```
df_cleaned['change'] = df_cleaned['2023']-df_cleaned['2022']
print(df_cleaned['change'])
```

```
0             NaN
1             NaN
2     137358350.0
3     142356038.0
4      71920188.0
5       6994010.0
6      12721654.0
7      10676901.0
9             NaN
10    -14687230.0
11     12265000.0
12      1546775.0
13     11131892.0
14      1879794.0
15    280788270.0
16     37790218.0
18    712741860.0
19            NaN
20            0.0
21    110221346.0
22    110221346.0
23            NaN
24            NaN
25    -65245845.0
26    141129177.0
27     97446644.0
28       180383.0
29     -9378540.0
31            NaN
32     49622300.0
33     24192394.0
34     15452624.0
35     -1788741.0
36     18060688.0
37     43133133.0
38    280788270.0
39      -816873.0
40      9744900.0
42    602520514.0
43    712741860.0
Name: change, dtype: float64
```

```
C:\Users\jbmad\AppData\Local\Temp\ipykernel_12320\2764789231.py:1: SettingWithCopyWar
ning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  df_cleaned['change'] = df_cleaned['2023']-df_cleaned['2022']
```

Creating a new Column to show the changes in amounts from 2022 to 2023

In [214…

```
df_cleaned.head(45)
```

Out[214]:

	Items	2023	2022	change
0	Assets	NaN	NaN	NaN
1	Non-current assets	NaN	NaN	NaN
2	Property, plant, and equipment	1.086548e+09	9.491893e+08	137358350.0
3	Right-of-use assets	1.091714e+09	9.493578e+08	142356038.0
4	Intangible assets	4.296368e+08	3.577166e+08	71920188.0
5	Deferred Tax assets	2.160931e+07	1.461530e+07	6994010.0
6	Contract assets	2.342408e+07	1.070243e+07	12721654.0
7	Receivables and prepayments	6.655294e+07	5.587604e+07	10676901.0
9	Current assets	NaN	NaN	NaN
10	Inventories	1.274521e+07	2.743244e+07	-14687230.0
11	Current Investments	1.226500e+07	0.000000e+00	12265000.0
12	Current income tax recoverable	1.976045e+06	4.292700e+05	1546775.0
13	Contract assets	2.171696e+07	1.058507e+07	11131892.0
14	Trade and other receivables	1.872431e+08	1.853633e+08	1879794.0
15	Mobile money deposits	1.488547e+09	1.207758e+09	280788270.0
16	Cash and cash equivalents	2.385629e+08	2.007727e+08	37790218.0
18	Total Assets	4.682540e+09	3.969799e+09	712741860.0
19	Equity	NaN	NaN	NaN
20	Ordinary share capital	2.238904e+07	2.238904e+07	0.0
21	Retained earnings	9.918299e+08	8.816085e+08	110221346.0
22	Total Equity	1.014219e+09	9.039976e+08	110221346.0
23	Liabilities	NaN	NaN	NaN
24	Non-current liabilities	NaN	NaN	NaN
25	Borrowings	1.765155e+07	8.289739e+07	-65245845.0
26	Lease liabilities	1.107021e+09	9.658918e+08	141129177.0
27	Other financial liability	9.744664e+07	0.000000e+00	97446644.0
28	Contract liabilities	1.239543e+07	1.221504e+07	180383.0
29	Employee share-based payment liability	1.013507e+07	1.951361e+07	-9378540.0
31	Current liabilities	NaN	NaN	NaN
32	Trade and other payables	5.100528e+08	4.604305e+08	49622300.0
33	Other financial liability	2.419239e+07	0.000000e+00	24192394.0
34	Contract liabilities	3.196024e+07	1.650762e+07	15452624.0
35	Current income tax payable	2.534440e+06	4.323181e+06	-1788741.0
36	Borrowings	1.847363e+08	1.666756e+08	18060688.0

	Items	2023	2022	change
37	Lease liabilities	1.497282e+08	1.065951e+08	43133133.0
38	Mobile money deposits	1.488547e+09	1.207758e+09	280788270.0
39	Employee share-based payment liability	4.629720e+06	5.446593e+06	-816873.0
40	Provisions	2.729119e+07	1.754629e+07	9744900.0
42	Total liabilities	3.668322e+09	3.065801e+09	602520514.0
43	Total equity and liabilities	4.682540e+09	3.969799e+09	712741860.0

In [215…

```
df_cleaned['%change'] = df_cleaned['change']/df_cleaned['2022']*100
print(df_cleaned['%change'])
```

```
0            NaN
1            NaN
2      14.471123
3      14.994982
4      20.105355
5      47.854023
6     118.866990
7      19.108191
9            NaN
10    -53.539647
11           inf
12    360.326834
13    105.165994
14      1.014114
15     23.248711
16     18.822387
18     17.954106
19           NaN
20      0.000000
21     12.502301
22     12.192660
23           NaN
24           NaN
25    -78.706753
26     14.611282
27           inf
28      1.476728
29    -48.061525
31           NaN
32     10.777371
33           inf
34     93.609065
35    -41.375575
36     10.835834
37     40.464471
38     23.248711
39    -14.997871
40     55.538230
42     19.652955
43     17.954106
Name: %change, dtype: float64
```

```
C:\Users\jbmad\AppData\Local\Temp\ipykernel_12320\514583357.py:1: SettingWithCopyWarn
ing:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  df_cleaned['%change'] = df_cleaned['change']/df_cleaned['2022']*100
```

Creating a new column to show the percentage change from 2022 to 2023

In [233…

```
df_cleaned.head(45)
```

Out[233]:

	Items	2023	2022	change	%change
0	Assets	NaN	NaN	NaN	NaN
1	Non-current assets	NaN	NaN	NaN	NaN
2	Property, plant, and equipment	1.086548e+09	9.491893e+08	137358350.0	14.471123
3	Right-of-use assets	1.091714e+09	9.493578e+08	142356038.0	14.994982
4	Intangible assets	4.296368e+08	3.577166e+08	71920188.0	20.105355
5	Deferred Tax assets	2.160931e+07	1.461530e+07	6994010.0	47.854023
6	Contract assets	2.342408e+07	1.070243e+07	12721654.0	118.866990
7	Receivables and prepayments	6.655294e+07	5.587604e+07	10676901.0	19.108191
9	Current assets	NaN	NaN	NaN	NaN
10	Inventories	1.274521e+07	2.743244e+07	-14687230.0	-53.539647
11	Current Investments	1.226500e+07	0.000000e+00	12265000.0	inf
12	Current income tax recoverable	1.976045e+06	4.292700e+05	1546775.0	360.326834
13	Contract assets	2.171696e+07	1.058507e+07	11131892.0	105.165994
14	Trade and other receivables	1.872431e+08	1.853633e+08	1879794.0	1.014114
15	Mobile money deposits	1.488547e+09	1.207758e+09	280788270.0	23.248711
16	Cash and cash equivalents	2.385629e+08	2.007727e+08	37790218.0	18.822387
18	Total Assets	4.682540e+09	3.969799e+09	712741860.0	17.954106
19	Equity	NaN	NaN	NaN	NaN
20	Ordinary share capital	2.238904e+07	2.238904e+07	0.0	0.000000
21	Retained earnings	9.918299e+08	8.816085e+08	110221346.0	12.502301
22	Total Equity	1.014219e+09	9.039976e+08	110221346.0	12.192660
23	Liabilities	NaN	NaN	NaN	NaN
24	Non-current liabilities	NaN	NaN	NaN	NaN
25	Borrowings	1.765155e+07	8.289739e+07	-65245845.0	-78.706753
26	Lease liabilities	1.107021e+09	9.658918e+08	141129177.0	14.611282
27	Other financial liability	9.744664e+07	0.000000e+00	97446644.0	inf
28	Contract liabilities	1.239543e+07	1.221504e+07	180383.0	1.476728
29	Employee share-based payment liability	1.013507e+07	1.951361e+07	-9378540.0	-48.061525
31	Current liabilities	NaN	NaN	NaN	NaN
32	Trade and other payables	5.100528e+08	4.604305e+08	49622300.0	10.777371
33	Other financial liability	2.419239e+07	0.000000e+00	24192394.0	inf
34	Contract liabilities	3.196024e+07	1.650762e+07	15452624.0	93.609065
35	Current income tax payable	2.534440e+06	4.323181e+06	-1788741.0	-41.375575
36	Borrowings	1.847363e+08	1.666756e+08	18060688.0	10.835834

	Items	2023	2022	change	%change
37	Lease liabilities	1.497282e+08	1.065951e+08	43133133.0	40.464471
38	Mobile money deposits	1.488547e+09	1.207758e+09	280788270.0	23.248711
39	Employee share-based payment liability	4.629720e+06	5.446593e+06	-816873.0	-14.997871
40	Provisions	2.729119e+07	1.754629e+07	9744900.0	55.538230
42	Total liabilities	3.668322e+09	3.065801e+09	602520514.0	19.652955
43	Total equity and liabilities	4.682540e+09	3.969799e+09	712741860.0	17.954106

Calculating the financial ratios

In [216…
```
total_assets = 4682540474
current_assets = 1963055901
current_liabilities = 2423671991
inventories = 12745207
total_liabilities = 3668321575
equity = 1014218899
net_income = 493077000
revenue = 2669146000
```

In [217…
```
Current_Ratio = current_assets/current_liabilities
print(Current_Ratio)
```

```
0.80995114367355
```

In [218…
```
Quick_Ratio = (current_assets-inventories)/current_liabilities
print(Quick_Ratio)
```

```
0.8046925084096497
```

In [219…
```
Debt_to_Equity_ratio = total_liabilities/equity
print(Debt_to_Equity_ratio)
```

```
3.6168933339902196
```

In [220…
```
Debt_to_Asset_Ratio = total_liabilities/total_assets
print(Debt_to_Asset_Ratio)
```

```
0.7834041361454338
```

In [221…
```
Net_profit_margin= net_income/revenue
print(Net_profit_margin)
```

```
0.18473212031114072
```

In [222…
```
Asset_Return = net_income/total_assets
print(Asset_Return)
```

```
0.10530117203211173
```

In [223…
```
Equity_Return = net_income/equity
print(Equity_Return)
```

```
0.48616427921641403
```

In [224…
```
asset_turnover = revenue/total_assets
print(asset_turnover)
```

```
0.5700209138224311
```

In [225…
```
print(df_cleaned.columns)
```

```
Index(['Items', '2023', '2022', 'change', '%change'], dtype='object')
```

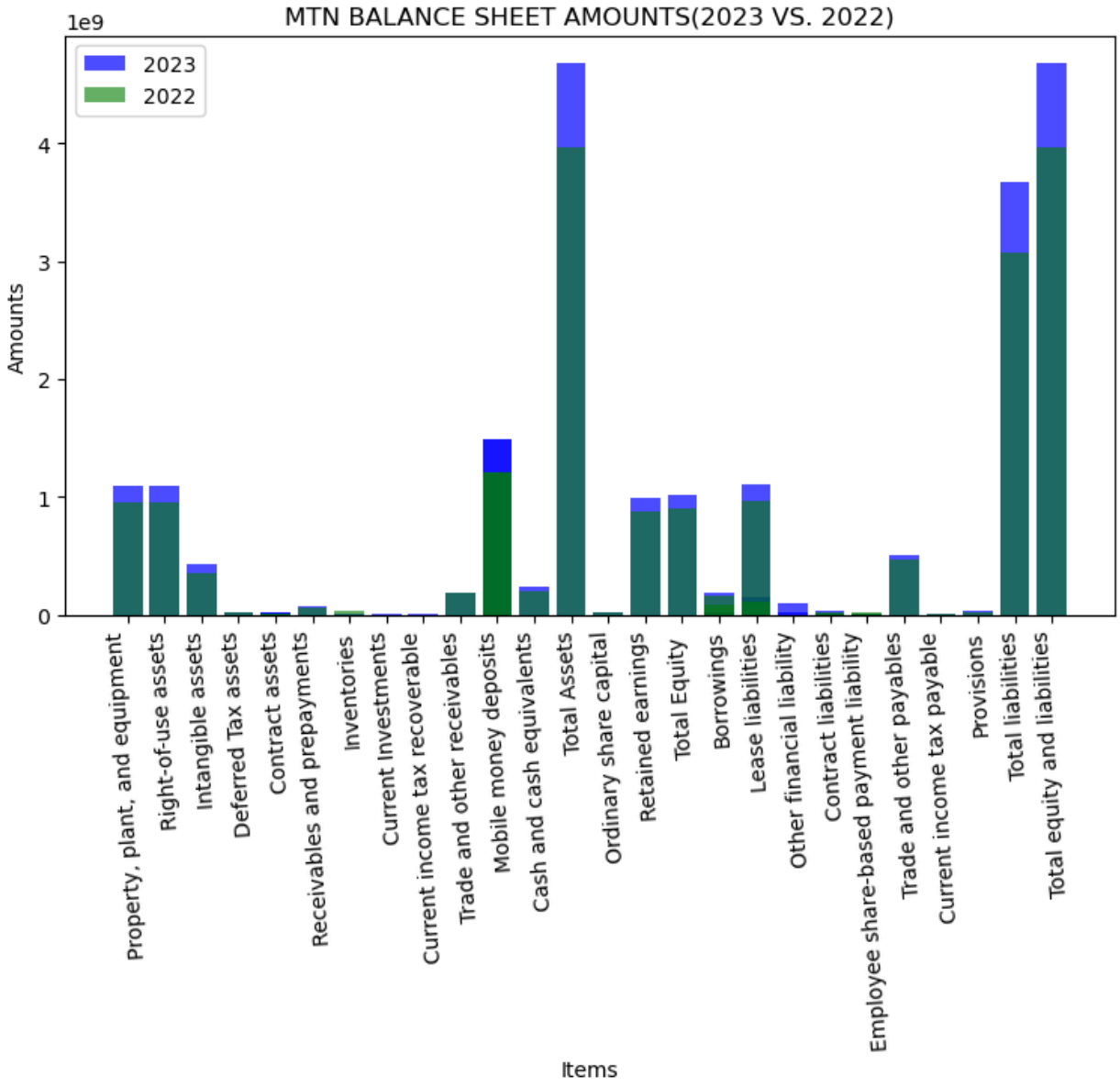
In [226…
```
df_cleaned.dtypes
```

Out[226]:
```
Items       object
2023       float64
2022       float64
change     float64
%change    float64
dtype: object
```

In [240…
```
rows_to_exclude = ['Assets','Non-current assets','Current assets','Equity','Liabilitie
df2 =df_cleaned[~df['Items'].isin(rows_to_exclude)]
plt.figure(figsize=(9,5))
plt.bar(df2['Items'],df2['2023'], label='2023', color = 'blue', alpha = 0.7)
plt.bar(df2['Items'],df2['2022'], label = '2022', color = 'green', alpha = 0.6)
plt.xlabel('Items')
plt.ylabel('Amounts')
plt.title('MTN BALANCE SHEET AMOUNTS(2023 VS. 2022)')
plt.xticks(rotation=94)
plt.legend()
```

```
C:\Users\jbmad\AppData\Local\Temp\ipykernel_12320\103881079.py:2: UserWarning: Boolea
n Series key will be reindexed to match DataFrame index.
  df2 =df_cleaned[~df['Items'].isin(rows_to_exclude)]
```

Out[240]:
```
<matplotlib.legend.Legend at 0x23c24eea910>
```



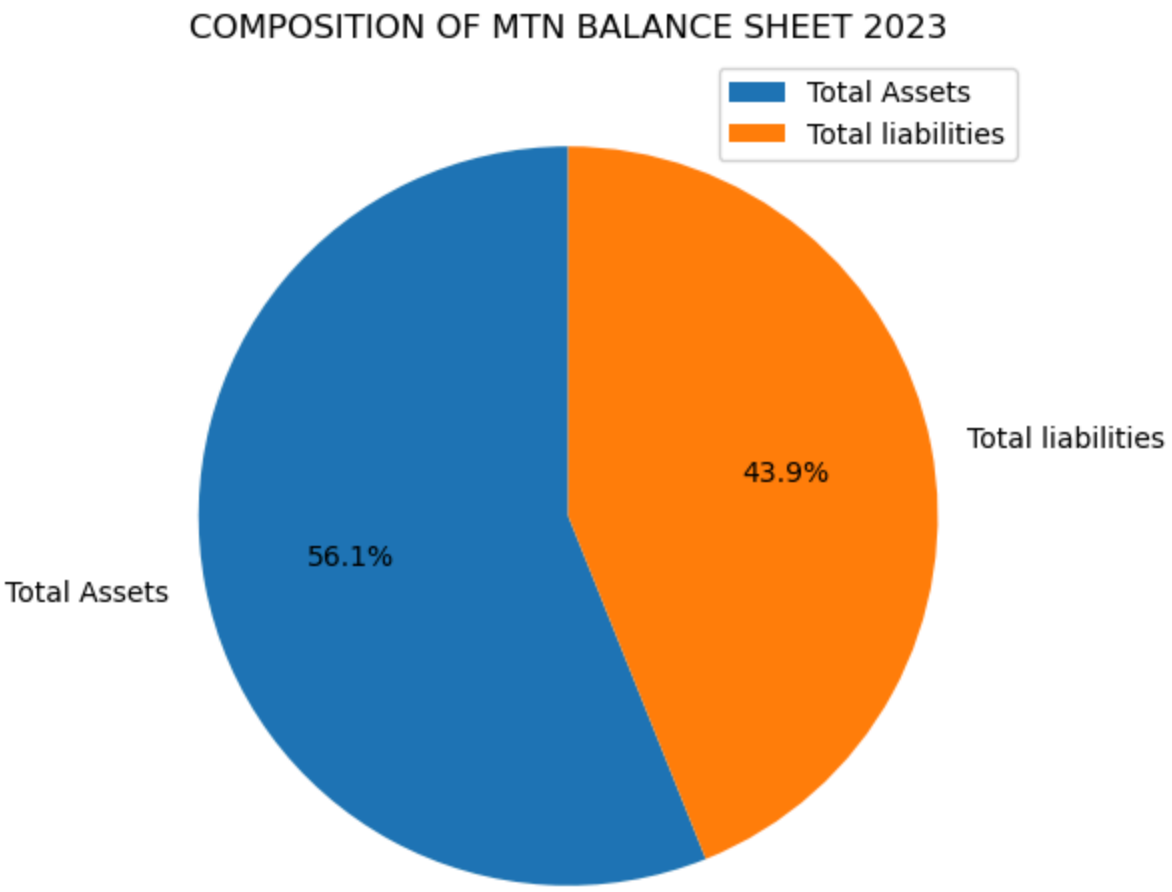
Bar graph comparing all the necessary items on the balance sheet from the years 2022 and 2023

In [237…

```
items_to_plot=['Total Assets', 'Total liabilities']
df3 = df_cleaned[df_cleaned['Items'].isin(items_to_plot)]
plt.figure(figsize=(9,6))
plt.pie(df3['2023'], labels = df3['Items'], autopct = '%1.1f%%', startangle = 90)
plt.title('COMPOSITION OF MTN BALANCE SHEET 2023')
plt.legend()
```

Out[237]:

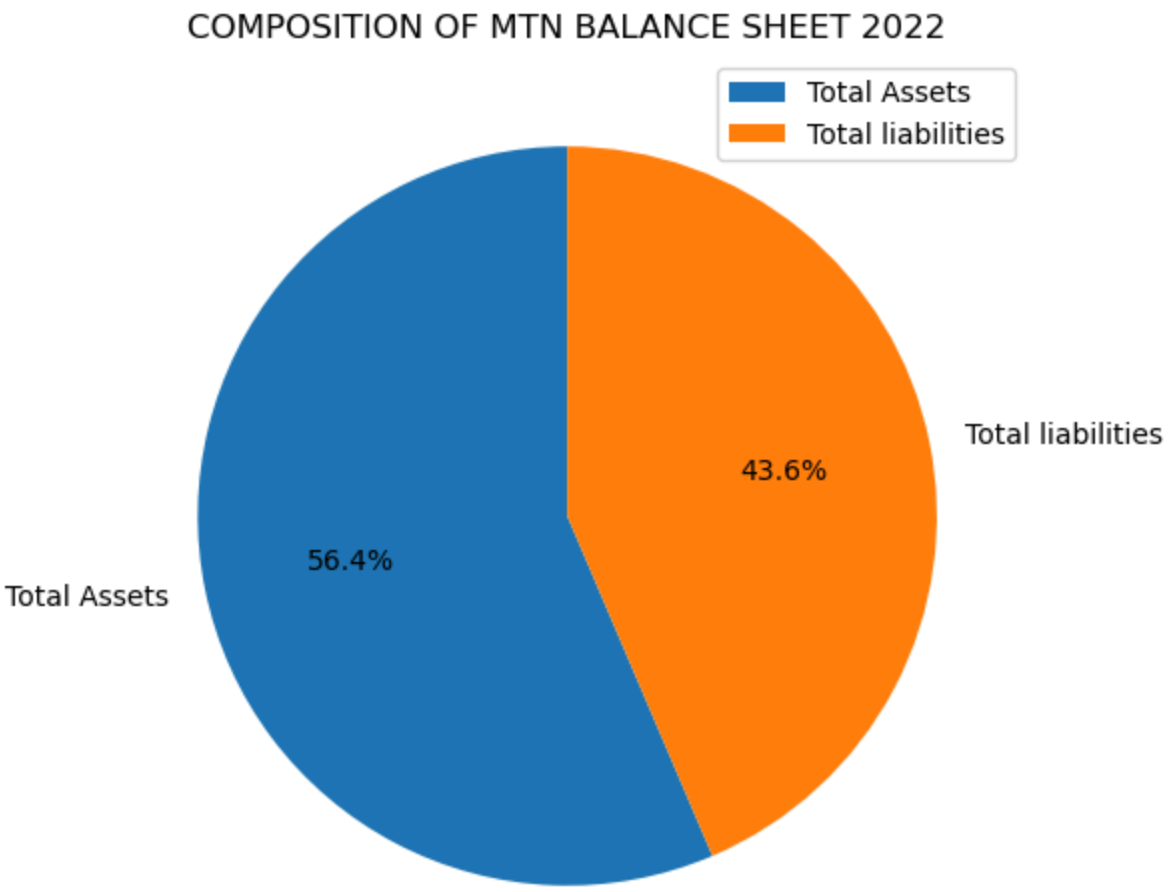
```
<matplotlib.legend.Legend at 0x23c2a4c8e50>
```



Pie chart of the Total Assets and Total Liabilities of the year 2023

In [238…
```
plt.figure(figsize=(9,6))
plt.pie(df3['2022'], labels=df3['Items'],autopct='%1.1f%%',startangle=90)
plt.title('COMPOSITION OF MTN BALANCE SHEET 2022')
plt.legend()
```

Out[238]:
```
<matplotlib.legend.Legend at 0x23c254c4d90>
```



Pie chart of Total Assets and Total Liabilities in the year 2022

Overall MTN's assets slightly dropped from 2022 to 2023 with a slight increase in ther liabilities as well owing to more telecom investments and serving more customers however the comapny is still in good financial health according to the ratios calculated

In [ ]: