

```
In [1]: import pandas as pd
import seaborn as sns
from matplotlib import pyplot as plt

df = pd.read_csv("Financial Analytics data.csv")
df['Sales Qtr - Crore'].fillna(0, inplace=True)
df[df.columns[-1]].fillna(0, inplace=True)
df=df.dropna()

df['Sales Qtr - Crore'] += df[df.columns[-1]]
df.drop(columns=[df.columns[-1]], inplace=True)
print(df)

df=df.head(10)
name=df["Name"]
marketsales=df["Mar Cap - Crore"]
fig=plt.subplots(figsize=(20,7))
plt.title("Comparing market capitalization of the top 10 companies",fontsize=20)
plt.xlabel("Name",fontsize=16)
plt.xticks(rotation=45)
plt.ylabel("Market Capitalization - Crore",fontsize=16)
plt.bar(name,marketsales,width=0.4,color=("red","blue","green","yellow"))
plt.show()
```

C:\Users\fathi\AppData\Local\Temp\ipykernel_24980\1752408924.py:6: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.

The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.

```
df['Sales Qtr - Crore'].fillna(0, inplace=True)
```

C:\Users\fathi\AppData\Local\Temp\ipykernel_24980\1752408924.py:7: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.

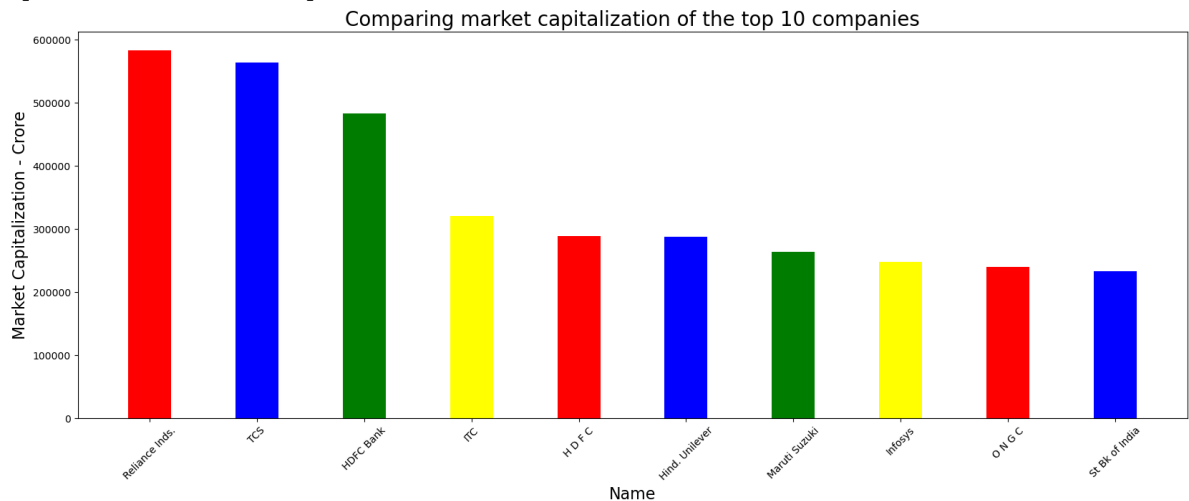
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.

```
df[df.columns[-1]].fillna(0, inplace=True)
```

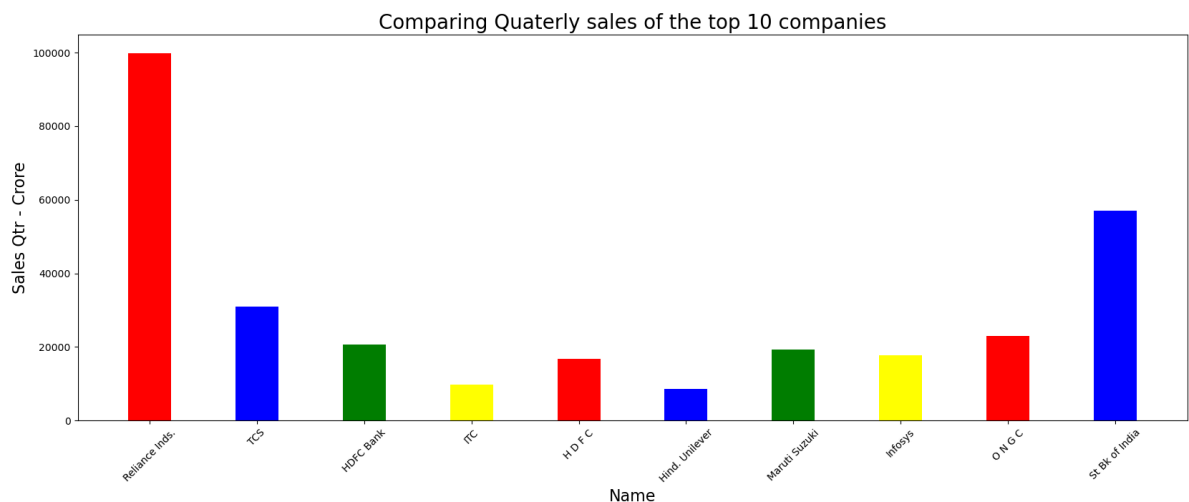
	S.No.	Name	Mar Cap - Crore	Sales Qtr - Crore
0	1	Reliance Inds.	583436.72	99810.00
1	2	TCS	563709.84	30904.00
2	3	HDFC Bank	482953.59	20581.27
3	4	ITC	320985.27	9772.02
4	5	H D F C	289497.37	16840.51
..
482	495	Prime Focus	3031.50	609.61
483	496	Lak. Vilas Bank	3029.57	790.17
484	497	NOCIL	3026.26	249.27
485	498	Orient Cement	3024.32	511.53
486	499	Natl.Fertilizer	3017.07	2840.75

[479 rows x 4 columns]



In []: Summary: Reliance Industries has the highest market capitalization, followed by TCS an

```
In [3]: df=df.head(10)
name=df["Name"]
marketsales=df["Sales Qtr - Crore"]
fig=plt.subplots(figsize=(20,7))
plt.title("Comparing Quaterly sales of the top 10 companies",fontsize=20)
plt.xlabel("Name",fontsize=16)
plt.xticks(rotation=45)
plt.ylabel("Sales Qtr - Crore",fontsize=16)
plt.bar(name,marketsales,width=0.4,color=("red","blue","green","yellow"))
plt.show()
```

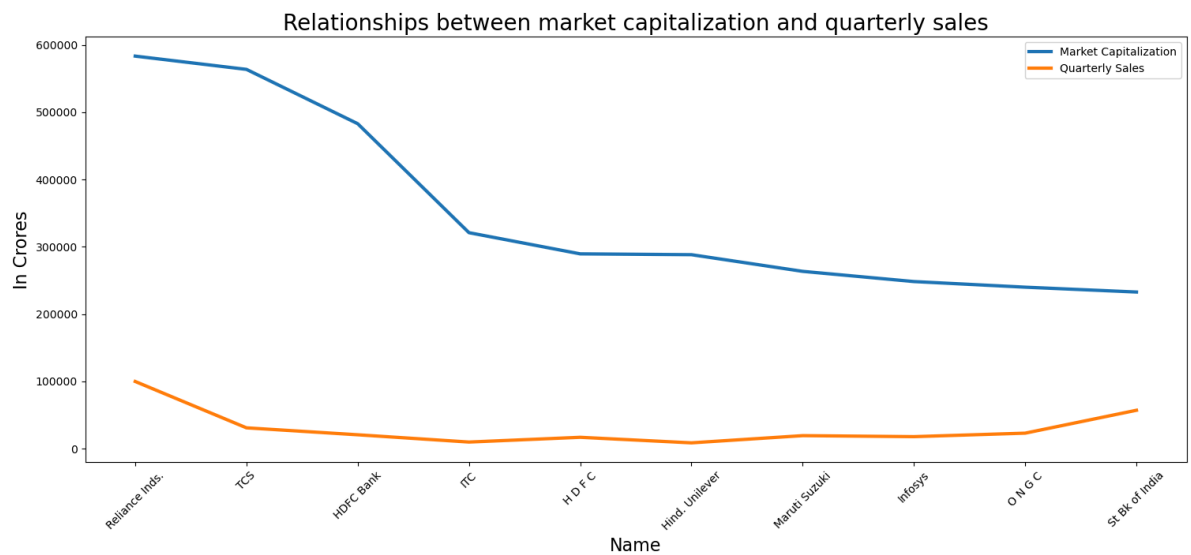


In []: Summary: Reliance Industries dominates quarterly sales, significantly outpacing other

```

In [5]: name=df["Name"]
marketcap=df["Mar Cap - Crore"]
fig=plt.subplots(figsize=(18,7))
plt.title("Relationships between market capitalization and quarterly sales",fontsize=
plt.xlabel("Name",fontsize=16)
plt.xticks(rotation=45)
plt.ylabel("In Crores",fontsize=16)
plt.plot(name,marketcap,label="Market Capitalization",linewidth=3.0)
qtrsales=df["Sales Qtr - Crore"]
plt.plot(name,qtrsales,label="Quarterly Sales",linewidth=3.0)
plt.legend()
plt.show()

```



```

In [ ]: Summary : Reliance Industries leads both market capitalization and sales, with varied

```