In [2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
icici=pd.read_csv(r"C:\Users\User\Downloads\ICICI.csv")
```

In [4]:
```
kotak=pd.read_csv(r"C:\Users\User\Downloads\Kotak Gold ETF.csv")
```

In [5]:
```
sbi=pd.read_csv(r"C:\Users\User\Downloads\SBI Gold ETF.csv")
```

In [6]:
```
hdfc=pd.read_csv(r"C:\Users\User\Downloads\HDFC Gold Exchange Traded Fund.csv")
```

In [7]:
```
nippon=pd.read_csv(r"C:\Users\User\Downloads\Nippon India ETF Gold BeES.csv")
```

In [8]:
```
gold=pd.read_excel(r"C:\Users\User\Downloads\Gold prices.xlsx")
```

Data cleaning

In [10]:
```
icici.columns
```

Out[10]:
```
Index(['Date ', 'series ', 'OPEN ', 'HIGH ', 'LOW ', 'PREV. CLOSE ', 'ltp ',
       'close ', 'vwap ', '52W H ', '52W L ', 'VOLUME ', 'VALUE ',
       'No of trades '],
      dtype='object')
```

In [11]:
```
icici.drop(columns=['series ', 'OPEN ', 'HIGH ', 'LOW ', 'ltp ', 'vwap ', '52W H ',
kotak.drop(columns=['series ', 'OPEN ', 'HIGH ', 'LOW ', 'ltp ', 'vwap ', '52W H ',
sbi.drop(columns=['series ', 'OPEN ', 'HIGH ', 'LOW ', 'ltp ', 'vwap ', '52W H ', '
nippon.drop(columns=['series ', 'OPEN ', 'HIGH ', 'LOW ', 'ltp ', 'vwap ', '52W H '
hdfc.drop(columns=['series ', 'OPEN ', 'HIGH ', 'LOW ', 'ltp ', 'vwap ', '52W H ',
```

In [12]:
```
icici.head()
```

Out[12]:

	Date	PREV. CLOSE	close	VOLUME	VALUE	No of trades
0	27-Oct-2025	103.72	103.34	61,76,486	64,13,33,104.01	33,931
1	24-Oct-2025	104.76	103.72	90,90,948	94,83,20,501.00	27,568
2	23-Oct-2025	109.17	104.76	1,54,99,131	1,60,33,59,590.59	60,806
3	21-Oct-2025	108.23	109.17	41,01,716	44,88,71,981.89	22,166
4	20-Oct-2025	111.83	108.23	1,59,40,422	1,72,86,57,896.89	56,512

In [13]:
```
icici.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 248 entries, 0 to 247
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Date           248 non-null    object 
 1   PREV. CLOSE    248 non-null    float64
 2   close          248 non-null    float64
 3   VOLUME         248 non-null    object 
 4   VALUE          248 non-null    object 
 5   No of trades   248 non-null    object 
dtypes: float64(2), object(4)
memory usage: 11.8+ KB
```

In [14]:
```
kotak.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 248 entries, 0 to 247
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Date           248 non-null    object 
 1   PREV. CLOSE    248 non-null    float64
 2   close          248 non-null    float64
 3   VOLUME         248 non-null    object 
 4   VALUE          248 non-null    object 
 5   No of trades   248 non-null    object 
dtypes: float64(2), object(4)
memory usage: 11.8+ KB
```

In [15]:
```
sbi.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 248 entries, 0 to 247
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Date           248 non-null    object 
 1   PREV. CLOSE    248 non-null    float64
 2   close          248 non-null    float64
 3   VOLUME         248 non-null    object 
 4   VALUE          248 non-null    object 
 5   No of trades   248 non-null    object 
dtypes: float64(2), object(4)
memory usage: 11.8+ KB
```

In [16]:
```
hdfc.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 248 entries, 0 to 247
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Date           248 non-null    object
 1   PREV. CLOSE    248 non-null    float64
 2   close          248 non-null    float64
 3   VOLUME         248 non-null    object
 4   VALUE          248 non-null    object
 5   No of trades   248 non-null    object
dtypes: float64(2), object(4)
memory usage: 11.8+ KB
```

In [17]:
```
nippon.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 248 entries, 0 to 247
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Date           248 non-null    object
 1   PREV. CLOSE    248 non-null    float64
 2   close          248 non-null    float64
 3   VOLUME         248 non-null    object
 4   VALUE          248 non-null    object
 5   No of trades   248 non-null    object
dtypes: float64(2), object(4)
memory usage: 11.8+ KB
```

In [18]:
```
icici.head()
```

Out[18]:

	Date	PREV. CLOSE	close	VOLUME	VALUE	No of trades
0	27-Oct-2025	103.72	103.34	61,76,486	64,13,33,104.01	33,931
1	24-Oct-2025	104.76	103.72	90,90,948	94,83,20,501.00	27,568
2	23-Oct-2025	109.17	104.76	1,54,99,131	1,60,33,59,590.59	60,806
3	21-Oct-2025	108.23	109.17	41,01,716	44,88,71,981.89	22,166
4	20-Oct-2025	111.83	108.23	1,59,40,422	1,72,86,57,896.89	56,512

In [19]:
```
icici["percent"] = icici.apply(lambda x: ((x["close "] - x["PREV. CLOSE "]) / x["PR
kotak["percent"] = kotak.apply(lambda x: ((x["close "] - x["PREV. CLOSE "]) / x["PR
sbi["percent"] = sbi.apply(lambda x: ((x["close "] - x["PREV. CLOSE "]) / x["PREV.
nippon["percent"] = nippon.apply(lambda x: ((x["close "] - x["PREV. CLOSE "]) / x["
hdfc["percent"] = hdfc.apply(lambda x: ((x["close "] - x["PREV. CLOSE "]) / x["PREV
```

In [20]:
```
icici.head()
```

Out[20]:

	Date	PREV. CLOSE	close	VOLUME	VALUE	No of trades	percent
0	27-Oct-2025	103.72	103.34	61,76,486	64,13,33,104.01	33,931	-0.366371
1	24-Oct-2025	104.76	103.72	90,90,948	94,83,20,501.00	27,568	-0.992745
2	23-Oct-2025	109.17	104.76	1,54,99,131	1,60,33,59,590.59	60,806	-4.039571
3	21-Oct-2025	108.23	109.17	41,01,716	44,88,71,981.89	22,166	0.868521
4	20-Oct-2025	111.83	108.23	1,59,40,422	1,72,86,57,896.89	56,512	-3.219172

In [21]:
```
gold.head()
```

Out[21]:

	Commodity	Unit	Location	Spot Price(Rs.)	Up/Down	Date	Time
0	GOLD	10 GRMS	AHMEDABAD	120909	-	2025-10-27	18:00:00
1	GOLD	10 GRMS	AHMEDABAD	121841	-	2025-10-27	13:30:00
2	GOLD	10 GRMS	AHMEDABAD	126854	=	2025-10-24	12:39:00
3	GOLD	10 GRMS	AHMEDABAD	126854	=	2025-10-23	15:10:00
4	GOLD	10 GRMS	AHMEDABAD	126854	=	2025-10-22	18:23:00

In [22]:
```
gold["Date"].nunique()
```

Out[22]: 275

In [23]:
```
gold["Time"].nunique()
```

Out[23]: 190

In [24]:
```
gold["Time"]=pd.to_datetime(gold["Time"],format="%H:%M:%S").dt.time
```

In [25]:
```
gold.head()
```

Out[25]:

	Commodity	Unit	Location	Spot Price(Rs.)	Up/Down	Date	Time
0	GOLD	10 GRMS	AHMEDABAD	120909	-	2025-10-27	18:00:00
1	GOLD	10 GRMS	AHMEDABAD	121841	-	2025-10-27	13:30:00
2	GOLD	10 GRMS	AHMEDABAD	126854	=	2025-10-24	12:39:00
3	GOLD	10 GRMS	AHMEDABAD	126854	=	2025-10-23	15:10:00
4	GOLD	10 GRMS	AHMEDABAD	126854	=	2025-10-22	18:23:00

In [26]:
```
gold=gold[gold["Time"]==gold.groupby("Date")["Time"].transform("max")]
```

In [27]:
```
gold.reset_index(inplace=True)
```

In [28]:
```
gold=gold.copy()
gold.drop(columns=["Unit","Location","Commodity","Time","Up/Down","index"],inplace=
```

In [29]:
```
gold.head()
```

Out[29]:

	Spot Price(Rs.)	Date
0	120909	2025-10-27
1	126854	2025-10-24
2	126854	2025-10-23
3	126854	2025-10-22
4	126854	2025-10-21

In [30]:
```
gold["PREV. CLOSE"]=gold["Spot Price(Rs.)"].shift(-1)
```

In [31]:
```
gold.head()
```

Out[31]:

	Spot Price(Rs.)	Date	PREV. CLOSE
0	120909	2025-10-27	126854.0
1	126854	2025-10-24	126854.0
2	126854	2025-10-23	126854.0
3	126854	2025-10-22	126854.0
4	126854	2025-10-21	126854.0

In [32]:
```
gold["percent"] = gold.apply(lambda x: ((x["Spot Price(Rs.)"] - x["PREV. CLOSE"]) /
```

In [33]:
```
gold.head()
```

Out[33]:

	Spot Price(Rs.)	Date	PREV. CLOSE	percent
0	120909	2025-10-27	126854.0	-4.68649
1	126854	2025-10-24	126854.0	0.00000
2	126854	2025-10-23	126854.0	0.00000
3	126854	2025-10-22	126854.0	0.00000
4	126854	2025-10-21	126854.0	0.00000

In [34]:
```
gold.drop(gold.index[-1])
```

Out[34]:

	Spot Price(Rs.)	Date	PREV. CLOSE	percent
0	120909	2025-10-27	126854.0	-4.686490
1	126854	2025-10-24	126854.0	0.000000
2	126854	2025-10-23	126854.0	0.000000
3	126854	2025-10-22	126854.0	0.000000
4	126854	2025-10-21	126854.0	0.000000
...	...	...	...	...
269	74590	2024-10-09	75279.0	-0.915262
270	75279	2024-10-08	75656.0	-0.498308
271	75656	2024-10-07	75694.0	-0.050202
272	75694	2024-10-04	75320.0	0.496548
273	75320	2024-10-03	75213.0	0.142263

274 rows × 4 columns

In [35]:
```
gold = gold[['Date','Spot Price(Rs.)','PREV. CLOSE','percent']]
```

In [36]:
```
gold.columns
```

Out[36]:
```
Index(['Date', 'Spot Price(Rs.)', 'PREV. CLOSE', 'percent'], dtype='object')
```

In [37]:
```
# gold["Date"]=pd.to_datetime(gold["Date"],format="%d-%b-%Y")
gold["Date"] = gold["Date"].dt.strftime("%d-%b-%Y")
gold.head()
```

Out[37]:

	Date	Spot Price(Rs.)	PREV. CLOSE	percent
0	27-Oct-2025	120909	126854.0	-4.68649
1	24-Oct-2025	126854	126854.0	0.00000
2	23-Oct-2025	126854	126854.0	0.00000
3	22-Oct-2025	126854	126854.0	0.00000
4	21-Oct-2025	126854	126854.0	0.00000

Does ETFs shows the same growth as prices of gold

In [39]:
```
from scipy.stats import f_oneway
```

In [40]:
```
merged=gold.merge(icici,left_on="Date",right_on="Date ",how="inner")   #merging dat
```

In [41]:
```
merged.head() #hence ETF shows the same growth therefore taking icici_etf for analy
```

Out[41]:

	Date	Spot Price(Rs.)	PREV. CLOSE	percent_x	Date	PREV. CLOSE	close	VOLUME	VALUE
0	27-Oct-2025	120909	126854.0	-4.686490	27-Oct-2025	103.72	103.34	61,76,486	64,13,33,104.01
1	24-Oct-2025	126854	126854.0	0.000000	24-Oct-2025	104.76	103.72	90,90,948	94,83,20,501.00
2	23-Oct-2025	126854	126854.0	0.000000	23-Oct-2025	109.17	104.76	1,54,99,131	1,60,33,59,590.59
3	21-Oct-2025	126854	126854.0	0.000000	21-Oct-2025	108.23	109.17	41,01,716	44,88,71,981.89
4	20-Oct-2025	126854	129828.0	-2.290723	20-Oct-2025	111.83	108.23	1,59,40,422	1,72,86,57,896.89

In [42]:
```
merged.rename(columns={"percent_x":"rate_gold"},inplace="True")
merged.rename(columns={"percent_y":"rate_icici"},inplace="True")
```

In [43]:
```
merged.head()
```

Out[43]:

	Date	Spot Price(Rs.)	PREV. CLOSE	rate_gold	Date	PREV. CLOSE	close	VOLUME	VALUE
0	27-Oct-2025	120909	126854.0	-4.686490	27-Oct-2025	103.72	103.34	61,76,486	64,13,33,104.01
1	24-Oct-2025	126854	126854.0	0.000000	24-Oct-2025	104.76	103.72	90,90,948	94,83,20,501.00
2	23-Oct-2025	126854	126854.0	0.000000	23-Oct-2025	109.17	104.76	1,54,99,131	1,60,33,59,590.59
3	21-Oct-2025	126854	126854.0	0.000000	21-Oct-2025	108.23	109.17	41,01,716	44,88,71,981.89
4	20-Oct-2025	126854	129828.0	-2.290723	20-Oct-2025	111.83	108.23	1,59,40,422	1,72,86,57,896.89

In [44]:
```
# Ho = Prices of gold are similar as prices of ETF
# Ha = There is noticable change in prices of ETF and gold
```

In [45]:
```
alpha=0.05
```

In [46]:
```
p_value = f_oneway(merged["rate_gold"],merged["rate_icici"])[1]
p_value
```

Out[46]: 0.9408902824879943

In [47]:
```
if p_value < alpha:
    print("Reject Ho")
else:
    print("Fail to Reject Ho")
```

```
Fail to Reject Ho
```

In [48]:
```
# Here p_value is greater then 0.05
# Since the p_vaue is greater then 0.05 therefore we assume there is no noticeable
```

Is any ETF is better then other

In [50]:
```
df=pd.DataFrame({
    "Date" : icici["Date "],
    "icici" : icici["percent"],
    "sbi" : sbi["percent"],
    "hdfc" : hdfc["percent"],
    "kotak" : kotak["percent"],
```

```
    "nippon" : nippon["percent"]
})
```

In [51]:
```
# Ho = All ETF are similar
# Ha = At least one ETF mean is different
```

In [52]:
```
df
```

Out[52]:

	Date	icici	sbi	hdfc	kotak	nippon
0	27-Oct-2025	-0.366371	-0.425080	-0.087125	-0.276954	-0.278829
1	24-Oct-2025	-0.992745	-1.278016	-1.852732	-1.490792	-1.258604
2	23-Oct-2025	-4.039571	-3.710166	-3.564229	-3.796400	-3.775192
3	21-Oct-2025	0.868521	0.165578	0.192784	0.489827	0.523112
4	20-Oct-2025	-3.219172	-2.510986	-2.357476	-2.435438	-2.747202
...	...	...	...	...	...	...
243	01-Nov-2024	-0.536621	-0.566284	-0.493469	-0.401487	-0.448296
244	31-Oct-2024	-0.144823	-0.144991	-0.101493	-0.118818	-0.149209
245	30-Oct-2024	1.053710	1.025341	1.054945	0.944528	1.085973
246	29-Oct-2024	0.826324	0.767528	0.842199	0.968816	0.790514
247	28-Oct-2024	0.429757	0.444774	0.326119	0.379881	0.504202

248 rows × 6 columns

In [53]:
```
p_value_2 = f_oneway(df["icici"],df["sbi"],df["hdfc"],df["kotak"],df["nippon"])[1]
p_value_2
```

Out[53]:
```
0.9999999875751644
```

In [54]:
```
if p_value_2 < alpha:
    print("Reject Ho")
else:
    print("Fail to Reject Ho")
```

```
Fail to Reject Ho
```

In [55]:
```
# Here p_value is less then 0.05
# Since the p_value is greater then 0.05 therefore we assume there is no noticeable
```

Flutuation rate of each ETF

In [57]:
```
print("icici standard deviation :- ",round(df["icici"].std(),ndigits=2))
print("sbi standard deviation :- ",round(df["sbi"].std(),ndigits=2))
print("hdfc standard deviation :- ",round(df["hdfc"].std(),ndigits=2))
```

```
print("kotak standard deviation :- ",round(df["kotak"].std(),ndigits=2))
print("nippon standard deviation :- ",round(df["nippon"].std(),ndigits=2))
```

```
icici standard deviation :-  1.09
sbi standard deviation :-  1.03
hdfc standard deviation :-  1.02
kotak standard deviation :-  1.05
nippon standard deviation :-  1.07
```

In [58]:
```
# Here hdfc shows the least standard deviation means it is the more stable ETF amou
# and icici shows the maximum standard deviatiom
```

In [59]:
```
merged_hdfc=gold.merge(hdfc[["Date ","percent",]],left_on="Date",right_on="Date ",h
merged_hdfc
```

Out[59]:

	Date	Spot Price(Rs.)	PREV. CLOSE	percent_gold	Date	percent_hdfc
0	27-Oct-2025	120909	126854.0	-4.686490	27-Oct-2025	-0.087125
1	24-Oct-2025	126854	126854.0	0.000000	24-Oct-2025	-1.852732
2	23-Oct-2025	126854	126854.0	0.000000	23-Oct-2025	-3.564229
3	21-Oct-2025	126854	126854.0	0.000000	21-Oct-2025	0.192784
4	20-Oct-2025	126854	129828.0	-2.290723	20-Oct-2025	-2.357476
...	...	...	...	...	...	...
242	01-Nov-2024	79181	79181.0	0.000000	01-Nov-2024	-0.493469
243	31-Oct-2024	79181	79362.0	-0.228069	31-Oct-2024	-0.101493
244	30-Oct-2024	79362	78493.0	1.107105	30-Oct-2024	1.054945
245	29-Oct-2024	78493	77917.0	0.739248	29-Oct-2024	0.842199
246	28-Oct-2024	77917	77622.0	0.380047	28-Oct-2024	0.326119

247 rows × 6 columns

In [60]:
```
df.head()
```

Out[60]:

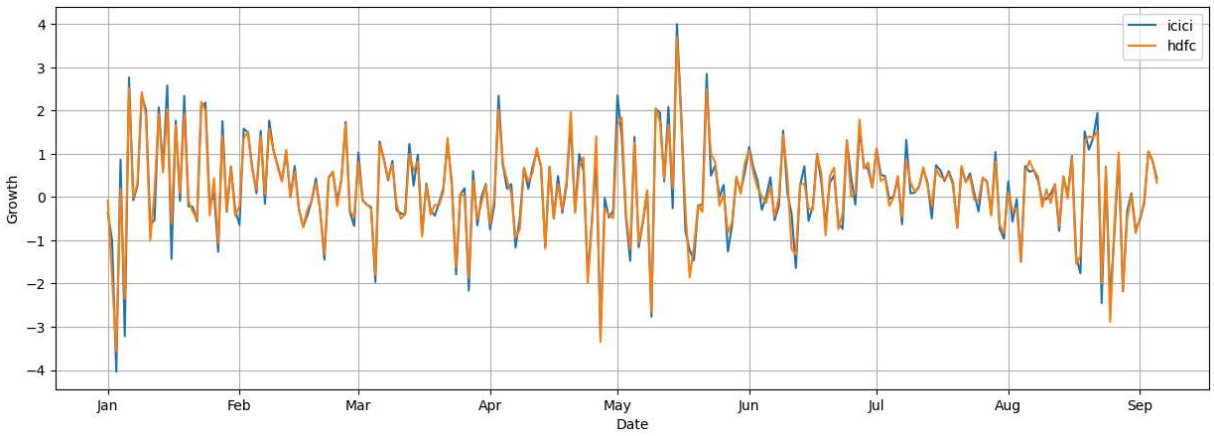
	Date	icici	sbi	hdfc	kotak	nippon
0	27-Oct-2025	-0.366371	-0.425080	-0.087125	-0.276954	-0.278829
1	24-Oct-2025	-0.992745	-1.278016	-1.852732	-1.490792	-1.258604
2	23-Oct-2025	-4.039571	-3.710166	-3.564229	-3.796400	-3.775192
3	21-Oct-2025	0.868521	0.165578	0.192784	0.489827	0.523112
4	20-Oct-2025	-3.219172	-2.510986	-2.357476	-2.435438	-2.747202

In [61]:
```
df["Date"]=icici["Date "]
```

In [62]:
```
import matplotlib.dates as mdates
```

In [63]:
```
plt.figure(figsize=(15,5))

sns.lineplot(data=df,x="Date",y="icici", label="icici")
sns.lineplot(data=df,x="Date",y="hdfc", label="hdfc")
plt.gca().xaxis.set_major_locator(mdates.MonthLocator())  # Show 1 tick per month
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter("%b"))  # Format as Jan, F
plt.xlabel("Date")
plt.ylabel("Growth")
plt.legend()
plt.grid(True)
plt.show()
```



In [64]:
```
# Here hdfc shows less reponse to the changes in gold rates where as icici ETF show
```

Visualization

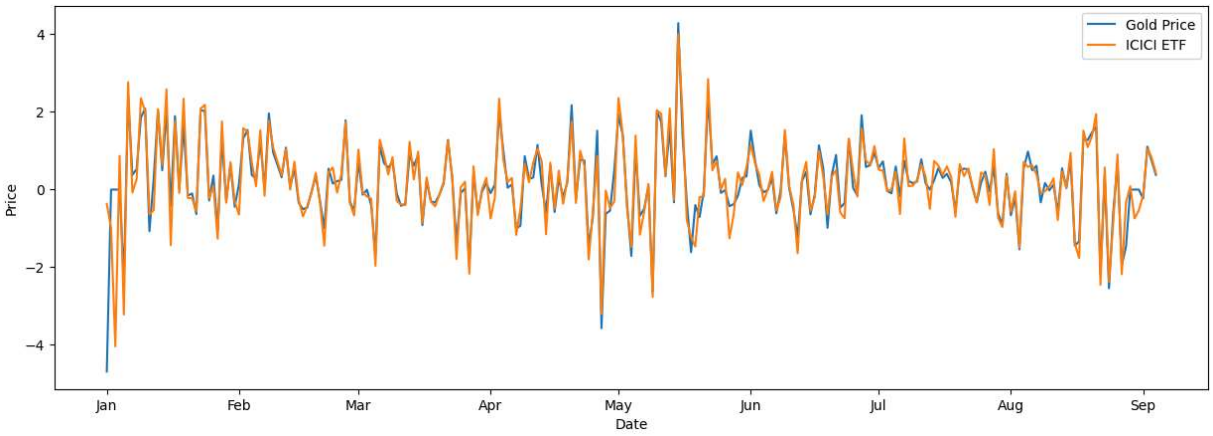
Gold vs ETF

In [67]:
```
# Comparing gold with icici etf as all ETFs are similar
```

In [68]:
```
plt.figure(figsize=(15,5))
```

```
sns.lineplot(data=merged, x="Date", y="rate_gold", label="Gold Price")
sns.lineplot(data=merged, x="Date", y="rate_icici", label="ICICI ETF")

plt.gca().xaxis.set_major_locator(mdates.MonthLocator())  # Show 1 tick per month
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter("%b"))  # Format as Jan, F
plt.xlabel("Date")
plt.ylabel("Price")
plt.legend()
plt.show()
```



ETF vs ETF

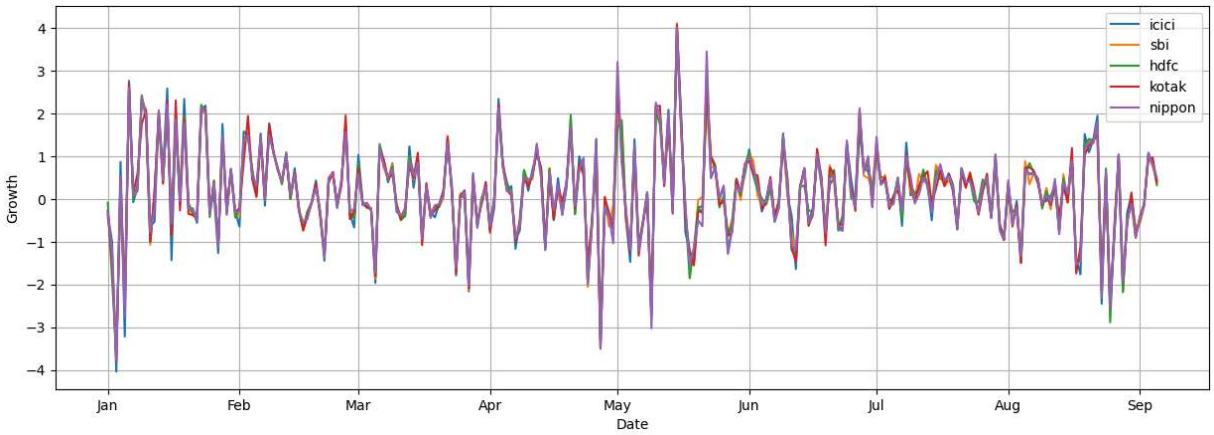
In [70]:
```
df.columns
```

Out[70]:
```
Index(['Date', 'icici', 'sbi', 'hdfc', 'kotak', 'nippon'], dtype='object')
```

In [71]:
```
plt.figure(figsize=(15,5))

sns.lineplot(data=df,x="Date",y="icici", label="icici")
sns.lineplot(data=df,x="Date",y="sbi", label="sbi")
sns.lineplot(data=df,x="Date",y="hdfc", label="hdfc")
sns.lineplot(data=df,x="Date",y="kotak", label="kotak")
sns.lineplot(data=df,x="Date",y="nippon", label="nippon")

plt.gca().xaxis.set_major_locator(mdates.MonthLocator())  # Show 1 tick per month
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter("%b"))  # Format as Jan, F
plt.xlabel("Date")
plt.ylabel("Growth")
plt.legend()
plt.grid(True)
plt.show()
```

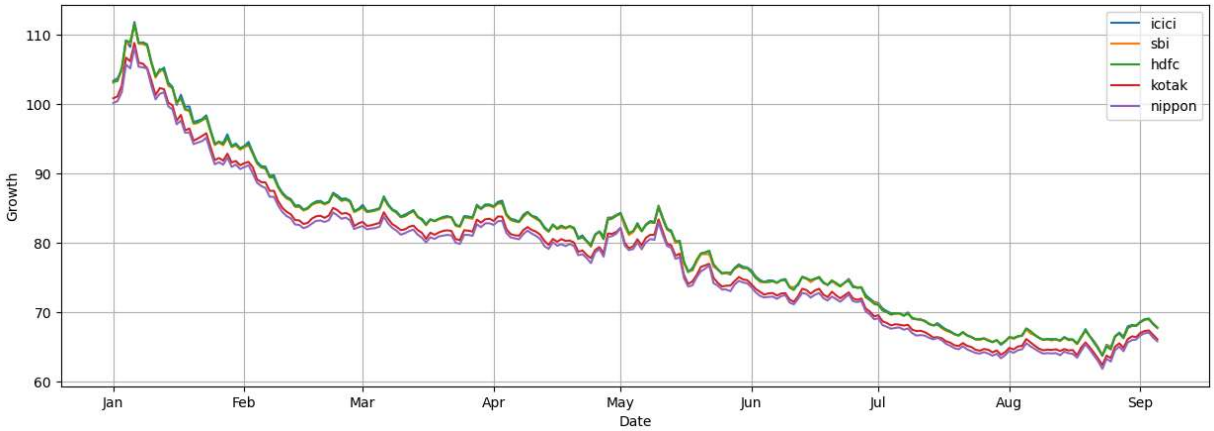


In [72]:
```
prices=pd.DataFrame({
    "Date":icici["Date "],
    "icici" :icici["close "],
    "nippon":nippon["close "],
    "hdfc":hdfc["close "],
    "sbi":sbi["close "],
    "kotak":kotak["close "]
})
```

In [73]:
```
plt.figure(figsize=(15,5))

sns.lineplot(data=prices,x="Date",y="icici", label="icici")
sns.lineplot(data=prices,x="Date",y="sbi", label="sbi")
sns.lineplot(data=prices,x="Date",y="hdfc", label="hdfc")
sns.lineplot(data=prices,x="Date",y="kotak", label="kotak")
sns.lineplot(data=prices,x="Date",y="nippon", label="nippon")

plt.gca().xaxis.set_major_locator(mdates.MonthLocator())  # Show 1 tick per month
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter("%b"))  # Format as Jan, F
plt.xlabel("Date")
plt.ylabel("Growth")
plt.legend()
plt.grid(True)
plt.show()
```



Gold VS SENSEX

In [75]:
```
sensex=pd.read_csv(r"C:\Users\User\Downloads\SENSEX_01102024_08112025.csv")
```

In [76]:
```
sensex.head()
```

Out[76]:

	Date	Open	High	Low	Close
0	1-October-2024	84257.17	84648.40	84098.94	84266.29
1	3-October-2024	83002.09	83752.81	82434.02	82497.10
2	4-October-2024	82244.25	83368.32	81532.68	81688.45
3	7-October-2024	81926.99	82137.77	80726.06	81050.00
4	8-October-2024	80826.56	81763.28	80813.07	81634.81

In [77]:
```
type(gold["Date"][0])
```

Out[77]:
```
str
```

In [78]:
```
sensex["Date"]=pd.to_datetime(sensex["Date"])
```

In [79]:
```
sensex.head()
```

Out[79]:

	Date	Open	High	Low	Close
0	2024-10-01	84257.17	84648.40	84098.94	84266.29
1	2024-10-03	83002.09	83752.81	82434.02	82497.10
2	2024-10-04	82244.25	83368.32	81532.68	81688.45
3	2024-10-07	81926.99	82137.77	80726.06	81050.00
4	2024-10-08	80826.56	81763.28	80813.07	81634.81

In [80]:
```
sensex["Date"]=sensex["Date"].dt.strftime("%d-%b-%Y")
```

In [81]:
```
sensex.info()   #To check if null value is present or not
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 274 entries, 0 to 273
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Date    274 non-null    object 
 1   Open    274 non-null    float64
 2   High    274 non-null    float64
 3   Low     274 non-null    float64
 4   Close   274 non-null    float64
dtypes: float64(4), object(1)
memory usage: 10.8+ KB
```

In [82]:
```
sensex["percent"]=sensex.apply(lambda x: ((x["Close"]-x["Open"])/x["Open"])*100,axi
```

In [83]:
```
sensex.head()
```

Out[83]:

	Date	Open	High	Low	Close	percent
0	01-Oct-2024	84257.17	84648.40	84098.94	84266.29	0.010824
1	03-Oct-2024	83002.09	83752.81	82434.02	82497.10	-0.608406
2	04-Oct-2024	82244.25	83368.32	81532.68	81688.45	-0.675792
3	07-Oct-2024	81926.99	82137.77	80726.06	81050.00	-1.070453
4	08-Oct-2024	80826.56	81763.28	80813.07	81634.81	0.999981

In [84]:
```
gold.columns
```

Out[84]:
```
Index(['Date', 'Spot Price(Rs.)', 'PREV. CLOSE', 'percent'], dtype='object')
```

In [85]:
```
merge_gold=sensex.merge(gold,on="Date",how="inner",suffixes=("_sensex","_gold"))
```

In [86]:
```
merge_gold.head()
```

Out[86]:

	Date	Open	High	Low	Close	percent_sensex	Spot Price(Rs.)	PREV. CLOSE	percer
0	01-Oct-2024	84257.17	84648.40	84098.94	84266.29	0.010824	75213	NaN	
1	03-Oct-2024	83002.09	83752.81	82434.02	82497.10	-0.608406	75320	75213.0	0.
2	04-Oct-2024	82244.25	83368.32	81532.68	81688.45	-0.675792	75694	75320.0	0.
3	07-Oct-2024	81926.99	82137.77	80726.06	81050.00	-1.070453	75656	75694.0	-0.
4	08-Oct-2024	80826.56	81763.28	80813.07	81634.81	0.999981	75279	75656.0	-0.

In [87]:
```
merge_gold =merge_gold[1:]
```

In [88]:
```
corr_=merge_gold[["percent_sensex","percent_gold"]]
```

In [89]:
```
corr_.corr()
```

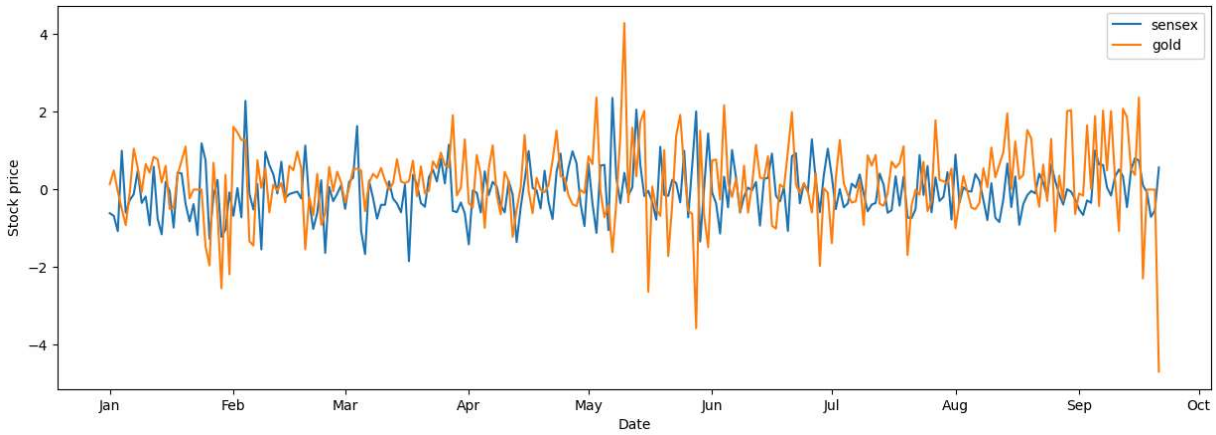
Out[89]:

	percent_sensex	percent_gold
percent_sensex	1.000000	-0.025542
percent_gold	-0.025542	1.000000

In [90]:
```
plt.figure(figsize=(15,5))

sns.lineplot(data=merge_gold, x="Date", y="percent_sensex", label="sensex")
sns.lineplot(data=merge_gold, x="Date", y="percent_gold", label="gold")

plt.gca().xaxis.set_major_locator(mdates.MonthLocator())  # Show 1 tick per month
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter("%b"))  # Format as Jan, F
plt.xlabel("Date")
plt.ylabel("Stock price")
plt.legend()
plt.show()
```



In [91]:
```
df_2=df.copy()
```

In [92]:
```
df_2.drop(columns=["Date"],inplace=True)
```

In [93]:
```
df_2.corr()
```

Out[93]:

	icici	sbi	hdfc	kotak	nippon
icici	1.000000	0.981932	0.976704	0.977669	0.980554
sbi	0.981932	1.000000	0.989353	0.983342	0.982434
hdfc	0.976704	0.989353	1.000000	0.980054	0.980754
kotak	0.977669	0.983342	0.980054	1.000000	0.985845
nippon	0.980554	0.982434	0.980754	0.985845	1.000000

In [94]:
```
from scipy.stats import pearsonr

corr, p_value = pearsonr(merge_gold["percent_sensex"], merge_gold['percent_gold'])
print(corr, p_value)
```

```
-0.02554151954037974 0.6795342890554503
```

Visualization of gold vs sensex

In [96]:
```
merge_gold.head()
```

Out[96]:

	Date	Open	High	Low	Close	percent_sensex	Spot Price(Rs.)	PREV. CLOSE	percer
1	03-Oct-2024	83002.09	83752.81	82434.02	82497.10	-0.608406	75320	75213.0	0.
2	04-Oct-2024	82244.25	83368.32	81532.68	81688.45	-0.675792	75694	75320.0	0.
3	07-Oct-2024	81926.99	82137.77	80726.06	81050.00	-1.070453	75656	75694.0	-0.
4	08-Oct-2024	80826.56	81763.28	80813.07	81634.81	0.999981	75279	75656.0	-0.
5	09-Oct-2024	81954.58	82319.21	81342.89	81467.10	-0.594817	74590	75279.0	-0.

In [97]:
```
merge_gold=merge_gold.copy()
merge_gold["sensex_close_100"]=(merge_gold["Close"]/merge_gold["Close"][1])*100
```

In [98]:
```
merge_gold["gold_close_100"]=(merge_gold["Spot Price(Rs.)"]/merge_gold["Spot Price(
```

In [99]:
```
plt.figure(figsize=(13,5))
sns.lineplot(data=merge_gold, x="Date", y="sensex_close_100", label="sensex")
sns.lineplot(data=merge_gold, x="Date", y="gold_close_100", label="gold")

plt.gca().xaxis.set_major_locator(mdates.MonthLocator())  # Show 1 tick per month
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter("%b"))  # Format as Jan, F
plt.grid()
```

In [ ]: