SmartHire Assignment

Dataset Given : NSE Stock Historical price data

By : YASH SEJWAL | 8700352079

Assignment includes :

01. Data Preprocessing and Exploratory Data Analysis, clean the dataset, handling any missing values or outliers, perform basic statistical analysis on the stock prices and volumes and Visualize the price trends of the first 10 stocks.

02. Correlation Analysis : Create a correlation matrix of stock return,Identify the most correlated pairs of stocks, visualize the correlations using a heatmap.

03. Time Series Decomposition Choose one stock and perform time series decomposition Identify trend, seasonality, and residual components Interpret the results and their implications for trading.

04. Anomaly Detection Develop a method to detect anomalous price movements Identify and list the top 5 most significant anomalies in the dataset Investigate and explain possible reasons for these anomalies.

```
Start coding or generate with AI.
```

1. Data Preprocessing and Exploratory Data Analysis

1.1 Installing the required libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from statsmodels.tsa.seasonal import seasonal_decompose
from sklearn.ensemble import IsolationForest
```

1.2 Loading the Dataset and Downloading the dataset from Kaggle and loading it into a Pandas DataFrame.

```
df = pd.read_csv('nse_all_stock_data (1).csv')
```

1.3 Checking for Missing Values (if there any exists)

```
print(df.isnull().sum())
```

```
Date            0
RELIANCE      1310
TCS           3034
HDFCBANK      1307
ICICIBANK     3005
              ...
DONEAR        4147
CAREERP       5081
INTLCONV      7778
SHREEPUSHK    6297
PRITIKAUTO    7753
Length: 1418, dtype: int64
```

1.4 Fill or drop missing values

```
df.ffill(inplace=True)
```

1.5 Checking the dataset if it has been correctly loaded

```
print(df.head())
print(df.columns)
```

```
         Date  RELIANCE  TCS  HDFCBANK  ICICIBANK  BHARTIARTL  SBIN  INFY  \
0  1991-01-02       NaN  NaN       NaN        NaN         NaN   NaN   NaN
1  1991-01-03       NaN  NaN       NaN        NaN         NaN   NaN   NaN
2  1991-01-04       NaN  NaN       NaN        NaN         NaN   NaN   NaN
3  1991-01-07       NaN  NaN       NaN        NaN         NaN   NaN   NaN
4  1991-01-08       NaN  NaN       NaN        NaN         NaN   NaN   NaN

   LICI  ITC  ...  COOLCAPS  ALLETEC  20MICRONS  VIKASECO  ORIENTBELL  DONEAR  \
```

```
0   NaN  NaN  ...       NaN      NaN        NaN       NaN         NaN     NaN
1   NaN  NaN  ...       NaN      NaN        NaN       NaN         NaN     NaN
2   NaN  NaN  ...       NaN      NaN        NaN       NaN         NaN     NaN
3   NaN  NaN  ...       NaN      NaN        NaN       NaN         NaN     NaN
4   NaN  NaN  ...       NaN      NaN        NaN       NaN         NaN     NaN

   CAREERP  INTLCONV  SHREEPUSHK  PRITIKAUTO
0      NaN       NaN         NaN         NaN
1      NaN       NaN         NaN         NaN
2      NaN       NaN         NaN         NaN
3      NaN       NaN         NaN         NaN
4      NaN       NaN         NaN         NaN

[5 rows x 1418 columns]
Index(['Date', 'RELIANCE', 'TCS', 'HDFCBANK', 'ICICIBANK', 'BHARTIARTL',
       'SBIN', 'INFY', 'LICI', 'ITC',
       ...
       'COOLCAPS', 'ALLETEC', '20MICRONS', 'VIKASECO', 'ORIENTBELL', 'DONEAR',
       'CAREERP', 'INTLCONV', 'SHREEPUSHK', 'PRITIKAUTO'],
      dtype='object', length=1418)
```

1.6 Detecting and handling outliers (if there any exists)

```
df['RELIANCE_z_score'] = np.abs(stats.zscore(df['RELIANCE'].dropna()))
df = df[df['RELIANCE_z_score'] < 3]
df.drop(columns=['RELIANCE_z_score'], inplace=True)
```

1.7 Basic Statistical Analysis

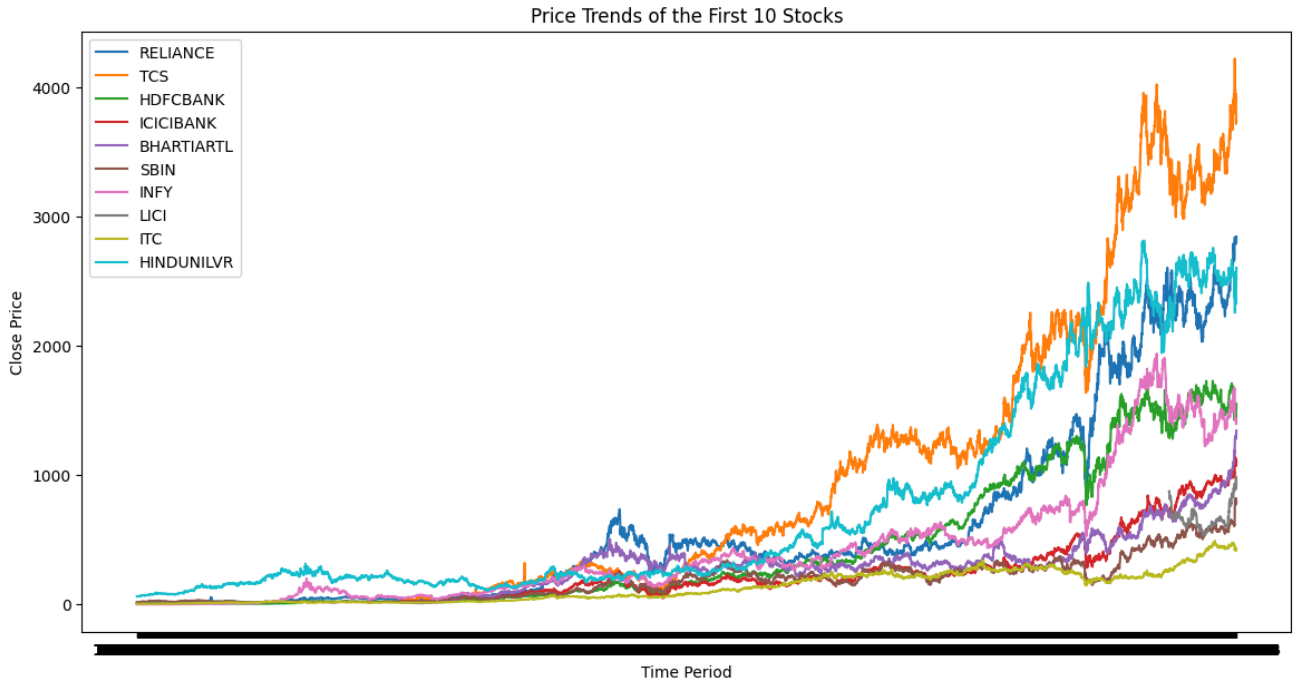
```
print(df.describe())
```

```
          RELIANCE          TCS     HDFCBANK    ICICIBANK   BHARTIARTL  \
count  7072.000000  5347.000000  7072.000000  5377.000000  5377.000000
mean    573.945305  1172.260276   423.119238   282.384655   343.680872
std     701.453921  1123.209480   516.952832   245.159280   210.296078
min      10.975116    35.474998     2.435000    20.100000     9.350544
25%      48.620407   237.203751    24.508750   130.300003   265.645569
50%     386.088425   735.775024   169.070000   200.727264   316.882050
75%     600.385162  1879.275024   638.112518   326.750000   383.822937
max    2841.500000  4219.250000  1728.199951  1146.300049  1344.349976

              SBIN         INFY        LICI          ITC   HINDUNILVR  ...  \
count  7072.000000  7072.000000  433.000000  7072.000000  7072.000000  ...
mean    180.708184   414.593949  666.470901   129.954263   734.064451  ...
std     155.455922   452.212740   87.365171   117.296752   803.246160  ...
min      13.346102     0.763183  531.849976     4.182222    61.805000  ...
25%      28.486073    78.651466  609.299988    18.548611   192.050003  ...
50%     172.937500   281.315613  649.750000    82.125000   264.799988  ...
75%     265.107491   539.518753  689.400024   225.474995   914.462494  ...
max     820.299988  1939.500000  990.000000   492.149994  2812.449951  ...

       COOLCAPS  ALLETEC    20MICRONS     VIKASECO   ORIENTBELL       DONEAR  \
count       0.0      0.0  3788.000000  3006.000000  4196.000000  4232.000000
mean        NaN      NaN    43.786602     4.384497   168.872867    48.508814
```

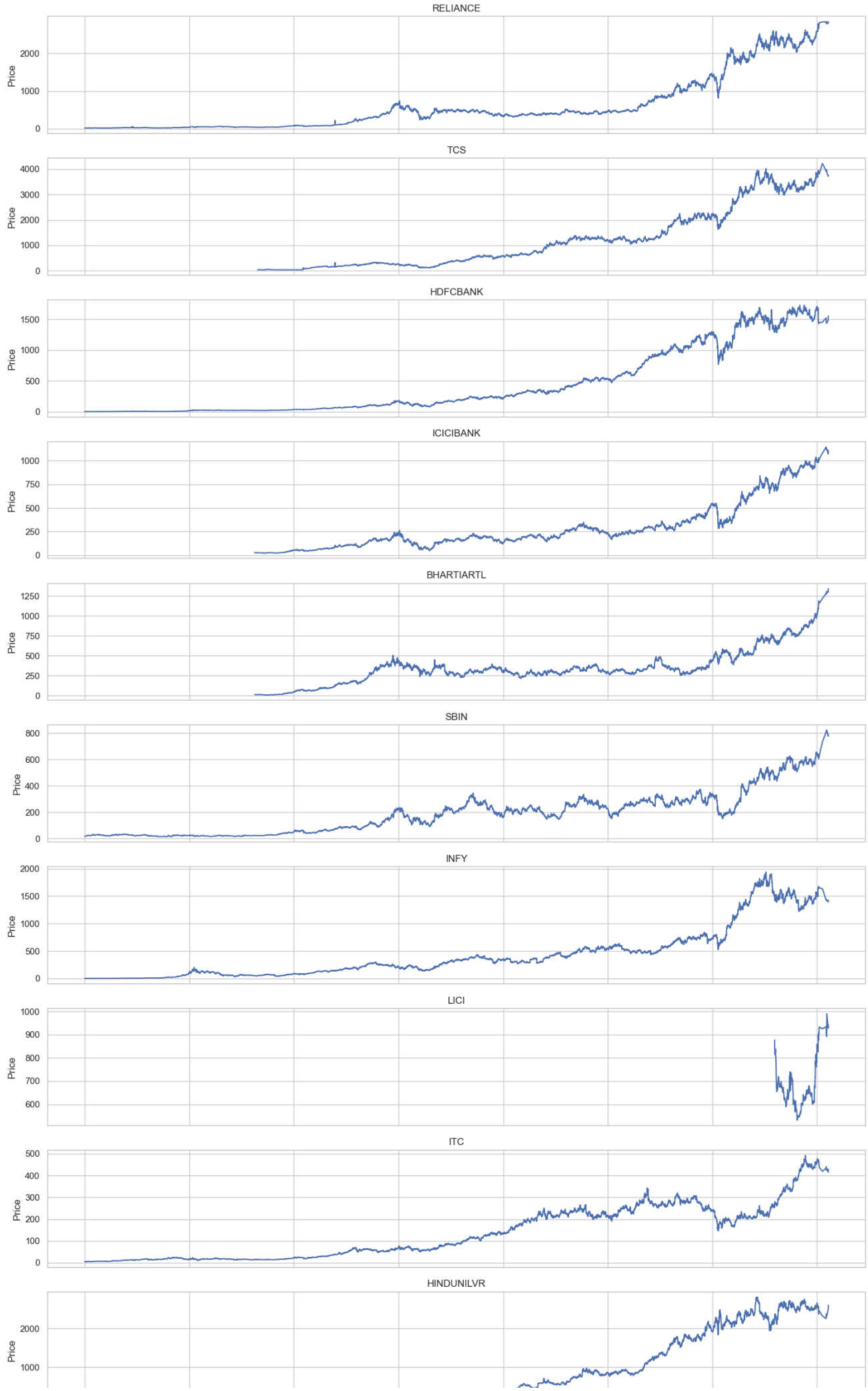
```
std          NaN      NaN    29.149753     2.935759   157.839159    45.153061
min          NaN      NaN     7.100000     0.624703    14.000000     8.900000
25%          NaN      NaN    28.600000     2.350015    54.737500    20.650000
50%          NaN      NaN    34.700001     3.450000   104.349998    31.350000
75%          NaN      NaN    47.812500     6.139939   223.012501    59.762500
max          NaN      NaN   189.500000    17.518017   806.200012   236.149994

            CAREERP    INTLCONV   SHREEPUSHK  PRITIKAUTO
count  3297.000000  597.000000  2078.000000  622.000000
mean    150.512466   68.865327   170.754235   18.633762
std      75.141700   12.338715    55.530474    5.053450
min      51.150002   47.750000    59.349998   12.700000
25%     108.349998   58.049999   121.099998   16.000000
50%     131.550003   66.949997   176.424995   17.150000
75%     164.000000   77.699997   212.387497   18.937500
max     660.000000   99.500000   332.000000   44.400002

[8 rows x 1417 columns]
```

1.8 Visualizing Price Trends of the First 10 Stocks

```
stocks = df.columns[1:11]

plt.figure(figsize=(14, 7))
for stock in stocks:
    plt.plot(df['Date'], df[stock], label=stock)

plt.xlabel('Time Period')
plt.ylabel('Close Price')
plt.title('Price Trends of the First 10 Stocks')
plt.legend()
plt.show()
```


Price Trends of the First 10 Stocks
RELIANCE
TCS
HDFCBANK
ICICIBANK
BHARTIARTL
SBIN
INFY
LICI
ITC
HINDUNILVR
Close Price
0
1000
2000
3000
4000
Time Period

```
df['Date'] = pd.to_datetime(df['Date'])

stocks = df.columns[1:11]

sns.set(style="whitegrid")

fig, axes = plt.subplots(nrows=len(stocks), ncols=1, figsize=(15, 25), sharex=True)

for i, stock in enumerate(stocks):
    sns.lineplot(ax=axes[i], x='Date', y=stock, data=df)
    axes[i].set_title(stock)
    axes[i].set_ylabel('Price')

axes[-1].set_xlabel('Date')

for ax in axes:
    for label in ax.get_xticklabels():
        label.set_rotation(45)

plt.tight_layout()

plt.show()
```


RELIANCE
Price
2000
1000
0
TCS
Price
4000
3000
2000
1000
0
HDFCBANK
Price
1500
1000
500
0
ICICIBANK
Price
1000
750
500
250
0
BHARTIARTL
Price
1250
1000
750
500
250
0
SBIN
Price
800
600
400
200
0
INFY
Price
2000
1500
1000
500
0
LICI
Price
1000
900
800
700
600
ITC
Price
500
400
300
200
100
0
HINDUNILVR
Price
2000
1000


0
1996
2000
2004
2008
2012
2016
2020
2024
Date

2. Correlation Analysis

2.1 Pivot Data and Calculate Returns

```
# Step 1: Inspect the dataset
print(df.info())
print(df.head())

# Step 2: Select only numeric columns
numeric_data = df.select_dtypes(include=[np.number])

# Check the percentage of missing values in each column
missing_percentage = numeric_data.isna().mean() * 100
print("Missing values percentage in each column:\n", missing_percentage)

# Drop columns with more than a certain threshold of missing values
threshold = 50  # For example, 50% missing values
numeric_data = numeric_data.loc[:, missing_percentage < threshold]

# Forward fill remaining missing values
numeric_data = numeric_data.ffill()

# Check if numeric_data is empty after dropping columns
if numeric_data.empty:
    raise ValueError("No numeric columns found in the dataset after dropping columns with

# Diagnostic: Print the first few rows of numeric_data
print("Numeric data after forward fill and dropping columns with too many missing values:
print(numeric_data.head())

# Step 3: Calculate daily returns
returns = numeric_data.pct_change().dropna()

# Diagnostic: Print the first few rows of returns
print("Returns after pct_change:")
print(returns.head())

# Check if returns is empty after calculating pct_change
if returns.empty:
    raise ValueError("No valid returns calculated. Check the data.")

# Step 4: Drop columns with all NaN values after pct_change
returns = returns.dropna(axis=1, how='all')

# Check if returns is empty after dropping columns with all NaN values
if returns.empty:
    raise ValueError("All columns contain NaN values after calculating returns.")

# Step 5: Compute the correlation matrix
correlation_matrix = returns.corr()

# Check if correlation_matrix is empty
if correlation_matrix.empty:
    raise ValueError("Correlation matrix is empty. Check the returns data.")

# Print the correlation matrix
print("Correlation matrix:\n", correlation_matrix)

# Step 6: Find the most correlated pairs of stocks
```

```
corr_pairs = correlation_matrix.unstack().sort_values(kind="quicksort")
most_correlated = corr_pairs[(corr_pairs < 1) & (corr_pairs == corr_pairs.max())]
most_correlated_pairs = most_correlated.index.tolist()

# Step 7: Visualize the correlation matrix using a heatmap
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Stock Return Correlation Matrix')
plt.show()

# Output the most correlated pairs
print("Most Correlated Pairs:\n", most_correlated)
```

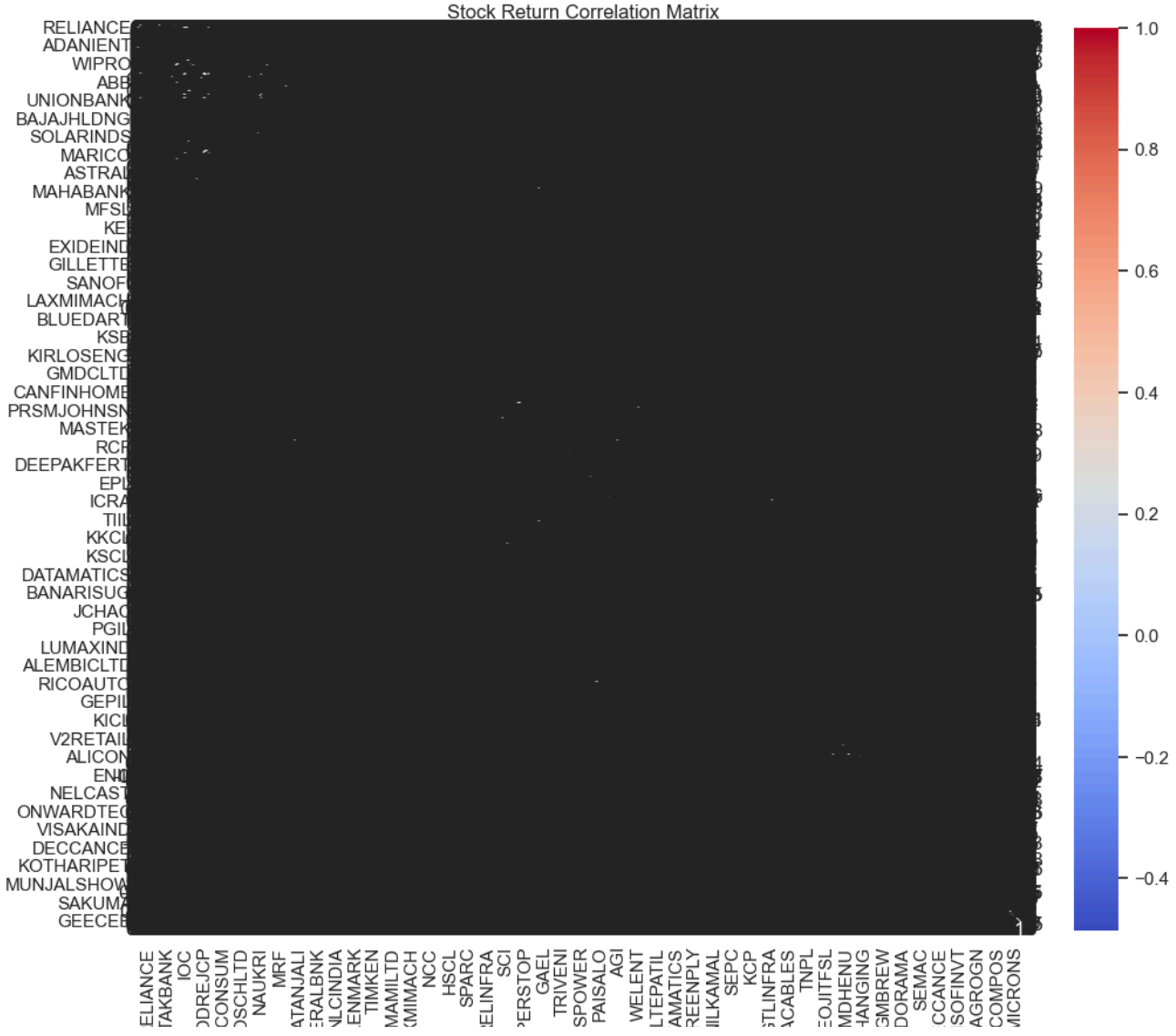
```
<class 'pandas.core.frame.DataFrame'>
Index: 7072 entries, 1303 to 8448
Columns: 1418 entries, Date to PRITIKAUTO
dtypes: datetime64[ns](1), float64(1417)
memory usage: 76.6 MB
None
           Date   RELIANCE  TCS  HDFCBANK  ICICIBANK  BHARTIARTL       SBIN  \
1303 1996-01-01  14.691803  NaN     2.980        NaN         NaN  18.823240
1304 1996-01-02  14.577553  NaN     2.975        NaN         NaN  18.224106
1305 1996-01-03  14.688232  NaN     2.985        NaN         NaN  17.738192
1306 1996-01-04  14.552561  NaN     2.965        NaN         NaN  17.676863
1307 1996-01-05  14.452592  NaN     2.960        NaN         NaN  17.577793

          INFY  LICI       ITC  ...  COOLCAPS  ALLETEC  20MICRONS  VIKASECO  \
1303  0.796679   NaN  5.583333  ...       NaN      NaN        NaN       NaN
1304  0.793457   NaN  5.372222  ...       NaN      NaN        NaN       NaN
1305  0.798828   NaN  5.200000  ...       NaN      NaN        NaN       NaN
1306  0.793554   NaN  5.297777  ...       NaN      NaN        NaN       NaN
1307  0.784179   NaN  5.202222  ...       NaN      NaN        NaN       NaN

      ORIENTBELL  DONEAR  CAREERP  INTLCONV  SHREEPUSHK  PRITIKAUTO
1303         NaN     NaN      NaN       NaN         NaN         NaN
1304         NaN     NaN      NaN       NaN         NaN         NaN
1305         NaN     NaN      NaN       NaN         NaN         NaN
1306         NaN     NaN      NaN       NaN         NaN         NaN
1307         NaN     NaN      NaN       NaN         NaN         NaN

[5 rows x 1418 columns]
Missing values percentage in each column:
 RELIANCE       0.000000
TCS            24.391968
HDFCBANK        0.000000
ICICIBANK      23.967760
BHARTIARTL     23.967760
                 ...
DONEAR         40.158371
CAREERP        53.379525
INTLCONV       91.558258
SHREEPUSHK     70.616516
PRITIKAUTO     91.204751
Length: 1417, dtype: float64
Numeric data after forward fill and dropping columns with too many missing values:
       RELIANCE  TCS  HDFCBANK  ICICIBANK  BHARTIARTL       SBIN      INFY  \
1303  14.691803  NaN     2.980        NaN         NaN  18.823240  0.796679
1304  14.577553  NaN     2.975        NaN         NaN  18.224106  0.793457
1305  14.688232  NaN     2.985        NaN         NaN  17.738192  0.798828
1306  14.552561  NaN     2.965        NaN         NaN  17.676863  0.793554
1307  14.452592  NaN     2.960        NaN         NaN  17.577793  0.784179

           ITC  HINDUNILVR  LT  ...  ASMS  BIRLAMONEY  NAGAFERT  GEECEE  \
1303  5.583333   61.805000 NaN  ...   NaN         NaN       NaN     NaN
1304  5.372222   62.465000 NaN  ...   NaN         NaN       NaN     NaN
1305  5.200000   62.095001 NaN  ...   NaN         NaN       NaN     NaN
1306  5.297777   62.099998 NaN  ...   NaN         NaN       NaN     NaN
1307  5.202222   62.000000 NaN  ...   NaN         NaN       NaN     NaN

      TCLCONS  REMSONSIND  PTL  20MICRONS  ORIENTBELL  DONEAR
1303      NaN         NaN  NaN        NaN         NaN     NaN
1304      NaN         NaN  NaN        NaN         NaN     NaN
1305      NaN         NaN  NaN        NaN         NaN     NaN
```

```
1306      NaN         NaN  NaN         NaN         NaN     NaN
1307      NaN         NaN  NaN         NaN         NaN     NaN

[5 rows x 693 columns]
Returns after pct_change:
      RELIANCE       TCS  HDFCBANK  ICICIBANK  BHARTIARTL      SBIN      INFY  \
4826 -0.012400  0.014251 -0.003757   0.020934    0.039045  0.007198  0.013159
4827 -0.016212 -0.023101 -0.006234  -0.015325   -0.080097 -0.035101 -0.005841
4828 -0.002337 -0.001788  0.015667   0.028277   -0.102186  0.011954 -0.004627
4829 -0.015369 -0.036633  0.010876  -0.021104   -0.001670 -0.026382 -0.025775
4830  0.008733 -0.034308  0.005752   0.004954   -0.069129  0.006780 -0.018622

          ITC  HINDUNILVR        LT  ...      ASMS  BIRLAMONEY  NAGAFERT  \
4826 -0.002146    0.010663 -0.014208  ... -0.005402    0.005006 -0.013216
4827  0.025801    0.006217 -0.006426  ... -0.021727   -0.013699 -0.010417
4828  0.037728    0.055046  0.015775  ... -0.003507   -0.007576  0.001504
4829 -0.010503   -0.005501 -0.001993  ... -0.002346   -0.022901  0.006006
4830  0.023678    0.025874 -0.015770  ...  0.001764   -0.009115  0.031343

        GEECEE   TCLCONS  REMSONSIND       PTL  20MICRONS  ORIENTBELL  \
4826 -0.006893 -0.015434    0.038889 -0.016204  -0.021505    0.010721
4827 -0.016551 -0.041033    0.000000  0.035294  -0.039072   -0.049180
4828  0.000000  0.008654    0.000000 -0.009091  -0.017789    0.034483
4829  0.010315  0.022402   -0.037433 -0.020642   0.050453    0.000000
4830  0.021494  0.092774    0.000000 -0.025761  -0.013547   -0.039216

        DONEAR
4826 -0.069800
4827 -0.035222
4828 -0.039683
4829 -0.001653
4830 -0.004967

[5 rows x 693 columns]
Correlation matrix:
             RELIANCE       TCS  HDFCBANK  ICICIBANK  BHARTIARTL      SBIN  \
RELIANCE     1.000000  0.264368  0.417298   0.417679    0.270239  0.385058
TCS          0.264368  1.000000  0.242183   0.221252    0.153991  0.165645
HDFCBANK     0.417298  0.242183  1.000000   0.546782    0.273927  0.446873
ICICIBANK    0.417679  0.221252  0.546782   1.000000    0.301742  0.661713
BHARTIARTL   0.270239  0.153991  0.273927   0.301742    1.000000  0.281013
...               ...       ...       ...        ...         ...       ...
REMSONSIND   0.110177  0.082815  0.084009   0.102341    0.050684  0.132629
PTL          0.146950  0.042442  0.122632   0.139065    0.081969  0.165196
20MICRONS    0.201034  0.102613  0.166876   0.222021    0.133770  0.223252
ORIENTBELL   0.194822  0.083892  0.187487   0.220236    0.113105  0.218453
DONEAR       0.141217  0.068376  0.126485   0.146742    0.094436  0.174994

                INFY       ITC  HINDUNILVR        LT  ...      ASMS  \
RELIANCE    0.242490  0.242099    0.238058  0.391762  ...  0.093106
TCS         0.549384  0.182497    0.186186  0.207219  ...  0.035708
HDFCBANK    0.249444  0.307567    0.253297  0.484910  ...  0.091334
ICICIBANK   0.231537  0.291341    0.197741  0.536150  ...  0.113482
BHARTIARTL  0.164874  0.202710    0.146826  0.289864  ...  0.065867
...              ...       ...         ...       ...  ...       ...
REMSONSIND  0.054334  0.063923    0.053120  0.106084  ...  0.022028
PTL         0.082678  0.101053    0.044703  0.154778  ...  0.111351
20MICRONS   0.109991  0.138228    0.086704  0.224453  ...  0.117974
ORIENTBELL  0.088562  0.133815    0.078349  0.219870  ...  0.127994
DONEAR      0.052887  0.104501    0.064448  0.174090  ...  0.126727
```

	BIRLAMONEY	NAGAFERT	GEECEE	TCLCONS	REMSONSIND	PTL	\
RELIANCE	0.184124	0.119252	0.182419	0.090990	0.110177	0.146950	
TCS	0.071835	0.014381	0.070952	0.064172	0.082815	0.042442	
HDFCBANK	0.176502	0.117987	0.167862	0.089121	0.084009	0.122632	
ICICIBANK	0.229340	0.101384	0.170756	0.114717	0.102341	0.139065	
BHARTIARTL	0.135084	0.062689	0.126762	0.063559	0.050684	0.081969	
...	...	...	...	...	...	...	
REMSONSIND	0.125261	0.061647	0.105841	0.010514	1.000000	0.082015	
PTL	0.173774	0.112207	0.142901	0.071300	0.082015	1.000000	
20MICRONS	0.233882	0.180755	0.179774	0.089629	0.094417	0.139325	
ORIENTBELL	0.209328	0.140628	0.195430	0.089401	0.066788	0.160015	
DONEAR	0.150979	0.133904	0.141383	0.103643	0.094757	0.129367	

	20MICRONS	ORIENTBELL	DONEAR
RELIANCE	0.201034	0.194822	0.141217
TCS	0.102613	0.083892	0.068376
HDFCBANK	0.166876	0.187487	0.126485
ICICIBANK	0.222021	0.220236	0.146742
BHARTIARTL	0.133770	0.113105	0.094436
...	...	...	...
REMSONSIND	0.094417	0.066788	0.094757
PTL	0.139325	0.160015	0.129367
20MICRONS	1.000000	0.203082	0.172593
ORIENTBELL	0.203082	1.000000	0.145906
DONEAR	0.172593	0.145906	1.000000

[693 rows x 693 columns]



```
Most Correlated Pairs:
 Series([], dtype: float64)
```

```
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Stock Return Correlation Matrix')
plt.show()

# Output the most correlated pairs
print("Most Correlated Pairs:\n", most_correlated)
```