In [1]:

```
# Importing necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

Quant Small Cap Fund

In [2]:

```
# Load Mutual Fund Data
Quant_Small_Cap_Fund = pd.read_csv("Quant Small Cap Fund.csv")
```

In [3]:

```
# Convert Date column to datetime format
Quant_Small_Cap_Fund['Date'] = pd.to_datetime(Quant_Small_Cap_Fund['Date'], format='%d-%m-%Y')
```

In [4]:

```
# Remove '%' and convert to float
Quant_Small_Cap_Fund['%Change'] = Quant_Small_Cap_Fund['%Change'].str.rstrip('%').astype(float)
```

In [5]:

```
# Display the sample of Mutual Fund
Quant_Small_Cap_Fund.head(10)
```

Out[5]:

	Date	Adj Close	%Change	Returns on 10000
0	2020-08-20	56.863	0.00	10000.00
1	2020-08-21	58.093	2.16	10216.20
2	2020-08-24	59.410	2.27	10447.76
3	2020-08-25	59.417	0.01	10449.00
4	2020-08-26	59.579	0.27	10477.48
5	2020-08-27	59.137	-0.74	10399.89
6	2020-08-28	58.319	-1.38	10256.02
7	2020-08-31	56.123	-3.77	9869.81
8	2020-09-01	56.718	1.06	9974.45
9	2020-09-02	58.371	2.91	10265.13

In [6]:
```
# Calcualte minimum of %Change
min_change = np.min(Quant_Small_Cap_Fund['%Change'])
print("Minimum %Change:", min_change)
```

```
Minimum %Change: -6.34
```

In [7]:
```
# Calcualte maximum of %Change
max_change = np.max(Quant_Small_Cap_Fund['%Change'])
print("Maximum %Change:", max_change)
```

```
Maximum %Change: 4.47
```

In [8]:
```
# Calculate arithmetic mean of %Change
arithmetic_mean = np.mean(Quant_Small_Cap_Fund['%Change'])
print("Arithmetic Mean of %Change:", arithmetic_mean)
```

```
Arithmetic Mean of %Change: 0.16882749326145555
```

In [9]:
```
# Calculate geometric mean of %Change
geometric_mean = np.exp(np.mean(np.log(1 + Quant_Small_Cap_Fund['%Change'] / 100))) - 1
print("Geometric Mean of %Change:", geometric_mean)
```

```
Geometric Mean of %Change: 0.0016046163086964604
```

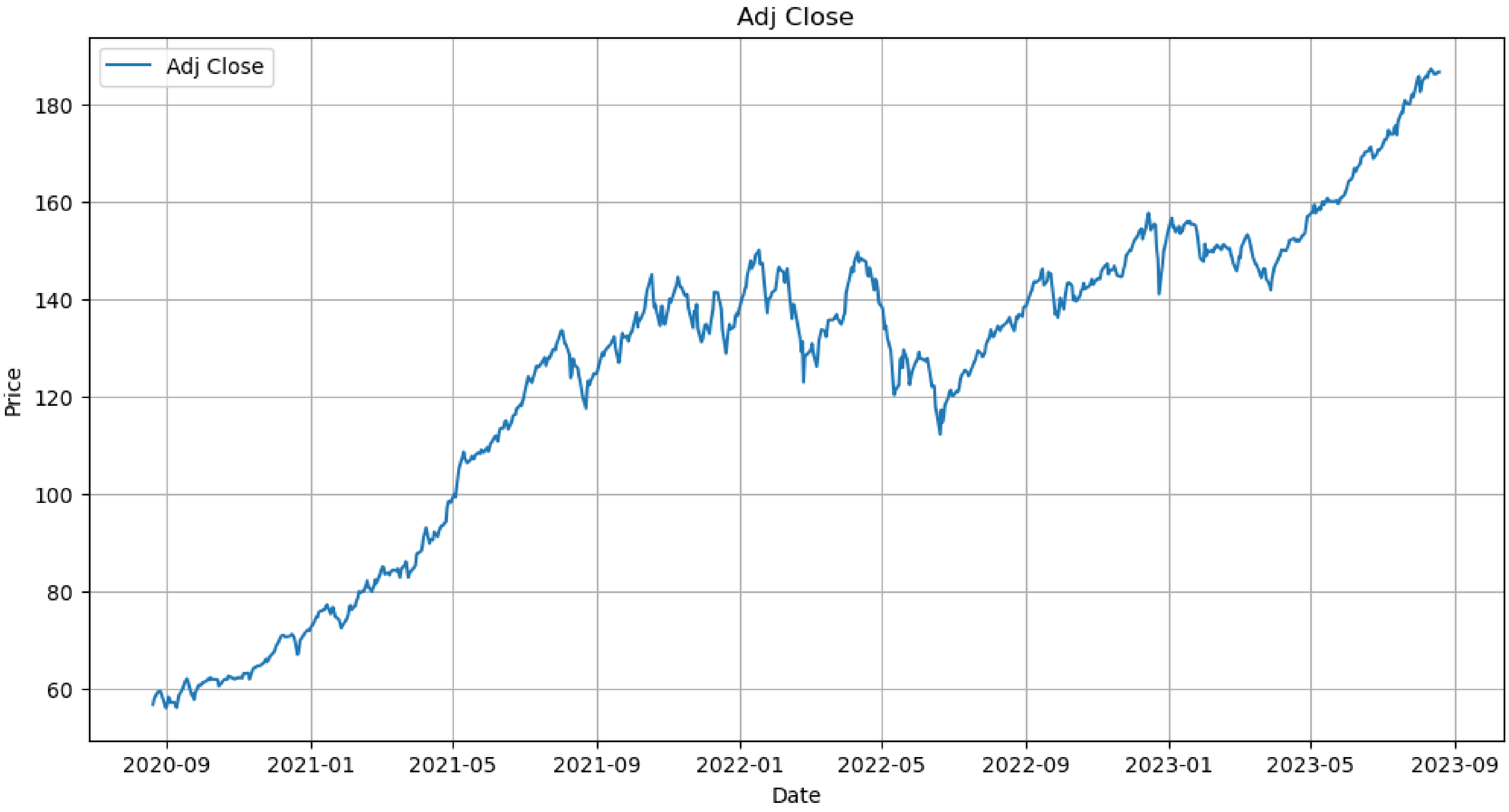
In [10]:
```
# Calculate standard deviation of %Change
std_deviation_change = np.std(Quant_Small_Cap_Fund['%Change'])
print("Standard Deviation of %Change:", std_deviation_change)
```

```
Standard Deviation of %Change: 1.2897843903055752
```

In [11]:
```
# Load Mutual Fund Holdings Data
Quant_Small_Cap_Fund_Holdings = pd.read_csv("Quant Small Cap Fund Top 10 Equity Holdings.csv")
```

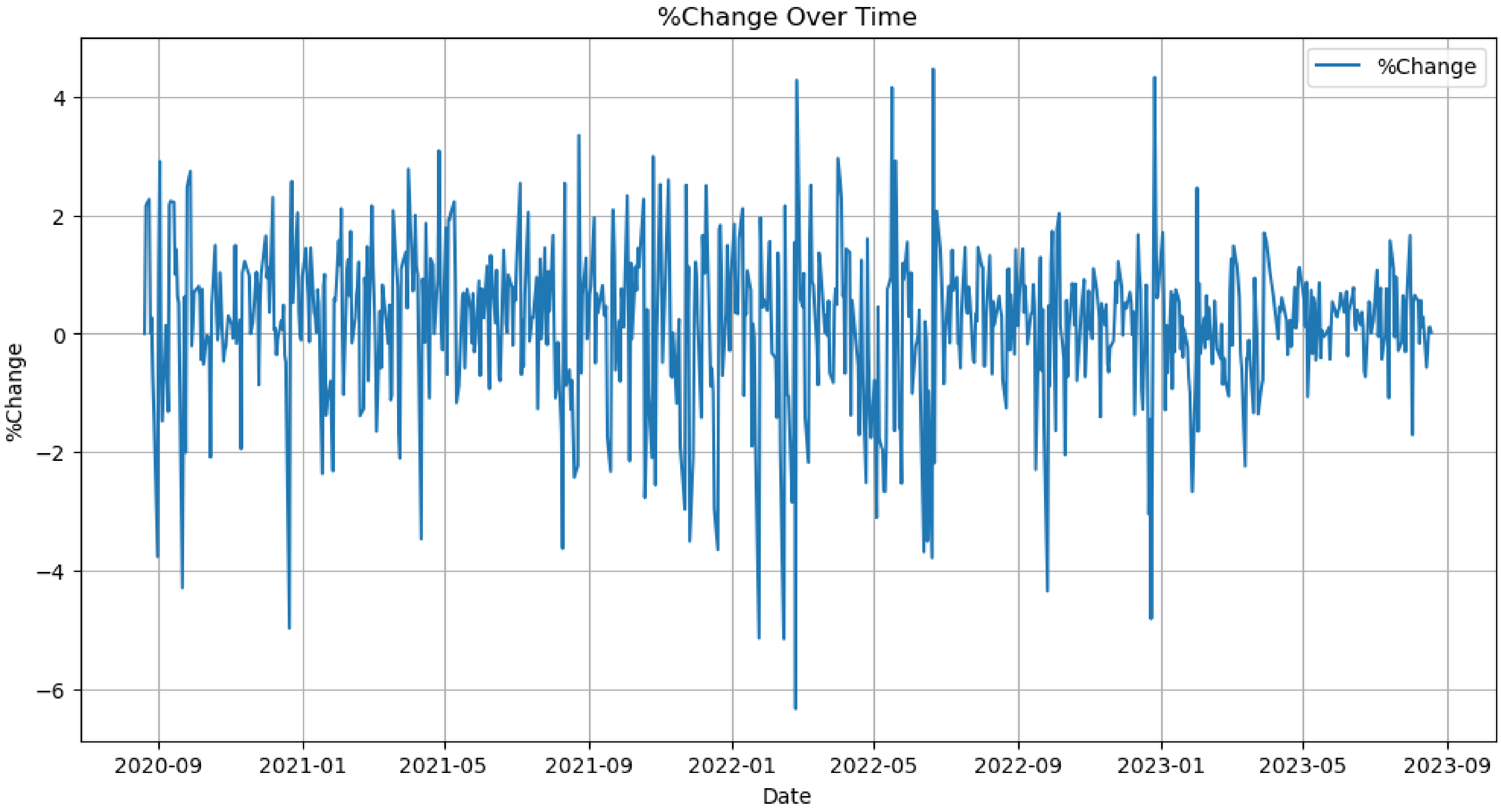
In [12]:
```
# Remove '%' and convert to float
Quant_Small_Cap_Fund_Holdings['Weightage'] = Quant_Small_Cap_Fund_Holdings['Weightage'].str.rstrip('%').astype(float)
```

```
In [13]: # Plot 1: Adj Close Over Time
         plt.figure(figsize=(12, 6))
         plt.plot(Quant_Small_Cap_Fund['Date'], Quant_Small_Cap_Fund['Adj Close'], label='Adj Close')
         plt.xlabel('Date')
         plt.ylabel('Price')
         plt.title('Adj Close')
         plt.legend()
         plt.grid()
         plt.show()
```

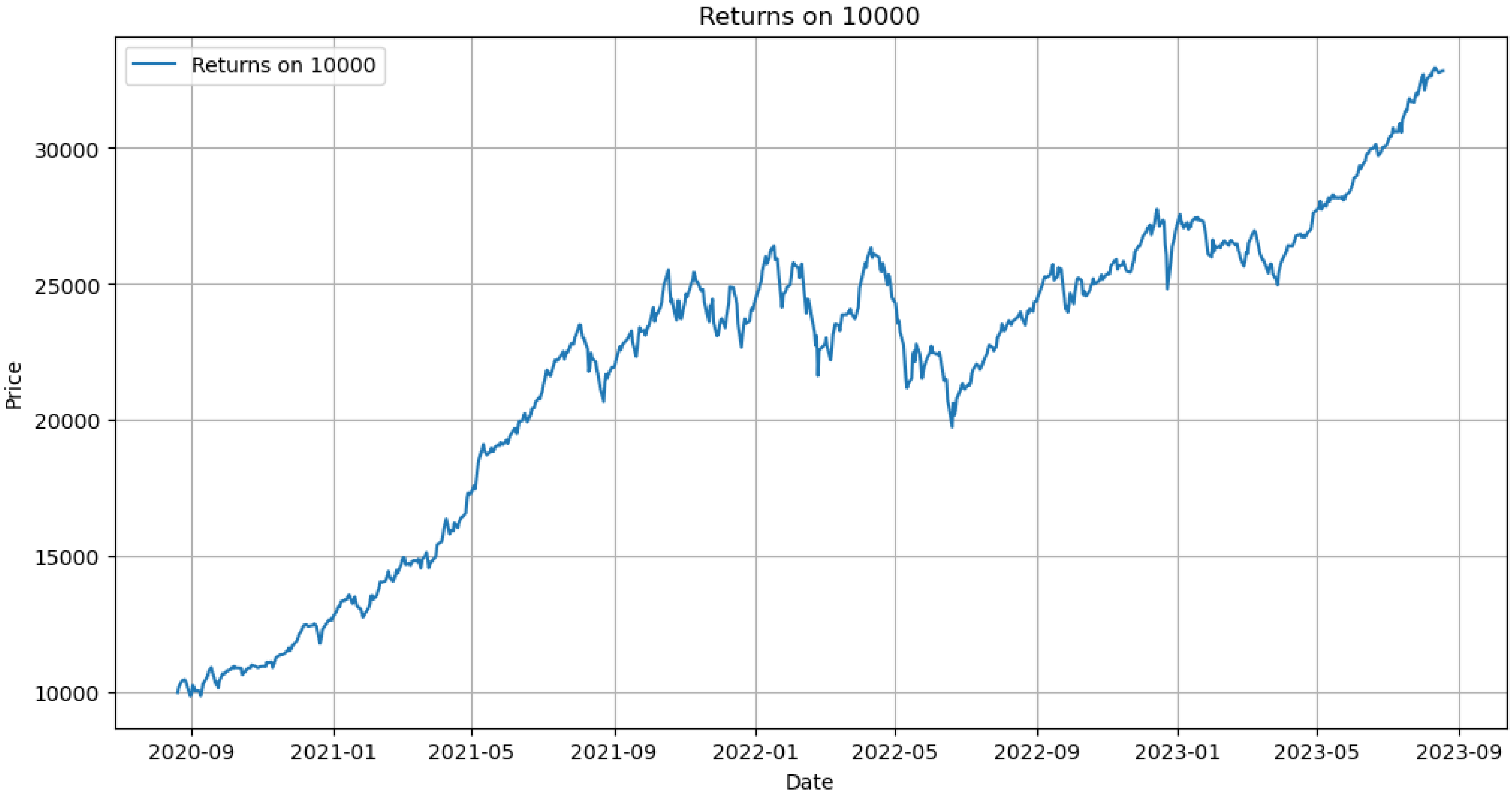


In [14]:

```
# Plot 2: %Change Over Time
plt.figure(figsize=(12, 6))
plt.plot(Quant_Small_Cap_Fund['Date'], Quant_Small_Cap_Fund['%Change'], label='%Change')
plt.xlabel('Date')
plt.ylabel('%Change')
plt.title('%Change Over Time')
plt.legend()
plt.grid()
plt.show()
```

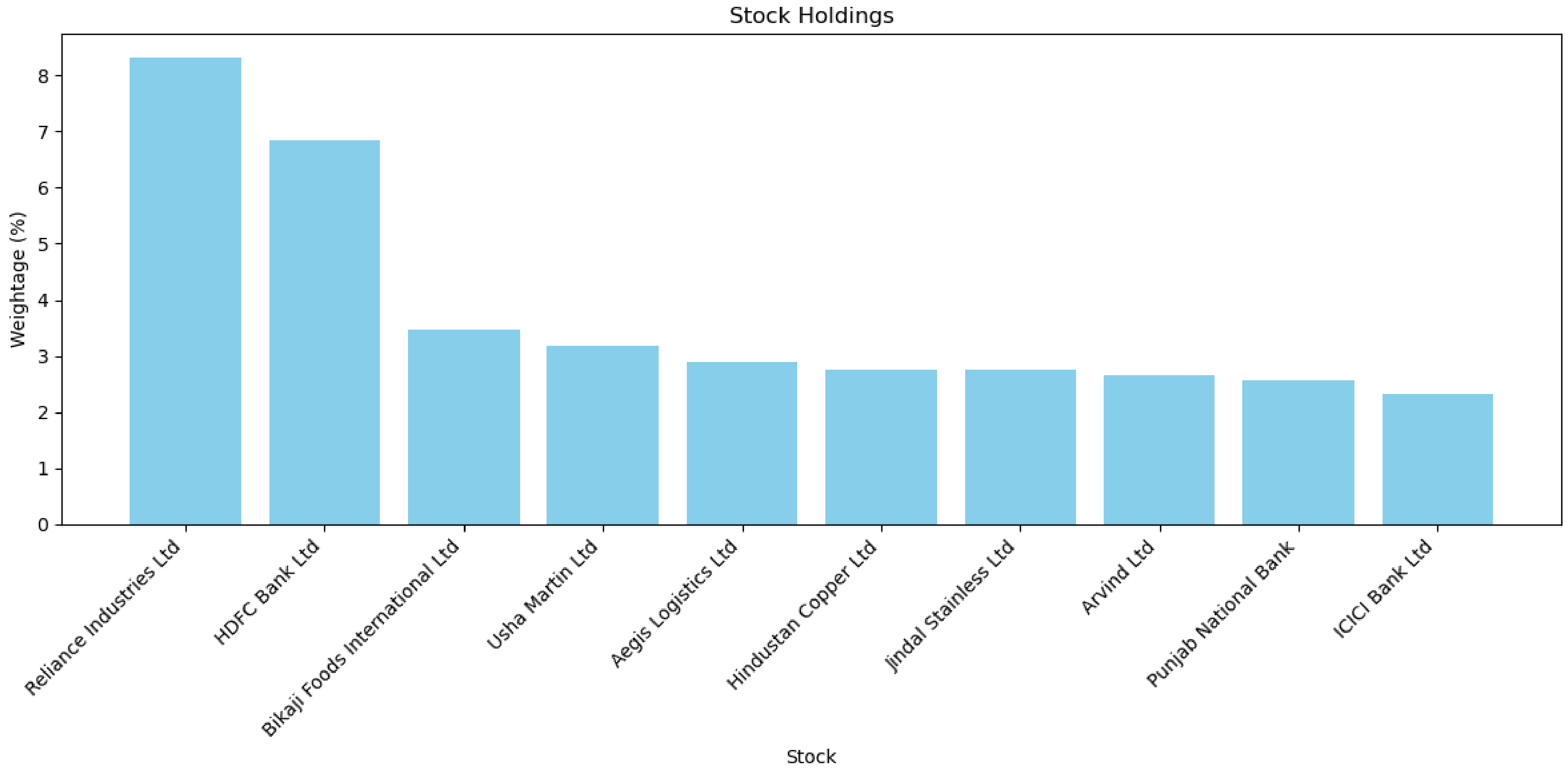


```
In [15]: # Plot 3: Returns on 10000 Over Time
         plt.figure(figsize=(12, 6))
         plt.plot(Quant_Small_Cap_Fund['Date'], Quant_Small_Cap_Fund['Returns on 10000'], label='Returns on 10000')
         plt.xlabel('Date')
         plt.ylabel('Price')
         plt.title('Returns on 10000')
         plt.legend()
         plt.grid()
         plt.show()
```



In [16]:

```
# Plot 4: Top 10 Stocks Holdings
plt.figure(figsize=(12, 6))
plt.bar(Quant_Small_Cap_Fund_Holdings['Stock'], Quant_Small_Cap_Fund_Holdings['Weightage'], color='skyblue')
plt.xticks(rotation=45, ha='right')
plt.xlabel('Stock')
plt.ylabel('Weightage (%)')
plt.title('Stock Holdings')
plt.tight_layout()
```



In [1]:
```
# Importing necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

Quant Mid Cap Fund

In [17]:
```
# Load Mutual Fund Data
Quant_Mid_Cap_Fund = pd.read_csv("Quant Mid Cap Fund.csv")
```

In [18]:
```
# Convert Date column to datetime format
Quant_Mid_Cap_Fund['Date'] = pd.to_datetime(Quant_Mid_Cap_Fund['Date'], format='%d-%m-%Y')
```

In [19]:
```
# Remove '%' and convert to float
Quant_Mid_Cap_Fund['%Change'] = Quant_Mid_Cap_Fund['%Change'].str.rstrip('%').astype(float)
```

In [20]:
```
# Display the sample of Mutual Fund
Quant_Mid_Cap_Fund.head(10)
```

Out[20]:

	Date	Adj Close	%Change	Returns on 10000
0	2020-08-20	64.564	0.00	10000.00
1	2020-08-21	65.117	0.86	10085.62
2	2020-08-24	65.693	0.89	10174.97
3	2020-08-25	65.569	-0.19	10155.77
4	2020-08-26	65.718	0.23	10178.80
5	2020-08-27	65.294	-0.64	10113.18
6	2020-08-28	65.148	-0.22	10090.44
7	2020-08-31	62.476	-4.10	9676.58
8	2020-09-01	63.122	1.03	9776.72
9	2020-09-02	64.091	1.54	9926.80

In [21]:

```
# Calcualte minimum of %Change
min_change = np.min(Quant_Mid_Cap_Fund['%Change'])
print("Minimum %Change:", min_change)
```

```
Minimum %Change: -5.82
```

In [22]:

```
# Calcualte maximum of %Change
max_change = np.max(Quant_Mid_Cap_Fund['%Change'])
print("Maximum %Change:", max_change)
```

```
Maximum %Change: 4.15
```

In [23]:

```
# Calculate arithmetic mean of %Change
arithmetic_mean = np.mean(Quant_Mid_Cap_Fund['%Change'])
print("Arithmetic Mean of %Change:", arithmetic_mean)
```

```
Arithmetic Mean of %Change: 0.13745283018867926
```

In [24]:

```
# Calculate geometric mean of %Change
geometric_mean = np.exp(np.mean(np.log(1 + Quant_Mid_Cap_Fund['%Change'] / 100))) - 1
print("Geometric Mean of %Change:", geometric_mean)
```

```
Geometric Mean of %Change: 0.001308236326515111
```

In [25]:

```
# Calculate standard deviation of %Change
std_deviation_change = np.std(Quant_Mid_Cap_Fund['%Change'])
print("Standard Deviation of %Change:", std_deviation_change)
```

```
Standard Deviation of %Change: 1.1488040692624597
```

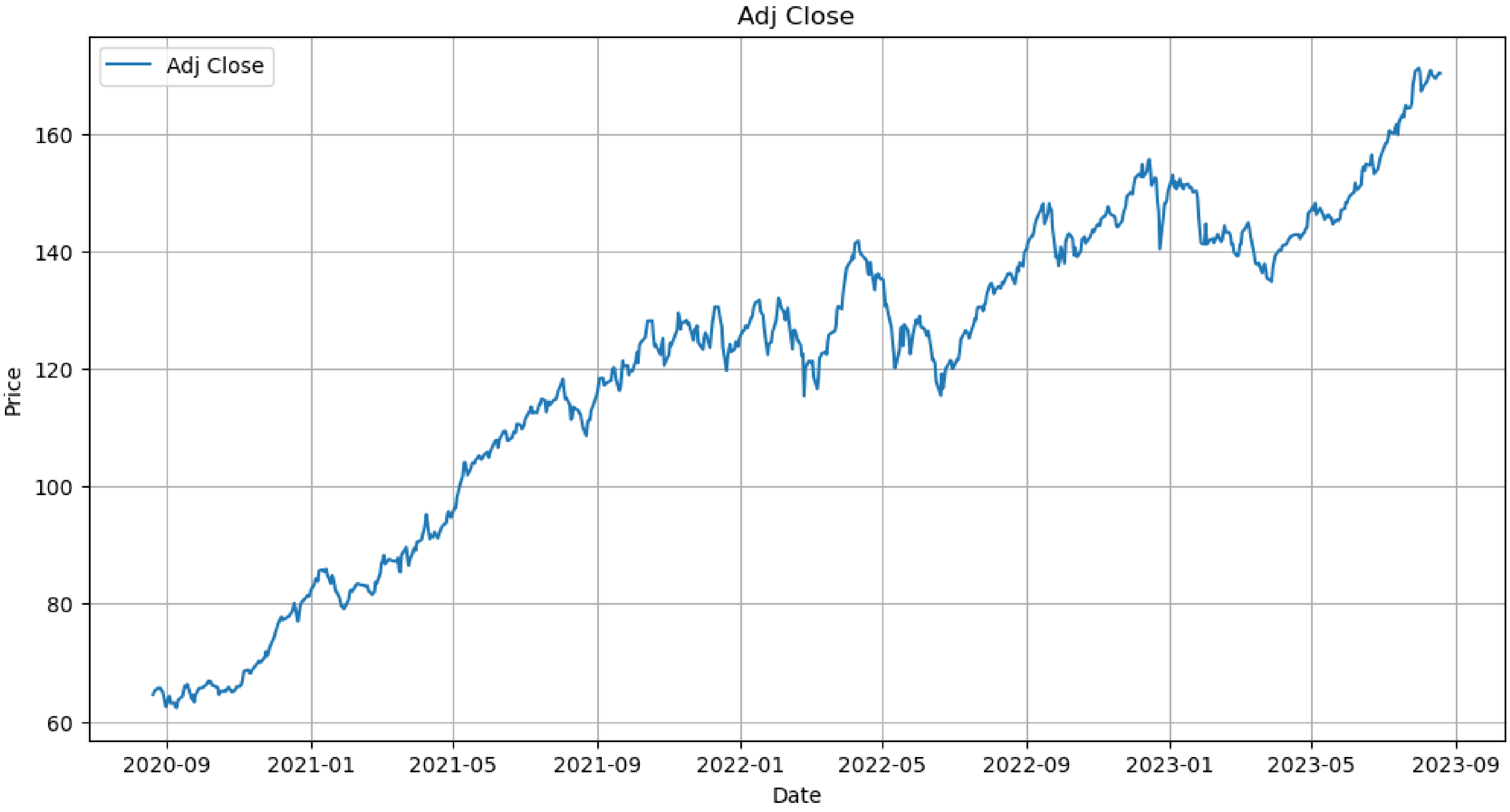
In [26]:

```
# Load Mutual Fund Holdings Data
Quant_Mid_Cap_Fund_Holdings = pd.read_csv("Quant Mid Cap Fund Top 10 Equity Holdings.csv")
```

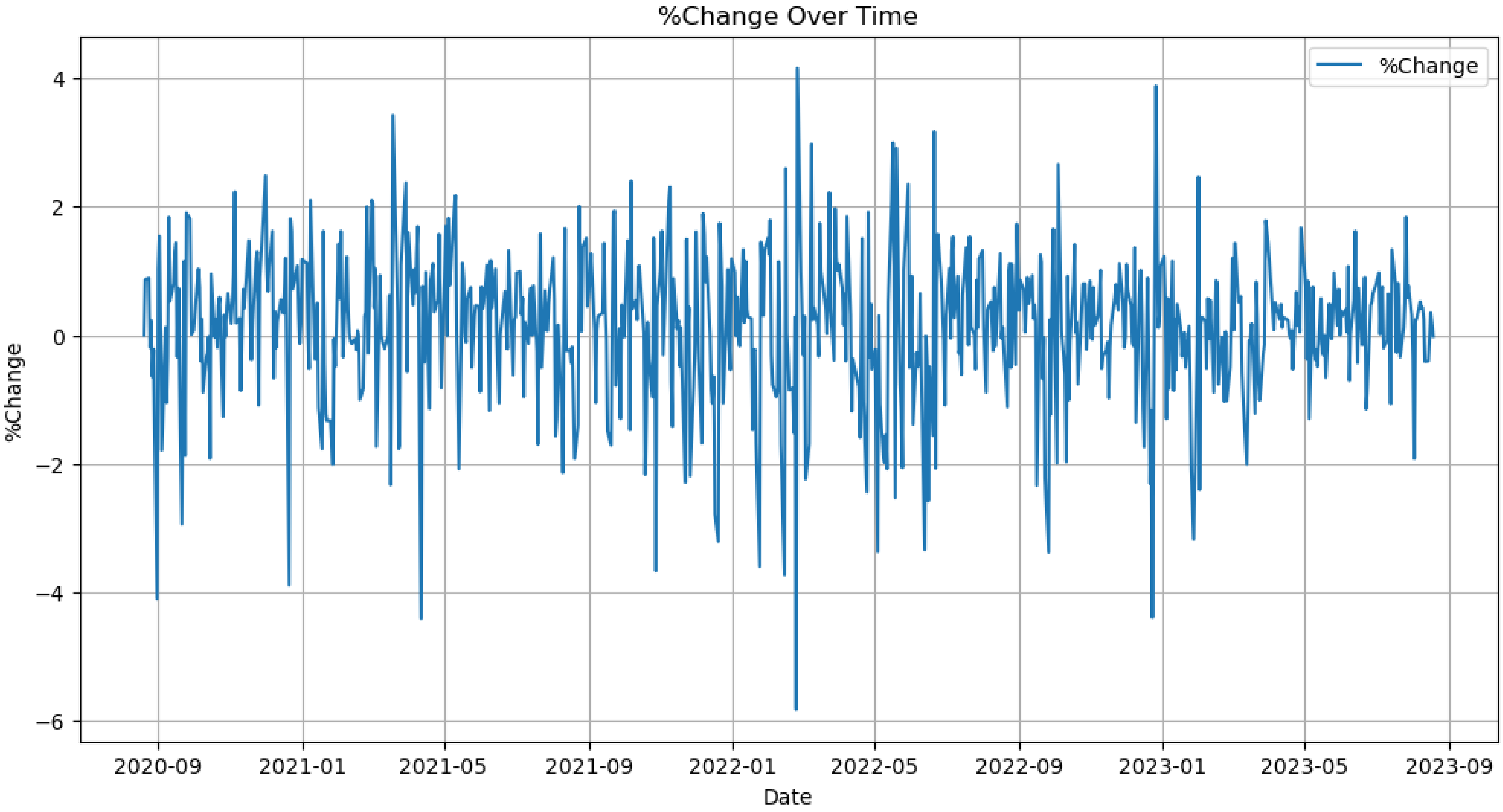
In [27]:

```
# Remove '%' and convert to float
Quant_Mid_Cap_Fund_Holdings['Weightage'] = Quant_Mid_Cap_Fund_Holdings['Weightage'].str.rstrip('%').astype(float)
```

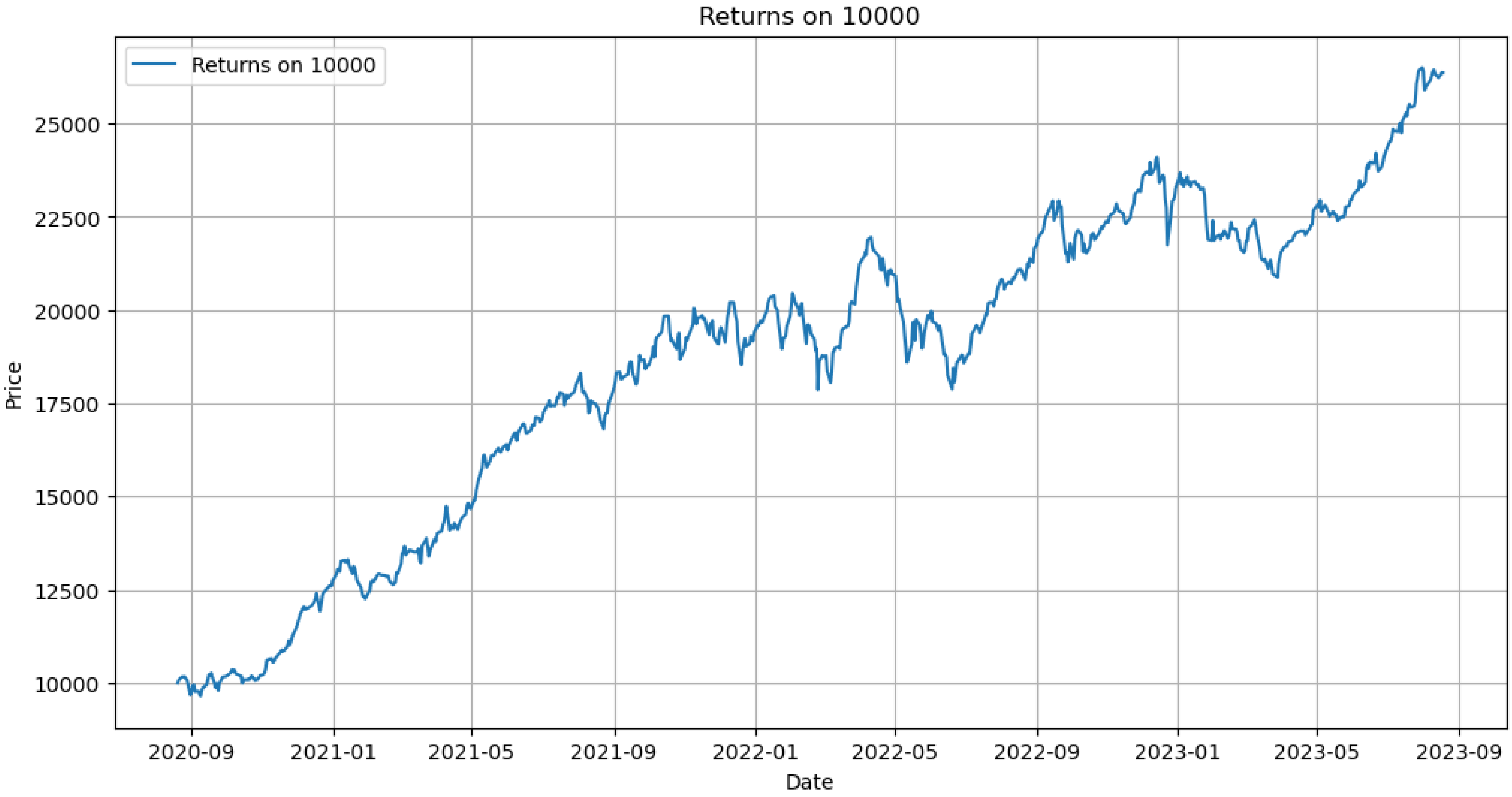
In [29]:
```
# Plot 1: Adj Close Over Time
plt.figure(figsize=(12, 6))
plt.plot(Quant_Mid_Cap_Fund['Date'], Quant_Mid_Cap_Fund['Adj Close'], label='Adj Close')
plt.xlabel('Date')
plt.ylabel('Price')
plt.title('Adj Close')
plt.legend()
plt.grid()
plt.show()
```



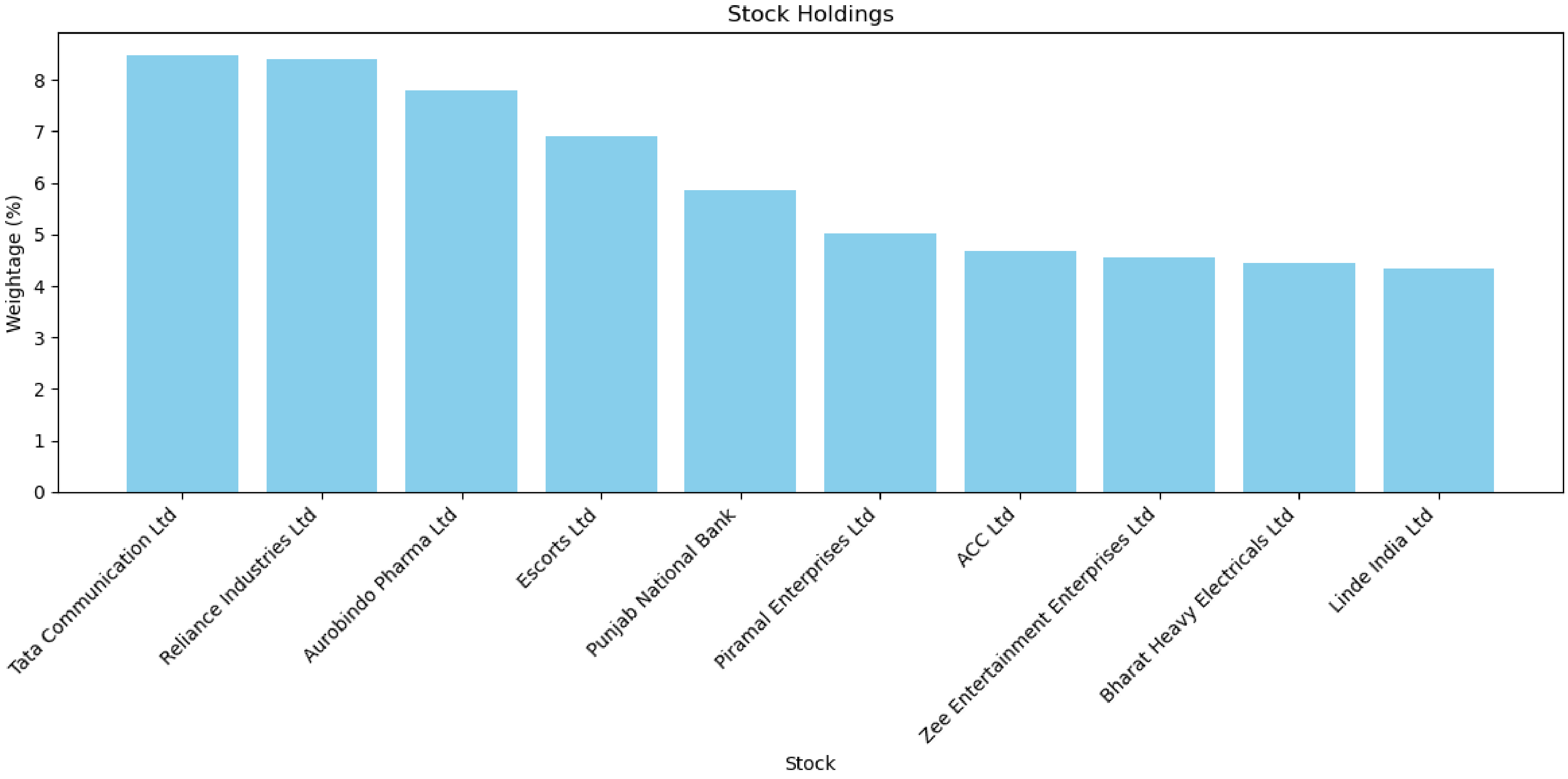
```
In [30]: # Plot 2: %Change Over Time
plt.figure(figsize=(12, 6))
plt.plot(Quant_Mid_Cap_Fund['Date'], Quant_Mid_Cap_Fund['%Change'], label='%Change')
plt.xlabel('Date')
plt.ylabel('%Change')
plt.title('%Change Over Time')
plt.legend()
plt.grid()
plt.show()
```



In [31]:
```
# Plot 3: Returns on 10000 Over Time
plt.figure(figsize=(12, 6))
plt.plot(Quant_Mid_Cap_Fund['Date'], Quant_Mid_Cap_Fund['Returns on 10000'], label='Returns on 10000')
plt.xlabel('Date')
plt.ylabel('Price')
plt.title('Returns on 10000')
plt.legend()
plt.grid()
plt.show()
```



In [32]:
```
# Plot 4: Top 10 Stocks Holdings
plt.figure(figsize=(12, 6))
plt.bar(Quant_Mid_Cap_Fund_Holdings['Stock'], Quant_Mid_Cap_Fund_Holdings['Weightage'], color='skyblue')
plt.xticks(rotation=45, ha='right')
plt.xlabel('Stock')
plt.ylabel('Weightage (%)')
plt.title('Stock Holdings')
plt.tight_layout()
```



In [1]:

```
# Importing necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

UTI Nifty 50 Index Fund

In [33]:

```
# Load Mutual Fund Data
UTI_Nifty_50_Index_Fund = pd.read_csv("UTI Nifty 50 Index Fund.csv")
```

In [34]:

```
# Convert Date column to datetime format
UTI_Nifty_50_Index_Fund['Date'] = pd.to_datetime(UTI_Nifty_50_Index_Fund['Date'], format='%d-%m-%Y')
```

In [35]:

```
# Remove '%' and convert to float
UTI_Nifty_50_Index_Fund['%Change'] = UTI_Nifty_50_Index_Fund['%Change'].str.rstrip('%').astype(float)
```

In [36]:

```
# Display the sample of Mutual Fund
UTI_Nifty_50_Index_Fund.head(10)
```

Out[36]:

	Date	Adj Close	%Change	Returns on 10000
0	2020-08-20	75.159	0.00	10000.00
1	2020-08-21	75.552	0.52	10052.30
2	2020-08-24	76.181	0.83	10135.91
3	2020-08-25	76.219	0.05	10140.99
4	2020-08-26	76.749	0.70	10211.54
5	2020-08-27	76.813	0.08	10220.04
6	2020-08-28	77.398	0.76	10297.82
7	2020-08-31	75.674	-2.23	10068.44
8	2020-09-01	76.221	0.72	10141.30
9	2020-09-02	76.666	0.58	10200.51

In [37]:
```
# Calcualte minimum of %Change
min_change = np.min(UTI_Nifty_50_Index_Fund['%Change'])
print("Minimum %Change:", min_change)
```

```
Minimum %Change: -4.77
```

In [38]:
```
# Calcualte maximum of %Change
max_change = np.max(UTI_Nifty_50_Index_Fund['%Change'])
print("Maximum %Change:", max_change)
```

```
Maximum %Change: 4.74
```

In [39]:
```
# Calculate arithmetic mean of %Change
arithmetic_mean = np.mean(UTI_Nifty_50_Index_Fund['%Change'])
print("Arithmetic Mean of %Change:", arithmetic_mean)
```

```
Arithmetic Mean of %Change: 0.08072776280323452
```

In [40]:
```
# Calculate geometric mean of %Change
geometric_mean = np.exp(np.mean(np.log(1 + UTI_Nifty_50_Index_Fund['%Change'] / 100))) - 1
print("Geometric Mean of %Change:", geometric_mean)
```

```
Geometric Mean of %Change: 0.0007609841649967031
```

In [41]:
```
# Calculate standard deviation of %Change
std_deviation_change = np.std(UTI_Nifty_50_Index_Fund['%Change'])
print("Standard Deviation of %Change:", std_deviation_change)
```

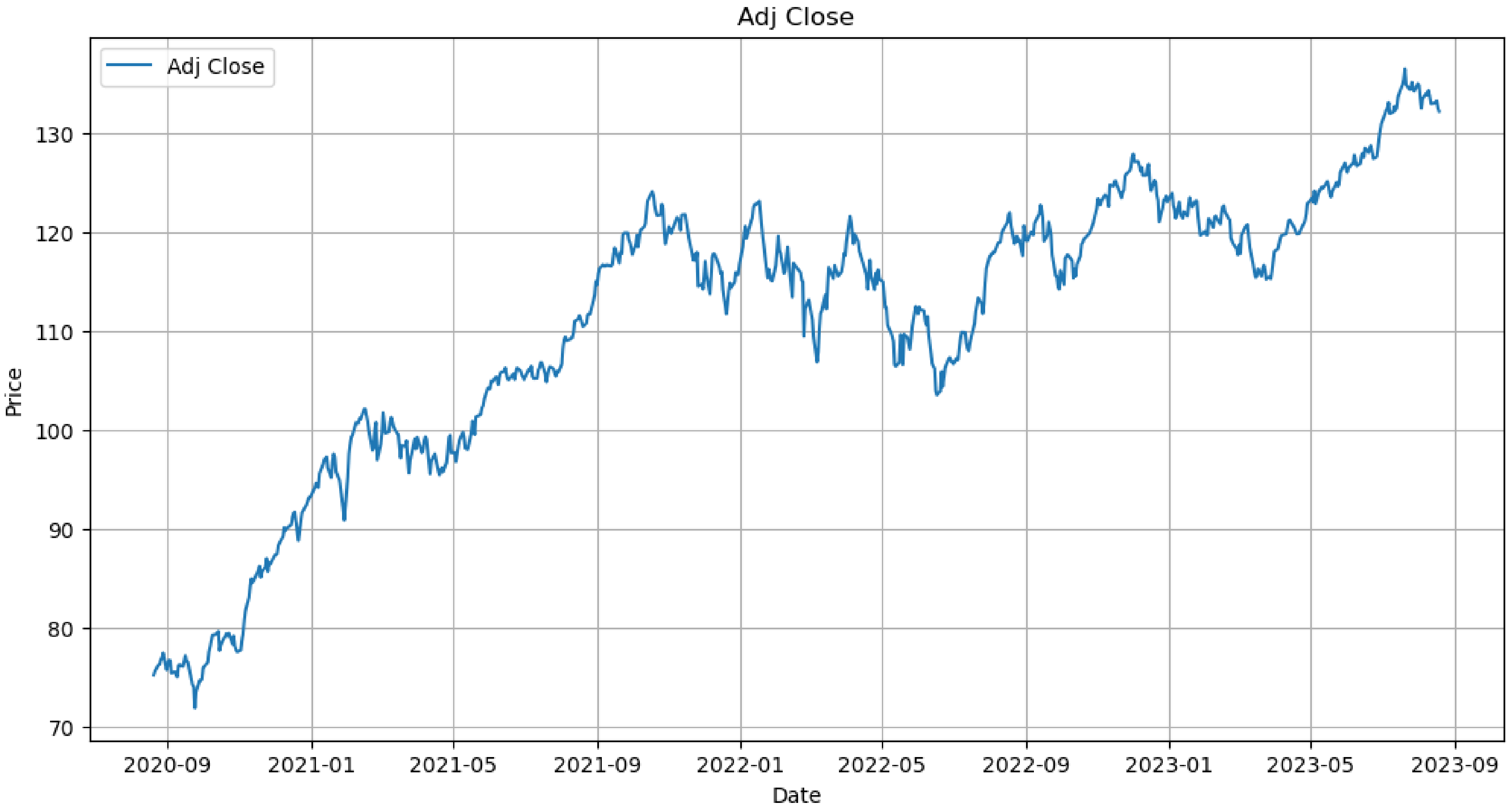
```
Standard Deviation of %Change: 0.9612494725764061
```

In [42]:
```
# Load Mutual Fund Holdings Data
UTI_Nifty_50_Index_Fund_Holdings = pd.read_csv("UTI Nifty 50 Index Fund Top 10 Equity Holdings.csv")
```

In [43]:
```
# Remove '%' and convert to float
UTI_Nifty_50_Index_Fund_Holdings['Weightage'] = UTI_Nifty_50_Index_Fund_Holdings['Weightage'].str.rstrip('%').astype(float)
```

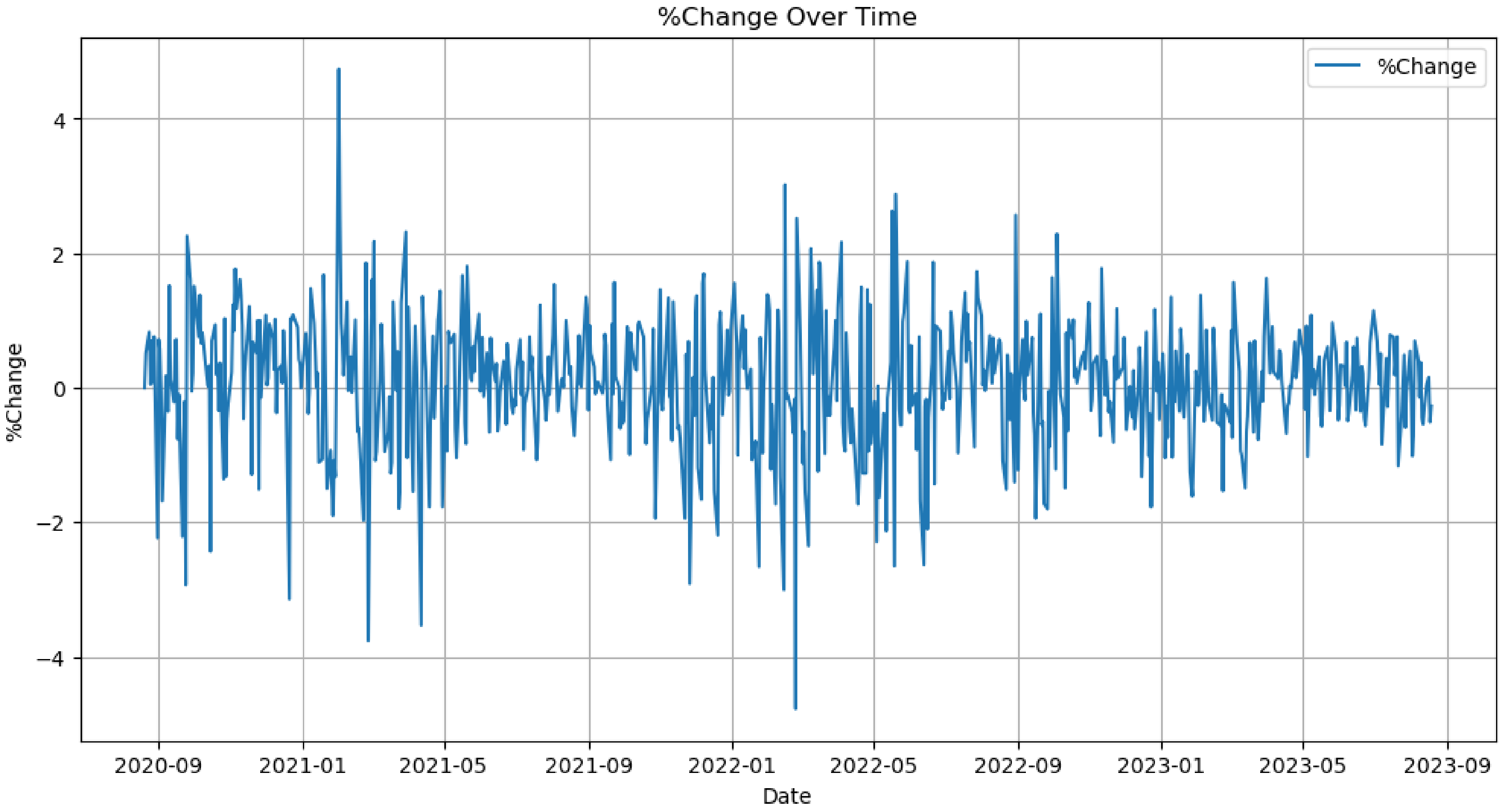
In [45]:

```
# Plot 1: Adj Close Over Time
plt.figure(figsize=(12, 6))
plt.plot(UTI_Nifty_50_Index_Fund['Date'], UTI_Nifty_50_Index_Fund['Adj Close'], label='Adj Close')
plt.xlabel('Date')
plt.ylabel('Price')
plt.title('Adj Close')
plt.legend()
plt.grid()
plt.show()
```

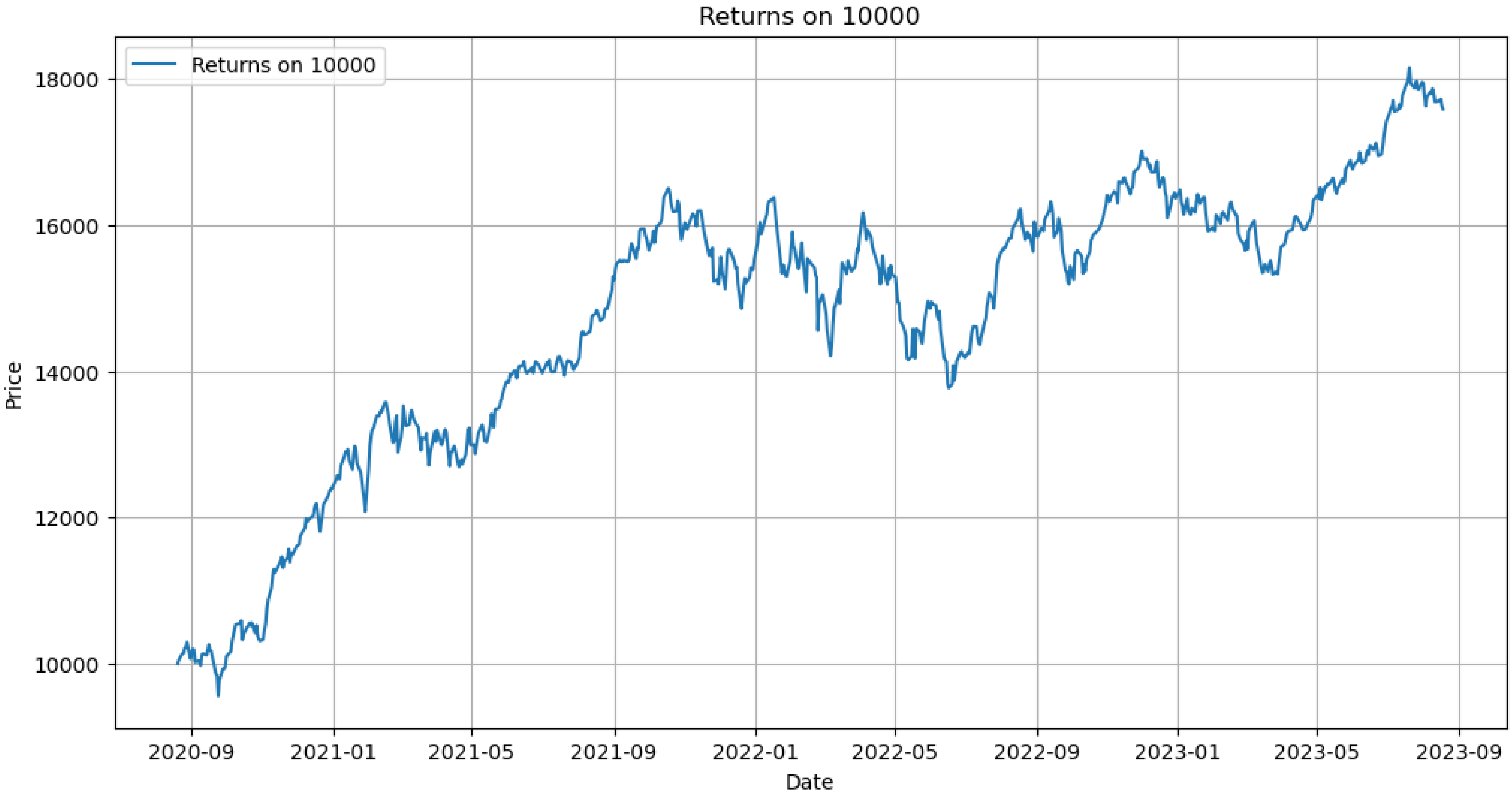


In [46]:

```
# Plot 2: %Change Over Time
plt.figure(figsize=(12, 6))
plt.plot(UTI_Nifty_50_Index_Fund['Date'], UTI_Nifty_50_Index_Fund['%Change'], label='%Change')
plt.xlabel('Date')
plt.ylabel('%Change')
plt.title('%Change Over Time')
plt.legend()
plt.grid()
plt.show()
```

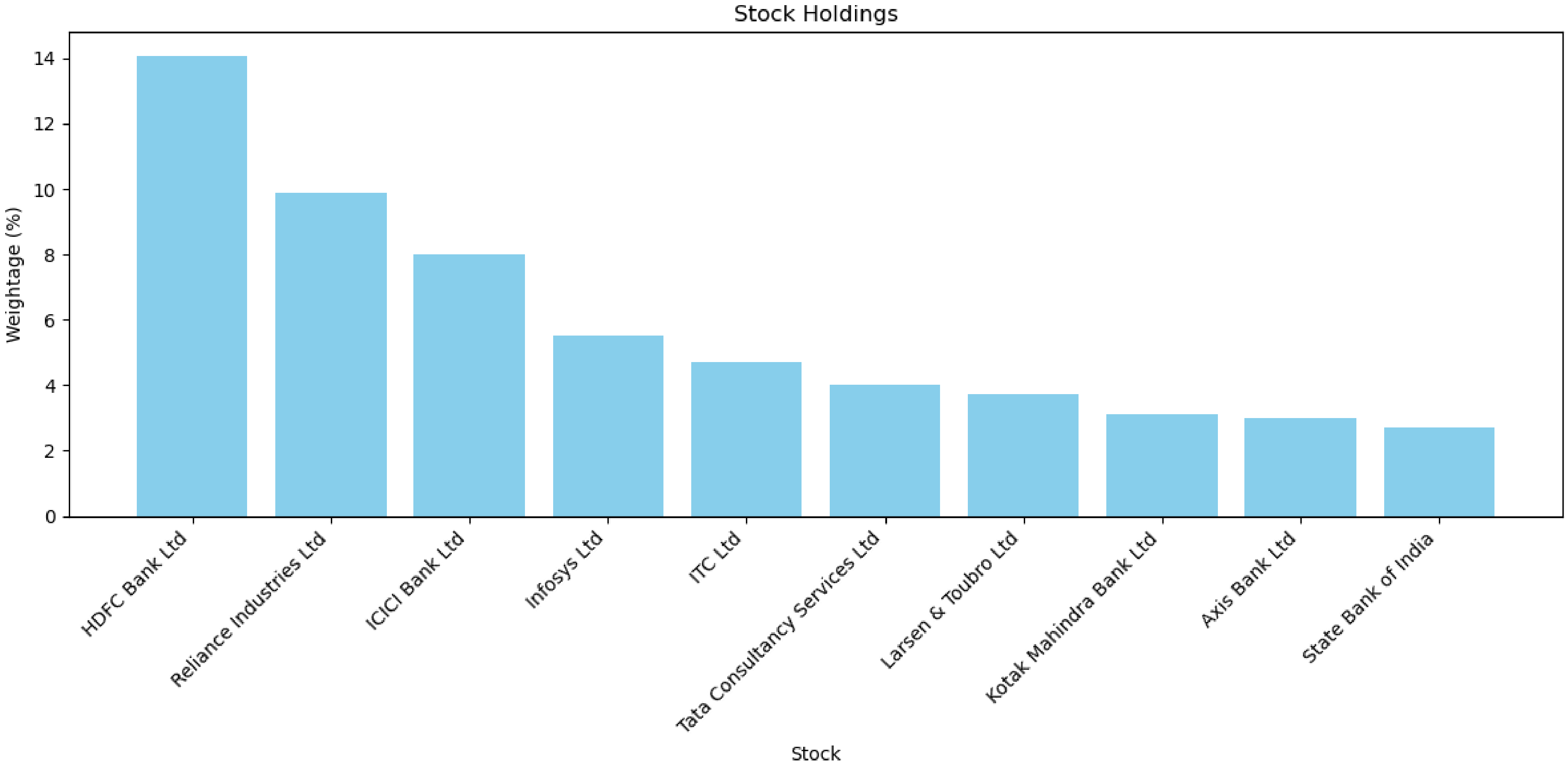


In [47]:
```
# Plot 3: Returns on 10000 Over Time
plt.figure(figsize=(12, 6))
plt.plot(UTI_Nifty_50_Index_Fund['Date'], UTI_Nifty_50_Index_Fund['Returns on 10000'], label='Returns on 10000')
plt.xlabel('Date')
plt.ylabel('Price')
plt.title('Returns on 10000')
plt.legend()
plt.grid()
plt.show()
```



In [48]:

```
# Plot 4: Top 10 Stocks Holdings
plt.figure(figsize=(12, 6))
plt.bar(UTI_Nifty_50_Index_Fund_Holdings['Stock'], UTI_Nifty_50_Index_Fund_Holdings['Weightage'], color='skyblue')
plt.xticks(rotation=45, ha='right')
plt.xlabel('Stock')
plt.ylabel('Weightage (%)')
plt.title('Stock Holdings')
plt.tight_layout()
```



In [1]:

```
# Importing necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

Nippon India Banking & Financial Services Fund

In [49]:

```
# Load Mutual Fund Data
Nippon_India_Banking_and_Financial_Services_Fund = pd.read_csv("Nippon India Banking & Financial Services Fund.csv")
```

In [50]:

```
# Convert Date column to datetime format
Nippon_India_Banking_and_Financial_Services_Fund['Date'] = pd.to_datetime(Nippon_India_Banking_and_Financial_Services_Fund['Date
```

In [51]:

```
# Remove '%' and convert to float
Nippon_India_Banking_and_Financial_Services_Fund['%Change'] = Nippon_India_Banking_and_Financial_Services_Fund['%Change'].str.rs
```

In [52]:

```
# Display the sample of Mutual Fund
Nippon_India_Banking_and_Financial_Services_Fund.head(10)
```

Out[52]:

	Date	Adj Close	%Change	Returns on 10000
0	2020-08-20	208.943	0.00	10000.00
1	2020-08-21	210.995	0.98	10098.19
2	2020-08-24	214.485	1.65	10265.24
3	2020-08-25	217.673	1.49	10417.80
4	2020-08-26	220.120	1.12	10534.95
5	2020-08-27	221.485	0.62	10600.28
6	2020-08-28	228.205	3.03	10921.86
7	2020-08-31	220.335	-3.45	10545.21
8	2020-09-01	220.811	0.22	10567.99
9	2020-09-02	221.308	0.23	10591.79

In [53]:

```
# Calcualte minimum of %Change
min_change = np.min(Nippon_India_Banking_and_Financial_Services_Fund['%Change'])
print("Minimum %Change:", min_change)
```

```
Minimum %Change: -5.84
```

In [54]:

```
# Calcualte maximum of %Change
max_change = np.max(Nippon_India_Banking_and_Financial_Services_Fund['%Change'])
print("Maximum %Change:", max_change)
```

```
Maximum %Change: 7.14
```

In [55]:

```
# Calculate arithmetic mean of %Change
arithmetic_mean = np.mean(Nippon_India_Banking_and_Financial_Services_Fund['%Change'])
print("Arithmetic Mean of %Change:", arithmetic_mean)
```

```
Arithmetic Mean of %Change: 0.12004043126684637
```

In [56]:

```
# Calculate geometric mean of %Change
geometric_mean = np.exp(np.mean(np.log(1 + Nippon_India_Banking_and_Financial_Services_Fund['%Change'] / 100))) - 1
print("Geometric Mean of %Change:", geometric_mean)
```

```
Geometric Mean of %Change: 0.0011235768263588852
```

In [57]:

```
# Calculate standard deviation of %Change
std_deviation_change = np.std(Nippon_India_Banking_and_Financial_Services_Fund['%Change'])
print("Standard Deviation of %Change:", std_deviation_change)
```

```
Standard Deviation of %Change: 1.2383858825312257
```

In [58]:

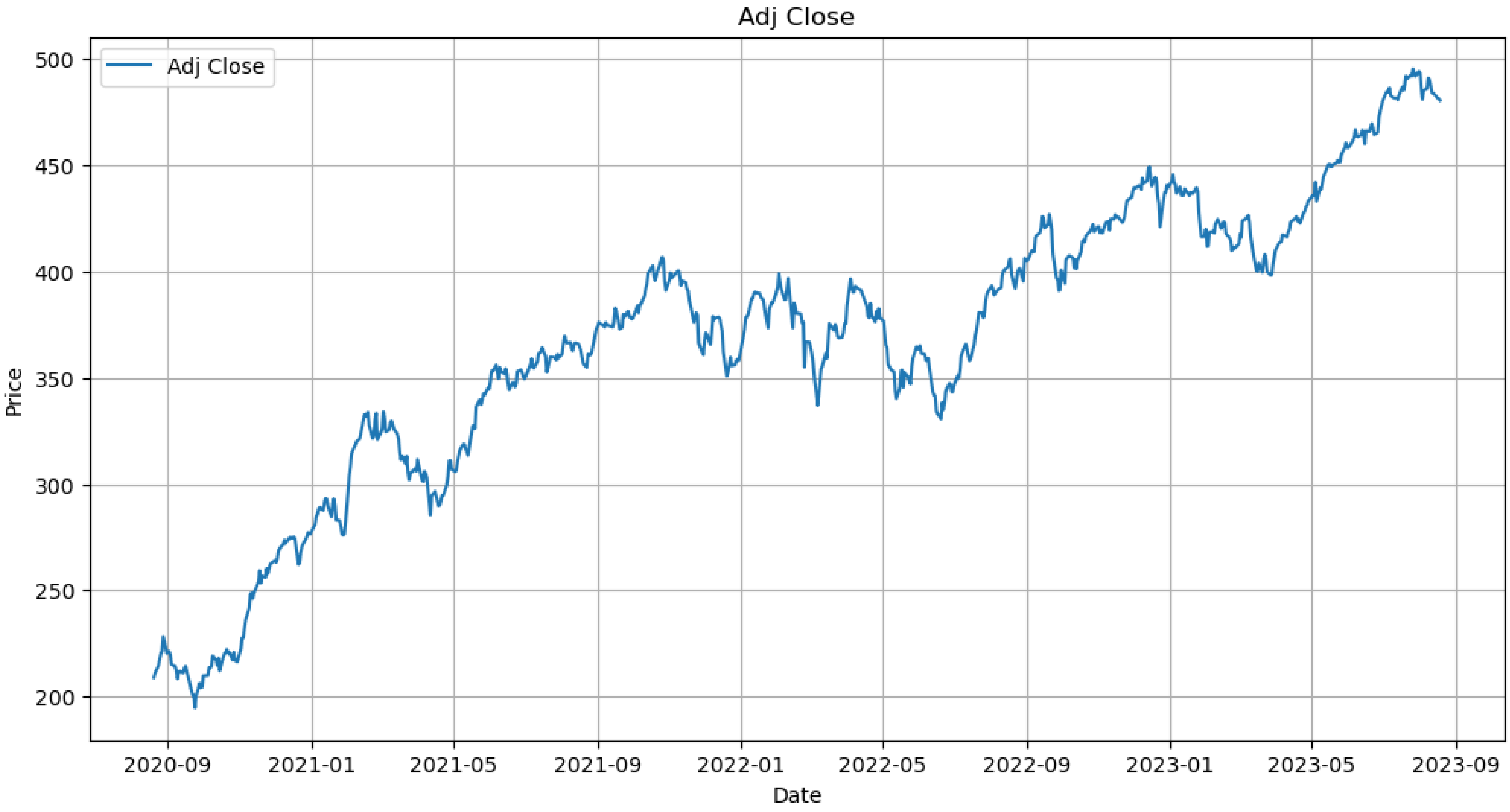
```
# Load Mutual Fund Holdings Data
Nippon_India_Banking_and_Financial_Services_Fund_Holdings = pd.read_csv("Nippon India Banking & Financial Services Fund Top 10 E
```

In [59]:

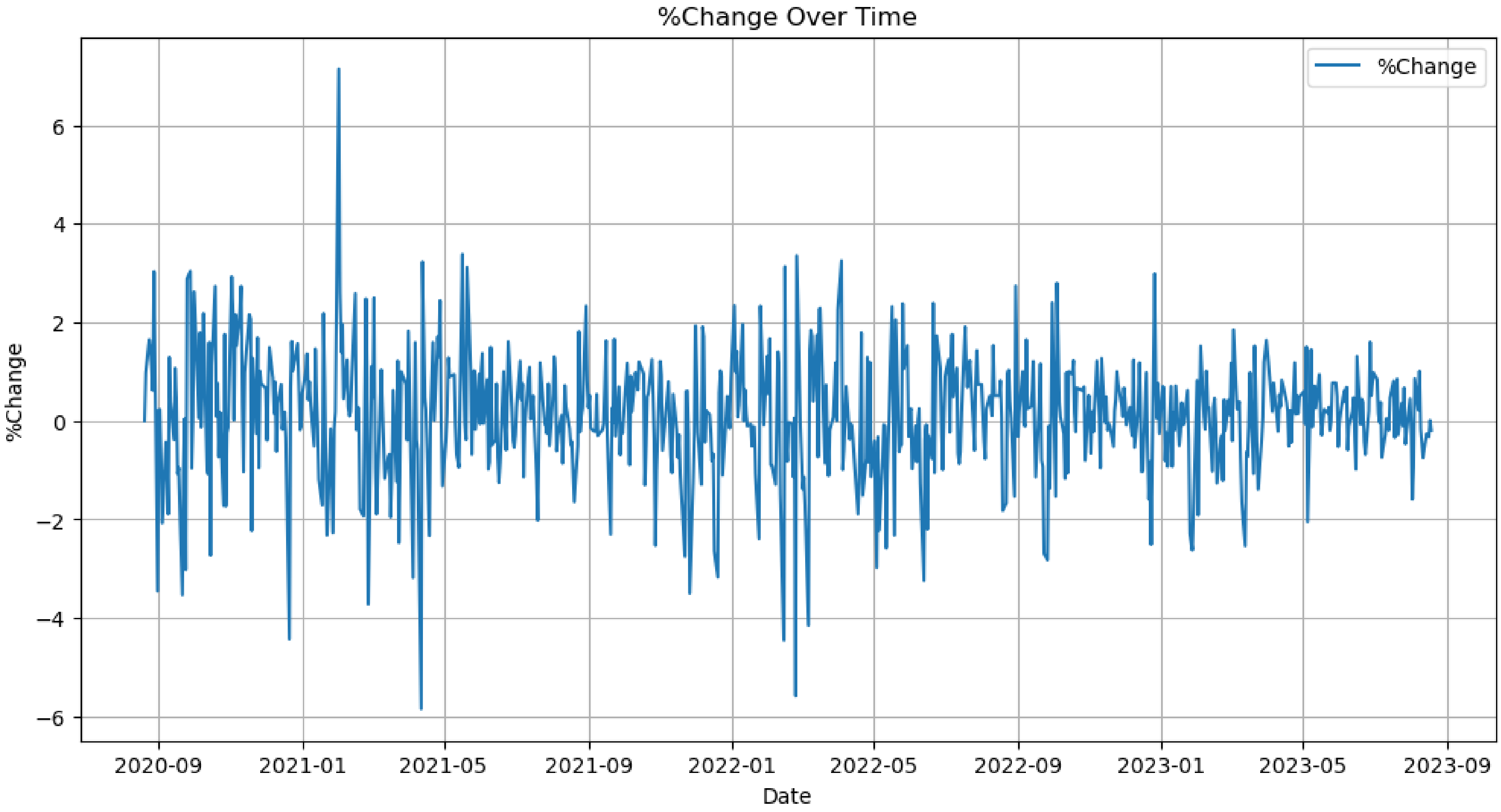
```
# Remove '%' and convert to float
Nippon_India_Banking_and_Financial_Services_Fund_Holdings['Weightage'] = Nippon_India_Banking_and_Financial_Services_Fund_Holdin
```

In [61]:

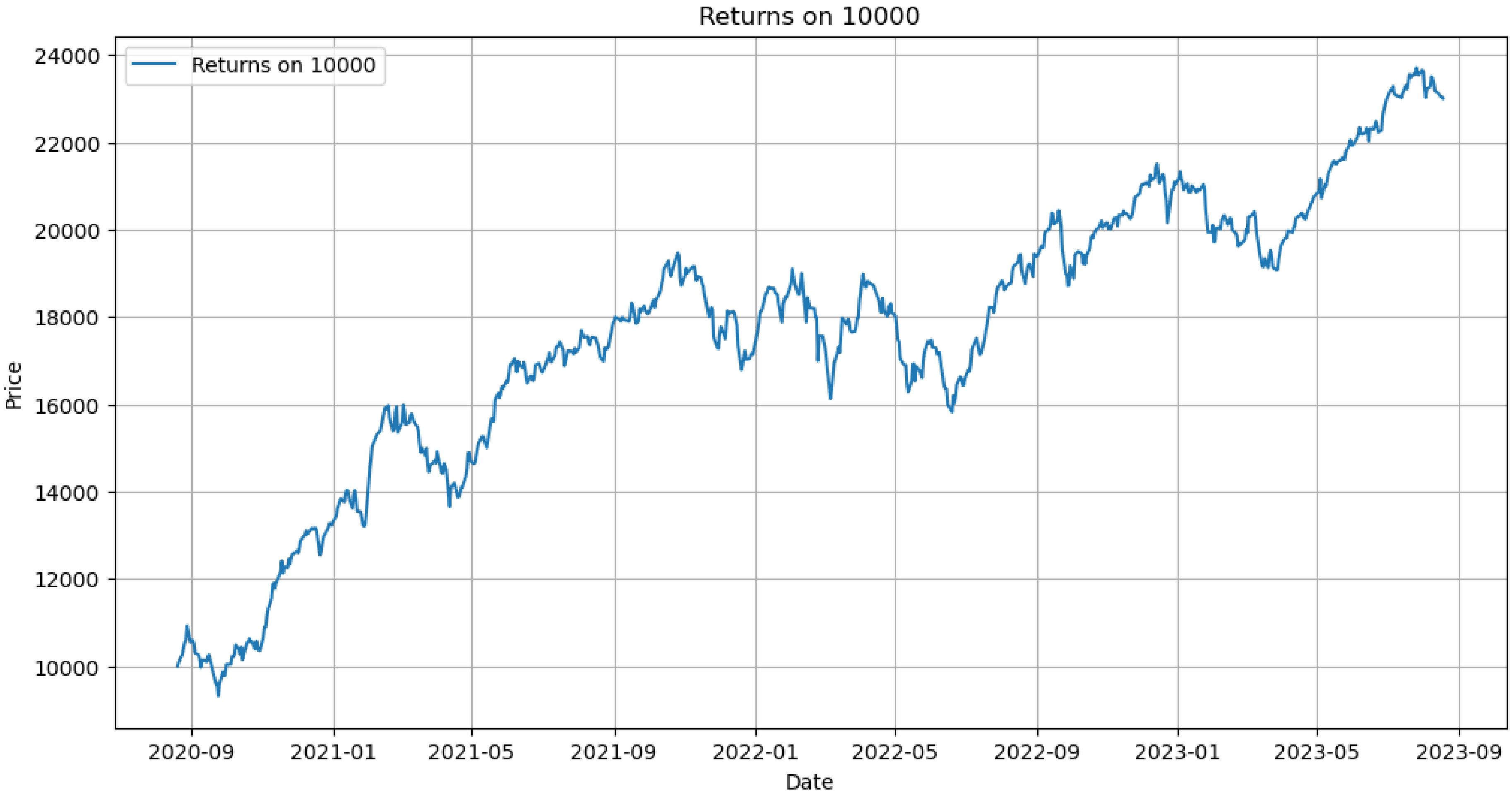
```
# Plot 1: Adj Close Over Time
plt.figure(figsize=(12, 6))
plt.plot(Nippon_India_Banking_and_Financial_Services_Fund['Date'], Nippon_India_Banking_and_Financial_Services_Fund['Adj Close']
plt.xlabel('Date')
plt.ylabel('Price')
plt.title('Adj Close')
plt.legend()
plt.grid()
plt.show()
```



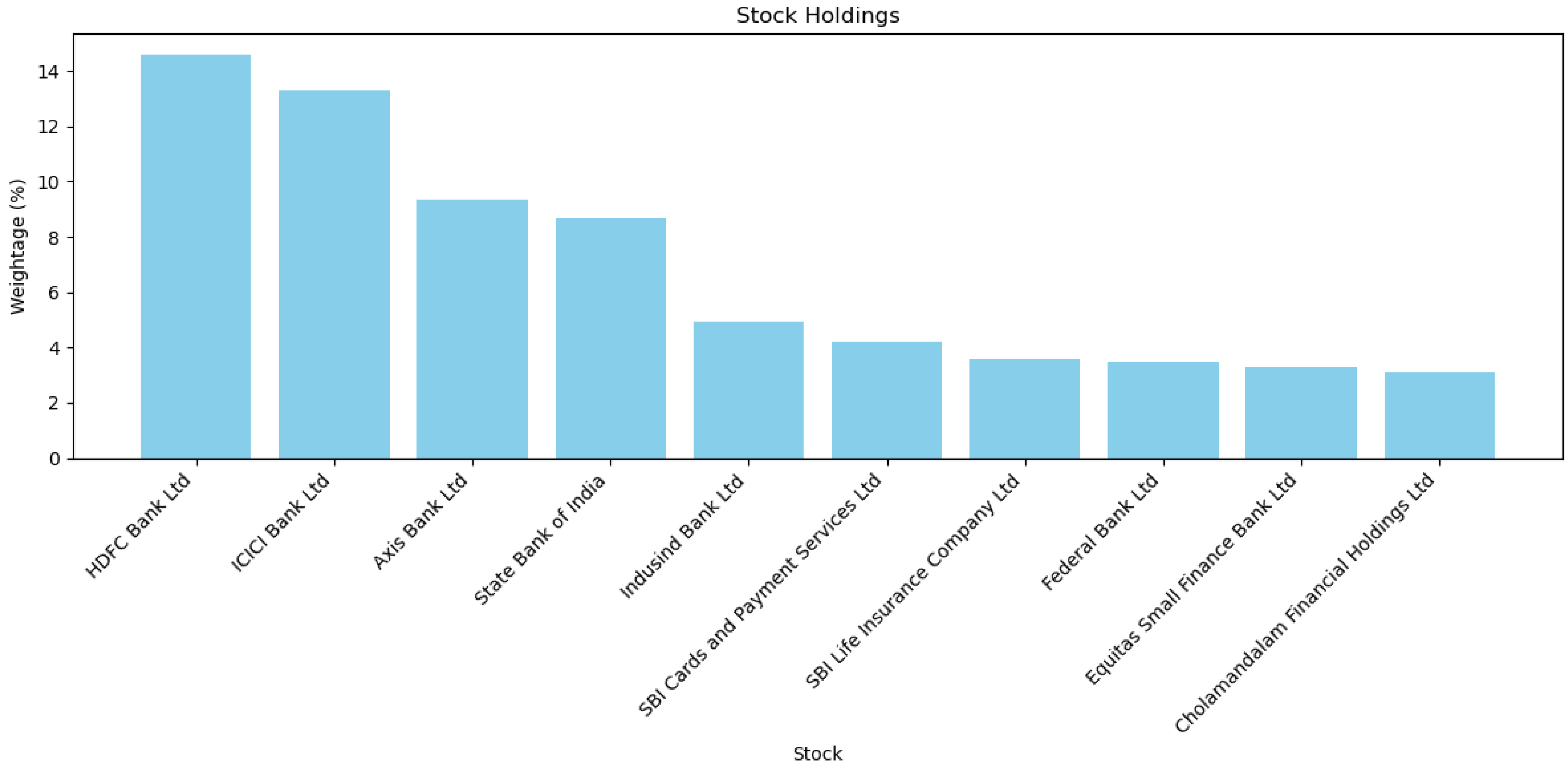
```
In [62]: # Plot 2: %Change Over Time
         plt.figure(figsize=(12, 6))
         plt.plot(Nippon_India_Banking_and_Financial_Services_Fund['Date'], Nippon_India_Banking_and_Financial_Services_Fund['%Change'], 
         plt.xlabel('Date')
         plt.ylabel('%Change')
         plt.title('%Change Over Time')
         plt.legend()
         plt.grid()
         plt.show()
```



In [63]:
```
# Plot 3: Returns on 10000 Over Time
plt.figure(figsize=(12, 6))
plt.plot(Nippon_India_Banking_and_Financial_Services_Fund['Date'], Nippon_India_Banking_and_Financial_Services_Fund['Returns on 
plt.xlabel('Date')
plt.ylabel('Price')
plt.title('Returns on 10000')
plt.legend()
plt.grid()
plt.show()
```



In [64]:
```
# Plot 4: Top 10 Stocks Holdings
plt.figure(figsize=(12, 6))
plt.bar(Nippon_India_Banking_and_Financial_Services_Fund_Holdings['Stock'], Nippon_India_Banking_and_Financial_Services_Fund_Hol
plt.xticks(rotation=45, ha='right')
plt.xlabel('Stock')
plt.ylabel('Weightage (%)')
plt.title('Stock Holdings')
plt.tight_layout()
```



In [1]:

```
# Importing necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

SBI Consumption Opp Fund

In [65]:

```
# Load Mutual Fund Data
SBI_Consumption_Opp_Fund = pd.read_csv("SBI Consumption Opp Fund.csv")
```

In [66]:

```
# Convert Date column to datetime format
SBI_Consumption_Opp_Fund['Date'] = pd.to_datetime(SBI_Consumption_Opp_Fund['Date'], format='%d-%m-%Y')
```

In [67]:

```
# Remove '%' and convert to float
SBI_Consumption_Opp_Fund['%Change'] = SBI_Consumption_Opp_Fund['%Change'].str.rstrip('%').astype(float)
```

In [68]:

```
# Display the sample of Mutual Fund
SBI_Consumption_Opp_Fund.head(10)
```

Out[68]:

```
         Date  Adj Close  %Change  Returns on 10000
0  2020-08-20    113.250     0.00          10000.00
1  2020-08-21    114.955     1.51          10150.54
2  2020-08-24    115.861     0.79          10230.52
3  2020-08-25    116.175     0.27          10258.26
4  2020-08-26    116.054    -0.10          10247.60
5  2020-08-27    116.582     0.45          10294.17
6  2020-08-28    116.966     0.33          10328.12
7  2020-08-31    113.151    -3.26           9991.21
8  2020-09-01    114.187     0.92          10082.73
9  2020-09-02    115.339     1.01          10184.42
```

In [69]:

```
# Calcualte minimum of %Change
min_change = np.min(SBI_Consumption_Opp_Fund['%Change'])
print("Minimum %Change:", min_change)
```

```
Minimum %Change: -4.61
```

In [70]:

```
# Calcualte maximum of %Change
max_change = np.max(SBI_Consumption_Opp_Fund['%Change'])
print("Maximum %Change:", max_change)
```

```
Maximum %Change: 2.81
```

In [71]:

```
# Calculate arithmetic mean of %Change
arithmetic_mean = np.mean(SBI_Consumption_Opp_Fund['%Change'])
print("Arithmetic Mean of %Change:", arithmetic_mean)
```

```
Arithmetic Mean of %Change: 0.11815363881401619
```

In [72]:

```
# Calculate geometric mean of %Change
geometric_mean = np.exp(np.mean(np.log(1 + SBI_Consumption_Opp_Fund['%Change'] / 100))) - 1
print("Geometric Mean of %Change:", geometric_mean)
```

```
Geometric Mean of %Change: 0.001143346671428569
```

In [73]:

```
# Calculate standard deviation of %Change
std_deviation_change = np.std(SBI_Consumption_Opp_Fund['%Change'])
print("Standard Deviation of %Change:", std_deviation_change)
```

```
Standard Deviation of %Change: 0.8720082241423731
```

In [74]:

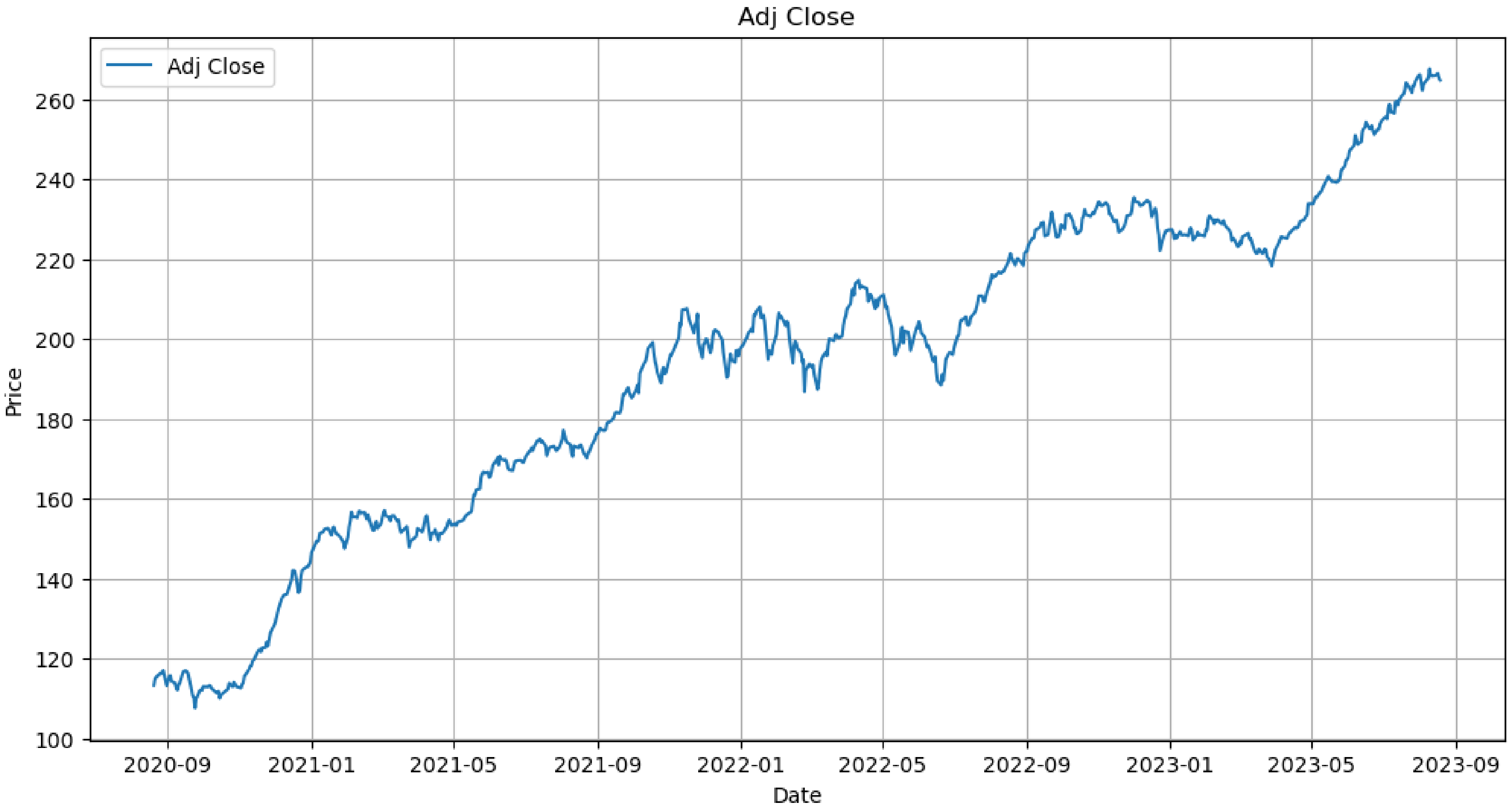
```
# Load Mutual Fund Holdings Data
SBI_Consumption_Opp_Fund_Holdings = pd.read_csv("SBI Consumption Opp Fund Top 10 Equity Holdings.csv")
```

In [75]:

```
# Remove '%' and convert to float
SBI_Consumption_Opp_Fund_Holdings['Weightage'] = SBI_Consumption_Opp_Fund_Holdings['Weightage'].str.rstrip('%').astype(float)
```

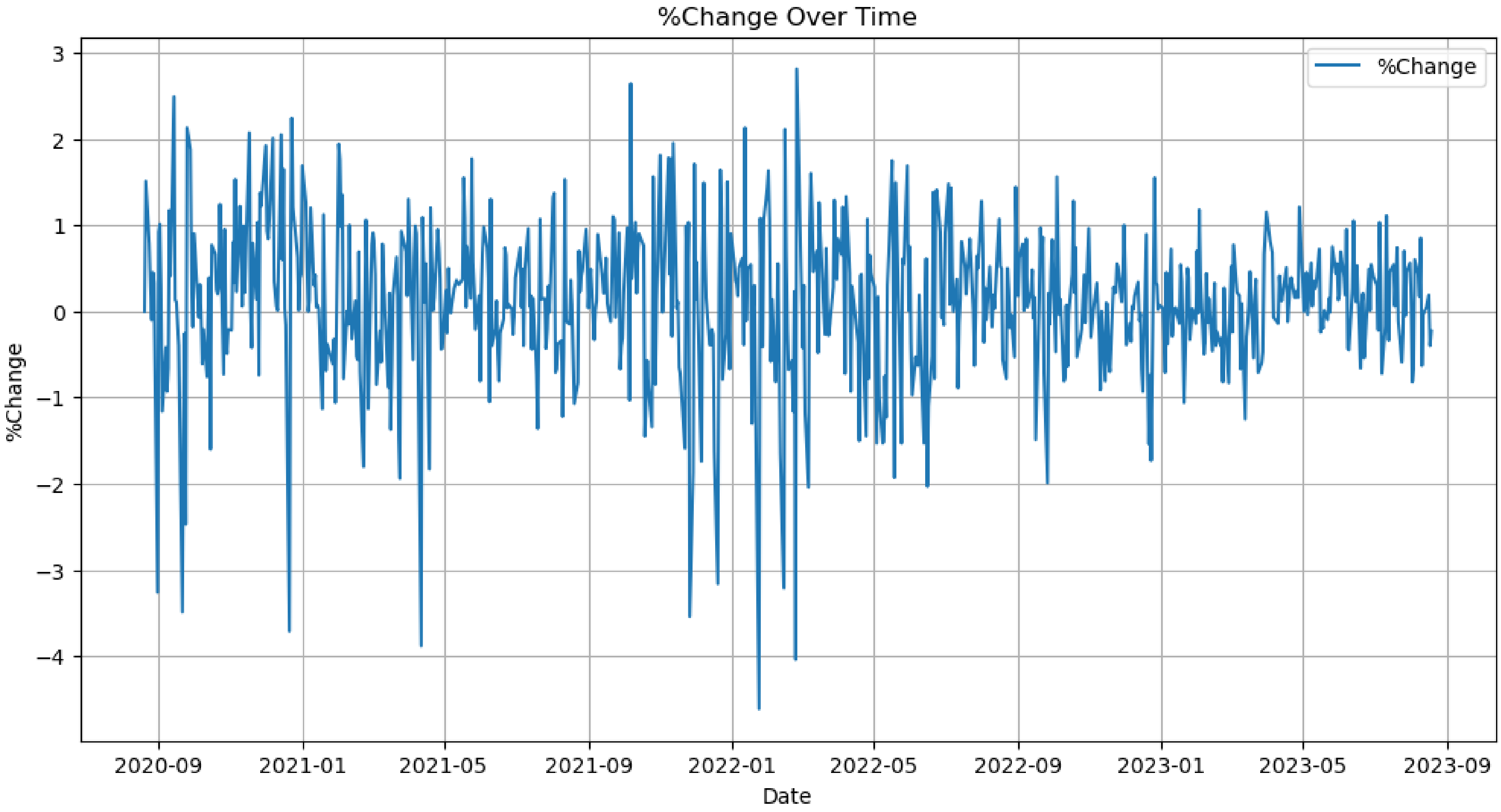
In [77]:

```
# Plot 1: Adj Close Over Time
plt.figure(figsize=(12, 6))
plt.plot(SBI_Consumption_Opp_Fund['Date'], SBI_Consumption_Opp_Fund['Adj Close'], label='Adj Close')
plt.xlabel('Date')
plt.ylabel('Price')
plt.title('Adj Close')
plt.legend()
plt.grid()
plt.show()
```



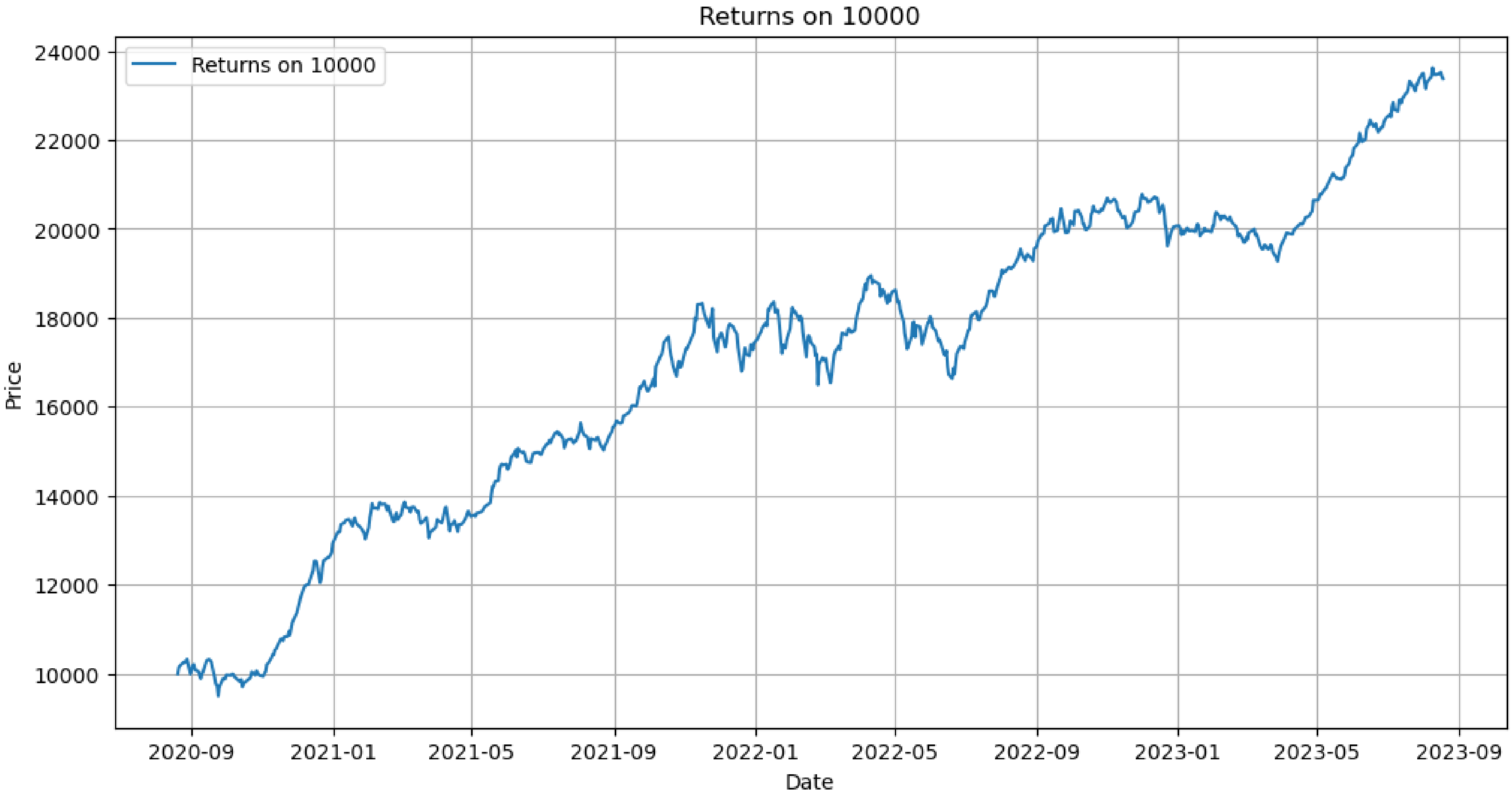
In [78]:

```
# Plot 2: %Change Over Time
plt.figure(figsize=(12, 6))
plt.plot(SBI_Consumption_Opp_Fund['Date'], SBI_Consumption_Opp_Fund['%Change'], label='%Change')
plt.xlabel('Date')
plt.ylabel('%Change')
plt.title('%Change Over Time')
plt.legend()
plt.grid()
plt.show()
```



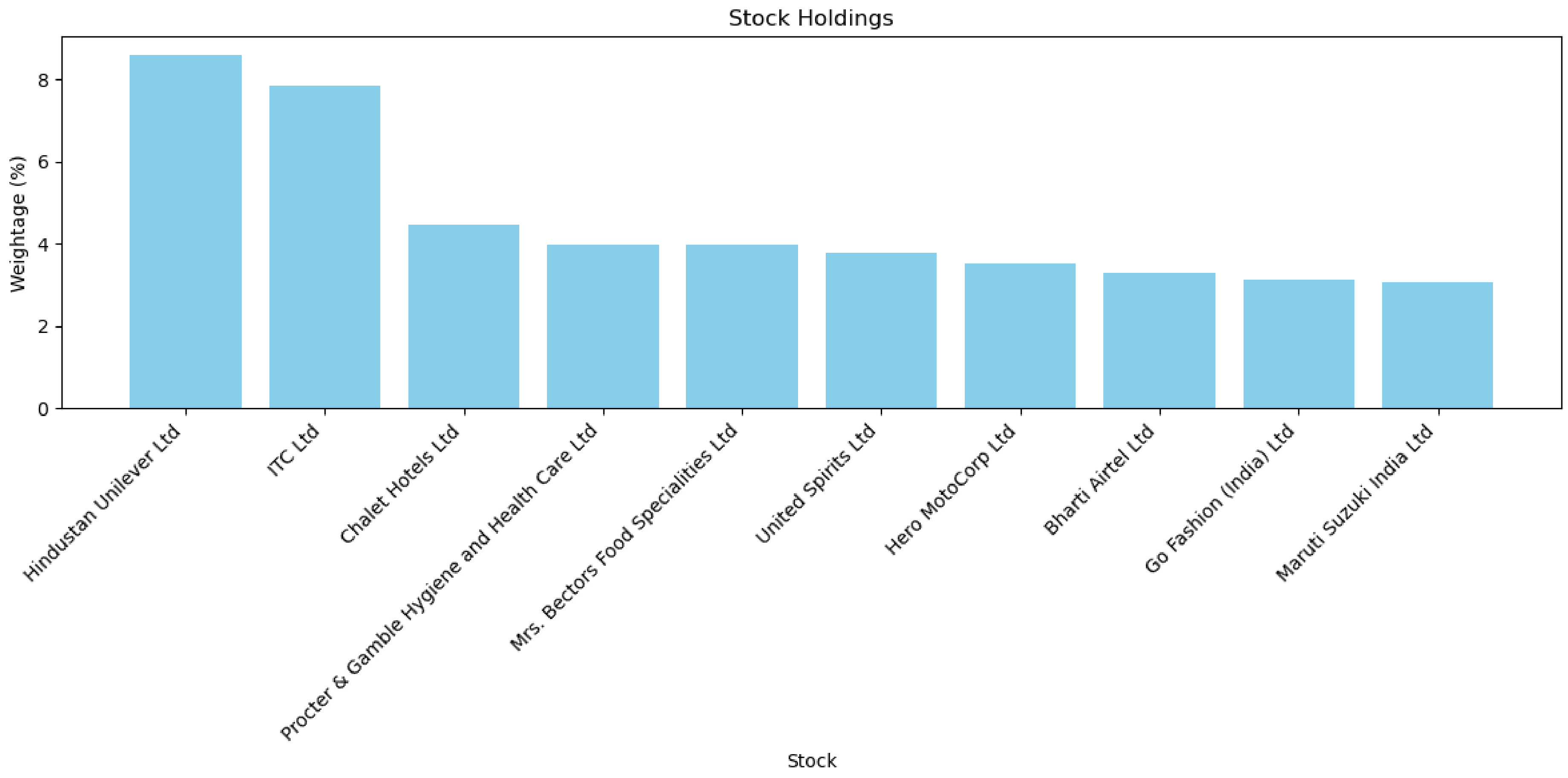
In [79]:

```
# Plot 3: Returns on 10000 Over Time
plt.figure(figsize=(12, 6))
plt.plot(SBI_Consumption_Opp_Fund['Date'], SBI_Consumption_Opp_Fund['Returns on 10000'], label='Returns on 10000')
plt.xlabel('Date')
plt.ylabel('Price')
plt.title('Returns on 10000')
plt.legend()
plt.grid()
plt.show()
```



In [80]:

```
# Plot 4: Top 10 Stocks Holdings
plt.figure(figsize=(12, 6))
plt.bar(SBI_Consumption_Opp_Fund_Holdings['Stock'], SBI_Consumption_Opp_Fund_Holdings['Weightage'], color='skyblue')
plt.xticks(rotation=45, ha='right')
plt.xlabel('Stock')
plt.ylabel('Weightage (%)')
plt.title('Stock Holdings')
plt.tight_layout()
```


Stock Holdings
Weightage (%)
Hindustan Unilever Ltd
ITC Ltd
Chalet Hotels Ltd
Procter & Gamble Hygiene and Health Care Ltd
Mrs. Bectors Food Specialities Ltd
United Spirits Ltd
Hero MotoCorp Ltd
Bharti Airtel Ltd
Go Fashion (India) Ltd
Maruti Suzuki India Ltd
Stock