1. Import Required Libraries

```
import pandas as pd
import numpy as np
```

2. Load and Inspect the Data

```
df = pd.read_csv(r"/content/yahoo_mutual_fund_gainers.csv")
```

View initial records

```
df.head()
```

	Symbol	Name	Price	Change	Change %	50 Day Avg	200 Day Avg	3 Month Return	YTD Return	52 Wk Change %	52 Wk Range
0	FTPAX	First Trust Private Assets Fund	11.51+0.89(+8.38%)	0.89	+8.38%	10.82	10.82	--	--	+11.53%	NaN
1	0P0001T9Z1	CSOP Ether ETF Unlisted Share C	6.23+0.28(+4.67%)	0.28	+4.67%	4.70	6.65	-43.65%	-48.37%	-31.09%	NaN
2	GRHIX	Goehring & Rozencwaig	14.13+0.52(+3.82%)	0.52	+3.82%	12.87	13.11	-2.39%	+0.80%	-5.61%	NaN

Next steps: Generate code with df View recommended plots New interactive sheet

Check shape and info

```
df.shape
```

```
(341, 10)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 341 entries, 0 to 340
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Symbol          341 non-null    object
 1   Name            341 non-null    object
 2   Price           341 non-null    float64
 3   Change          341 non-null    float64
 4   Change %        341 non-null    object
 5   50 Day Avg      341 non-null    object
 6   200 Day Avg     341 non-null    object
 7   3 Month Return  341 non-null    object
 8   YTD Return      341 non-null    object
 9   52 Wk Change %  341 non-null    object
dtypes: float64(2), object(8)
memory usage: 29.3+ KB
```

3. Data Cleaning and Preprocessing

3.1 Remove Duplicate Records

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

```
df.duplicated().sum()
```

```
np.int64(0)
```

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

	0
Symbol	0
Name	0
Price	0
Change	0
Change %	0
50 Day Avg	0
200 Day Avg	0
3 Month Return	0
YTD Return	0
52 Wk Change %	0

3.2 Drop Unnecessary Columns

```
# 1. Drop the '52 Wk Range' column
df.drop(columns=['52 Wk Range'], inplace=True)
```

3.3 Extract Only Numeric Price

```
# 2. Extract only the numeric price from 'Price' column (before '+' sign)
df['Price'] = df['Price'].str.extract(r'([\d.]+)')

# Optional: Convert 'Price' to float type
df['Price'] = df['Price'].astype(float)
```

3.4 Replace '--' with NaN in Specific Columns

3.5 Clean and Convert Return Columns to Float

```
# Re-check for '--' and replace with NaN again just to be sure
df[['3 Month Return', 'YTD Return']] = df[['3 Month Return', 'YTD Return']].replace('--', np.nan)

# Ensure values are clean for conversion (remove %, spaces)
df['3 Month Return'] = df['3 Month Return'].astype(str).str.replace('%', '', regex=False).str.strip()
df['YTD Return'] = df['YTD Return'].astype(str).str.replace('%', '', regex=False).str.strip()

# Convert to float (coerce will convert any invalid string to NaN)
df['3 Month Return'] = pd.to_numeric(df['3 Month Return'], errors='coerce')
df['YTD Return'] = pd.to_numeric(df['YTD Return'], errors='coerce')

# Count how many NaNs exist (i.e., originally '--')
count_3m_na = df['3 Month Return'].isna().sum()
count_ytd_na = df['YTD Return'].isna().sum()
print("NaNs in '3 Month Return':", count_3m_na)
print("NaNs in 'YTD Return':", count_ytd_na)
```

```
NaNs in '3 Month Return': 0
NaNs in 'YTD Return': 0
```

3.6 Fill NaN with Column Mean

```
df['3 Month Return'] = df['3 Month Return'].fillna(df['3 Month Return'].mean())
df['YTD Return'] = df['YTD Return'].fillna(df['YTD Return'].mean())
```

```
df['YTD Return'] = df['YTD Return'].fillna(df['YTD Return'].mean())
```

3.7 Add '%' Symbol Back After Imputation

```
#If you want to add the % symbol back:
df['3 Month Return'] = df['3 Month Return'].round(2).astype(str) + '%'
df['YTD Return'] = df['YTD Return'].round(2).astype(str) + '%'
```

4. Rename Columns for Clarity

```
df.rename(columns={
    'Change %': 'Daily Change %',
    '50 Day Avg': '50-Day Moving Avg',
    '200 Day Avg': '200-Day Moving Avg',
    '3 Month Return': '3-Month Return %',
    'YTD Return': 'Year-To-Date Return %',
    '52 Wk Change %': '52-Week Change %'
}, inplace=True)
```

5. Final Check

```
df.shape
df.head()
```

	Symbol	Name	Price	Change	Daily Change %	50-Day Moving Avg	200-Day Moving Avg	3-Month Return %	Year-To-Date Return %	52-Week Change %
0	FTPAX	First Trust Private Assets Fund	11.51	0.89	+8.38%	10.82	10.82	6.59%	10.1%	+11.53%
1	0P0001T9Z1	CSOP Ether ETF Unlisted Share C	6.23	0.28	+4.67%	4.70	6.65	-43.65%	-48.37%	-31.09%
2	GRHIX	Goehring & Rozencwajg	14.13	0.52	+3.82%	12.87	13.11	-2.39%	0.8%	-5.61%

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
# Suppose your cleaned DataFrame is named df
df.to_csv("cleaned_mutual_fund_data.csv", index=False)
```

Start coding or generate with AI.