

hw_2_part_2_question_2

February 10, 2025

```
[1]: import pandas as pd
import numpy as np
import statistics
from scipy.stats import skew, kurtosis, zscore
import matplotlib.pyplot as plt
import seaborn as sns
```

Matplotlib is building the font cache; this may take a moment.

Traded Company Data Utilizing Yahoo Finance, I opted to use Apple stock, as it is something I am interested in, for this dataset analysis. I accounted for holidays to assure I had the required 30 values over six weeks. I read in the Excel file to Python to complete the calculations. The column headers had white space that needed to be cleaned prior to use. The required column, Adj Close, defaulted to a float data type which allowed for calculations. Using the .diff() function, a difference column was created for reference. Using the .pct_change() function allowed for the percentage change to be documented for all weeks except the first (no previous value to compare). It was found that field needed to be multiplied by 100 in order to obtain the correct percentage value.

Code samples and definitions were located using ChatGPT. Cells will be noted where this was utilized.

```
[2]: # Homework 2, Question 2 (a) Read in data from apple_stock_history.xlsx
df = pd.read_excel('/Users/helenamabey/Downloads/apple_stock_history.xlsx')
# Used .head() to limit returns for ease of viewing
df.head()
```

```
[2]:
```

| | Date | Open | High | Low | Close | Adj Close | Volume |
|---|------------|--------|--------|--------|--------|-----------|----------|
| 0 | 2025-02-07 | 232.51 | 234.00 | 227.27 | 227.63 | 227.63 | 38983016 |
| 1 | 2025-02-06 | 231.29 | 233.80 | 230.43 | 233.22 | 233.22 | 29925300 |
| 2 | 2025-02-05 | 228.53 | 232.67 | 228.27 | 232.47 | 232.47 | 39620300 |
| 3 | 2025-02-04 | 227.25 | 233.13 | 226.65 | 232.80 | 232.80 | 45067300 |
| 4 | 2025-02-03 | 229.99 | 231.83 | 225.70 | 228.01 | 228.01 | 73063300 |

```
[3]: # Removed white space from column names
df.columns = df.columns.str.strip()
```

```
[5]: # Use .diff() to calculate the difference between the row prior. Index 0 will
      ↪ have a null value leaving 29 values.
      # Found sample code in ChatGPT to determine correct function to use.
      df['Difference'] = df['Adj Close'].diff()
      df
```

```
[5]:      Date      Open      High      Low      Close      Adj Close      Volume \
0  2025-02-07  232.51  234.00  227.27  227.63      227.63  38983016
1  2025-02-06  231.29  233.80  230.43  233.22      233.22  29925300
2  2025-02-05  228.53  232.67  228.27  232.47      232.47  39620300
3  2025-02-04  227.25  233.13  226.65  232.80      232.80  45067300
4  2025-02-03  229.99  231.83  225.70  228.01      228.01  73063300
5  2025-01-31  247.19  247.19  233.44  236.00      236.00  101075100
6  2025-01-30  238.67  240.79  237.21  237.59      237.59  55658300
7  2025-01-29  234.12  239.86  234.01  239.36      239.36  45486100
8  2025-01-28  230.85  240.19  230.81  238.26      238.26  75707600
9  2025-01-27  224.02  232.15  223.98  229.86      229.86  94863400
10 2025-01-24  224.78  225.63  221.41  222.78      222.78  54697900
11 2025-01-23  224.74  227.03  222.30  223.66      223.66  60234800
12 2025-01-22  219.79  224.12  219.79  223.83      223.83  64126500
13 2025-01-21  224.00  224.42  219.38  222.64      222.64  98070400
14 2025-01-17  232.12  232.29  228.48  229.98      229.98  68488300
15 2025-01-16  237.35  238.01  228.03  228.26      228.26  71759100
16 2025-01-15  234.64  238.96  234.43  237.87      237.87  39832000
17 2025-01-14  234.75  236.12  232.47  233.28      233.28  39435300
18 2025-01-13  233.53  234.67  229.72  234.40      234.40  49630700
19 2025-01-10  240.01  240.16  233.00  236.85      236.85  61710900
20 2025-01-08  241.92  243.71  240.05  242.70      242.70  37628900
21 2025-01-07  242.98  245.55  241.35  242.21      242.21  40856000
22 2025-01-06  244.31  247.33  243.20  245.00      245.00  45045600
23 2025-01-03  243.36  244.18  241.89  243.36      243.36  40244100
24 2025-01-02  248.93  249.10  241.82  243.85      243.85  55740700
25 2024-12-31  252.44  253.28  249.43  250.42      250.42  39480700
26 2024-12-30  252.23  253.50  250.75  252.20      252.20  35557500
27 2024-12-27  257.83  258.70  253.06  255.59      255.59  42355300
28 2024-12-26  258.19  260.10  257.63  259.02      259.02  27237100
29 2024-12-24  255.49  258.21  255.29  258.20      258.20  23234700
```

```
      Difference
0           NaN
1          5.59
2         -0.75
3          0.33
4         -4.79
5          7.99
6          1.59
7          1.77
```

```

8      -1.10
9      -8.40
10     -7.08
11      0.88
12      0.17
13     -1.19
14      7.34
15     -1.72
16      9.61
17     -4.59
18      1.12
19      2.45
20      5.85
21     -0.49
22      2.79
23     -1.64
24      0.49
25      6.57
26      1.78
27      3.39
28      3.43
29     -0.82

```

```

[6]: # Homework 2, Question 2 (b)
      # Use .pct_change() to calculate the percentage difference between the row prior.
      ↪ Index 0 will have a null value leaving 29 values.
      # Found sample code in ChatGPT to determine correct function to use and how to
      ↪ correct the decimal placing for a percentage.
df['Percentage Change'] = df['Adj Close'].pct_change() * 100
df

```

```

[6]:      Date    Open    High    Low    Close  Adj Close    Volume  \
0  2025-02-07  232.51  234.00  227.27  227.63    227.63  38983016
1  2025-02-06  231.29  233.80  230.43  233.22    233.22  29925300
2  2025-02-05  228.53  232.67  228.27  232.47    232.47  39620300
3  2025-02-04  227.25  233.13  226.65  232.80    232.80  45067300
4  2025-02-03  229.99  231.83  225.70  228.01    228.01  73063300
5  2025-01-31  247.19  247.19  233.44  236.00    236.00  101075100
6  2025-01-30  238.67  240.79  237.21  237.59    237.59  55658300
7  2025-01-29  234.12  239.86  234.01  239.36    239.36  45486100
8  2025-01-28  230.85  240.19  230.81  238.26    238.26  75707600
9  2025-01-27  224.02  232.15  223.98  229.86    229.86  94863400
10 2025-01-24  224.78  225.63  221.41  222.78    222.78  54697900
11 2025-01-23  224.74  227.03  222.30  223.66    223.66  60234800
12 2025-01-22  219.79  224.12  219.79  223.83    223.83  64126500
13 2025-01-21  224.00  224.42  219.38  222.64    222.64  98070400
14 2025-01-17  232.12  232.29  228.48  229.98    229.98  68488300

```

| | | | | | | | |
|----|------------|--------|--------|--------|--------|--------|----------|
| 15 | 2025-01-16 | 237.35 | 238.01 | 228.03 | 228.26 | 228.26 | 71759100 |
| 16 | 2025-01-15 | 234.64 | 238.96 | 234.43 | 237.87 | 237.87 | 39832000 |
| 17 | 2025-01-14 | 234.75 | 236.12 | 232.47 | 233.28 | 233.28 | 39435300 |
| 18 | 2025-01-13 | 233.53 | 234.67 | 229.72 | 234.40 | 234.40 | 49630700 |
| 19 | 2025-01-10 | 240.01 | 240.16 | 233.00 | 236.85 | 236.85 | 61710900 |
| 20 | 2025-01-08 | 241.92 | 243.71 | 240.05 | 242.70 | 242.70 | 37628900 |
| 21 | 2025-01-07 | 242.98 | 245.55 | 241.35 | 242.21 | 242.21 | 40856000 |
| 22 | 2025-01-06 | 244.31 | 247.33 | 243.20 | 245.00 | 245.00 | 45045600 |
| 23 | 2025-01-03 | 243.36 | 244.18 | 241.89 | 243.36 | 243.36 | 40244100 |
| 24 | 2025-01-02 | 248.93 | 249.10 | 241.82 | 243.85 | 243.85 | 55740700 |
| 25 | 2024-12-31 | 252.44 | 253.28 | 249.43 | 250.42 | 250.42 | 39480700 |
| 26 | 2024-12-30 | 252.23 | 253.50 | 250.75 | 252.20 | 252.20 | 35557500 |
| 27 | 2024-12-27 | 257.83 | 258.70 | 253.06 | 255.59 | 255.59 | 42355300 |
| 28 | 2024-12-26 | 258.19 | 260.10 | 257.63 | 259.02 | 259.02 | 27237100 |
| 29 | 2024-12-24 | 255.49 | 258.21 | 255.29 | 258.20 | 258.20 | 23234700 |

| | Difference | Percentage Change |
|----|------------|-------------------|
| 0 | NaN | NaN |
| 1 | 5.59 | 2.455740 |
| 2 | -0.75 | -0.321585 |
| 3 | 0.33 | 0.141954 |
| 4 | -4.79 | -2.057560 |
| 5 | 7.99 | 3.504232 |
| 6 | 1.59 | 0.673729 |
| 7 | 1.77 | 0.744981 |
| 8 | -1.10 | -0.459559 |
| 9 | -8.40 | -3.525560 |
| 10 | -7.08 | -3.080136 |
| 11 | 0.88 | 0.395009 |
| 12 | 0.17 | 0.076008 |
| 13 | -1.19 | -0.531653 |
| 14 | 7.34 | 3.296802 |
| 15 | -1.72 | -0.747891 |
| 16 | 9.61 | 4.210111 |
| 17 | -4.59 | -1.929625 |
| 18 | 1.12 | 0.480110 |
| 19 | 2.45 | 1.045222 |
| 20 | 5.85 | 2.469918 |
| 21 | -0.49 | -0.201895 |
| 22 | 2.79 | 1.151893 |
| 23 | -1.64 | -0.669388 |
| 24 | 0.49 | 0.201348 |
| 25 | 6.57 | 2.694279 |
| 26 | 1.78 | 0.710806 |
| 27 | 3.39 | 1.344171 |
| 28 | 3.43 | 1.341993 |
| 29 | -0.82 | -0.316578 |

```
[14]: # Verified that all rows except Difference and Percentage Change have 30 values.
      ↪ Difference and Percentage Change have 29 values.
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Date                  30 non-null    datetime64[ns]
1   Open                  30 non-null    float64
2   High                  30 non-null    float64
3   Low                   30 non-null    float64
4   Close                 30 non-null    float64
5   Adj Close             30 non-null    float64
6   Volume                30 non-null    int64
7   Difference             29 non-null    float64
8   Percentage Change     29 non-null    float64
dtypes: datetime64[ns](1), float64(7), int64(1)
memory usage: 2.2 KB
```

```
[17]: df.describe()
```

```
[17]:
```

| | | Date | Open | High | Low | Close \ |
|-------|---------------------|------------|------------|------------|------------|-----------|
| count | | 30 | 30.000000 | 30.000000 | 30.000000 | 30.000000 |
| mean | 2025-01-16 23:12:00 | 237.593667 | 240.022667 | 234.708333 | 237.376667 | |
| min | 2024-12-24 00:00:00 | 219.790000 | 224.120000 | 219.380000 | 222.640000 | |
| 25% | 2025-01-06 06:00:00 | 230.205000 | 232.785000 | 227.460000 | 229.890000 | |
| 50% | 2025-01-16 12:00:00 | 234.695000 | 239.410000 | 232.735000 | 236.425000 | |
| 75% | 2025-01-28 18:00:00 | 244.072500 | 246.780000 | 241.702500 | 243.195000 | |
| max | 2025-02-07 00:00:00 | 258.190000 | 260.100000 | 257.630000 | 259.020000 | |
| std | | NaN | 10.813714 | 10.130513 | 10.685454 | 10.314393 |

| | Adj Close | Volume | Difference | Percentage Change |
|-------|------------|--------------|------------|-------------------|
| count | 30.000000 | 3.000000e+01 | 29.000000 | 29.000000 |
| mean | 237.376667 | 5.316054e+07 | 1.054138 | 0.451616 |
| min | 222.640000 | 2.323470e+07 | -8.400000 | -3.525560 |
| 25% | 229.890000 | 3.951560e+07 | -1.100000 | -0.459559 |
| 50% | 236.425000 | 4.527670e+07 | 0.880000 | 0.395009 |
| 75% | 243.195000 | 6.352260e+07 | 3.390000 | 1.341993 |
| max | 259.020000 | 1.010751e+08 | 9.610000 | 4.210111 |
| std | 10.314393 | 2.036697e+07 | 4.264495 | 1.831889 |

Distribution Analysis

Homework 2, Question 2 1.), 2.), and 3.): When analyzing the distribution of percentage change and adjusted close price of Apple stock over the six-week period from 12/24/2024 to

02/07/2025, the skewness and kurtosis values for both metrics were reviewed first then validated with histograms and box plots.

The percentage change has a very slight left-skewed distribution with a skewness value of -0.0515. A symmetrical skewness would be at 0 so this value is nearly balanced. The kurtosis value of 0.1686 suggests a distribution that is somewhat peaked with moderate tails on both the left and right sides, indicating some outliers. When comparing this value to the normal kurtosis value of 3, this amount is just a bit higher so the distribution is also nearly normal. When reviewing the adjusted close price, the skewness value of 0.5653 shows a slight right-skewed distribution, with more lower values and a few higher outliers. The kurtosis value of -0.3554 shows a flatter distribution, with the values more evenly spread across the range rather than in a normal distribution with a bell-curve shape.

To confirm the initial findings, histograms and box plots were created for both metrics. The histogram and box plot for percentage change indicate a nearly normal distribution, with data balanced on both sides and few outliers. The use of 10 bins captures this distribution well. The box plot is nearly centered, and the mean and median are very similar, which both support the almost normal distribution of the values. For the adjusted close price, the histogram reveals a larger number of values toward the lower end, with a few outliers on the right. The bin count was increased to 12 which made it easier to define where the largest concentration of values were located. The box plot shows a similar result with the interquartile range pulled toward lower values. Even with the relatively wide range between the minimum and maximum values (222.64 and 259.02 consecutively), the mean and median are very similar.

Both the percentage change and adjusted close price seem approximately evenly distributed with the percentage change being closer to a normal distribution.

Based on the results of this analysis, Apple stock adjusted close price tends to have little fluctuation day over day, at least based on this small sample size. A few days saw gains and losses above +/- 3% but generally, the stock prices appears stable. In order to validate this further, a larger sample size may give insight to the stock's long-term performance and stability.

```
[15]: # Gather skewness and kurtosis for Percentage Change
print("Skewness:", df["Percentage Change"].skew())
print("Kurtosis:", df["Percentage Change"].kurt())
```

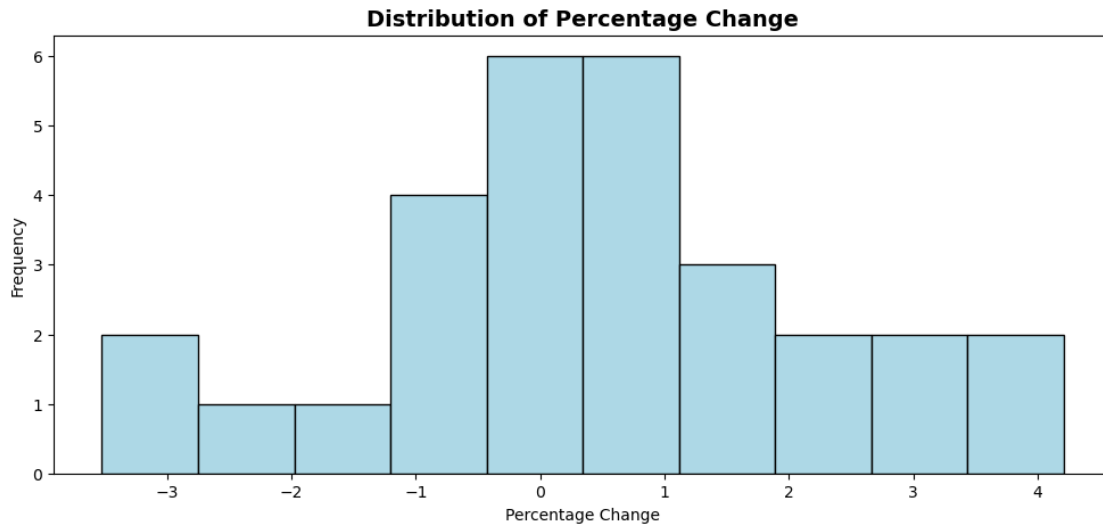
```
Skewness: -0.051498495794347926
Kurtosis: 0.16860820499134999
```

```
[16]: # Gather skewness and kurtosis for Adjusted Close
print("Skewness:", df["Adj Close"].skew())
print("Kurtosis:", df["Adj Close"].kurt())
```

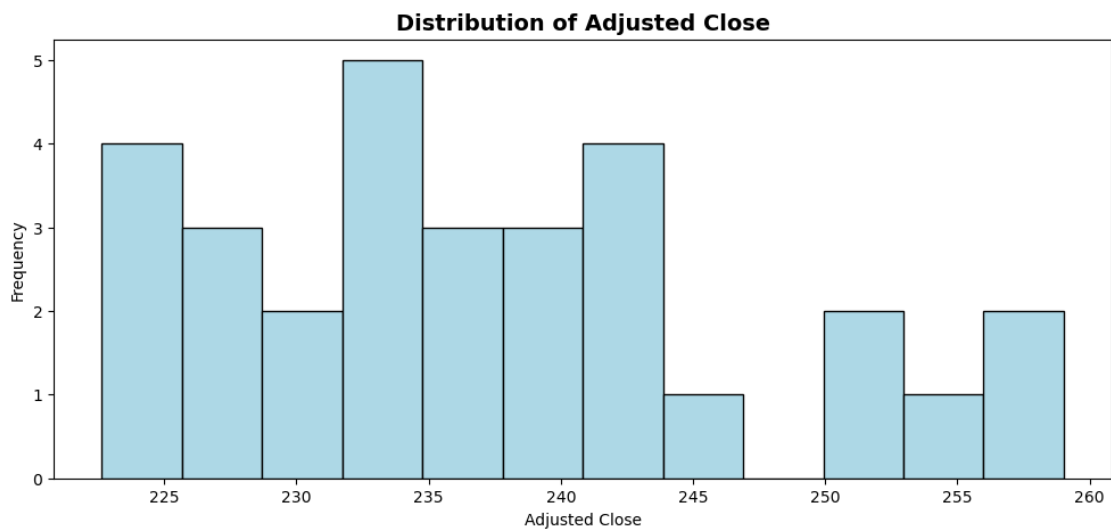
```
Skewness: 0.5653429215722883
Kurtosis: -0.3553698723722314
```

```
[23]: #Percentage Change Histogram
plt.figure(figsize=(12, 5))
df['Percentage Change'].plot(kind='hist', bins=10, color='lightblue',
                             edgecolor='black')
```

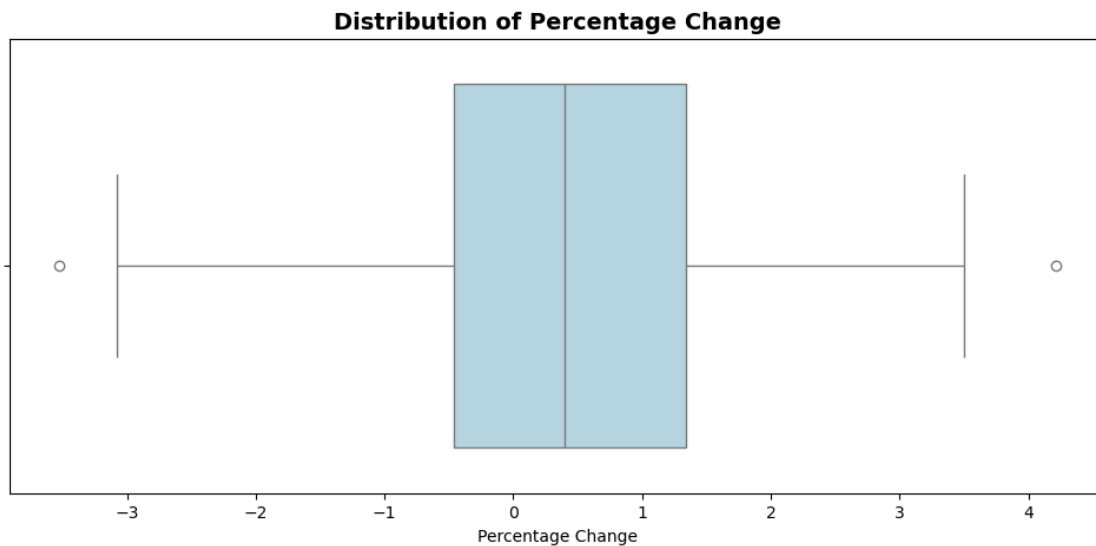
```
plt.title('Distribution of Percentage Change', fontsize=14, fontweight='bold')
plt.xlabel('Percentage Change')
plt.ylabel('Frequency')
plt.show()
```



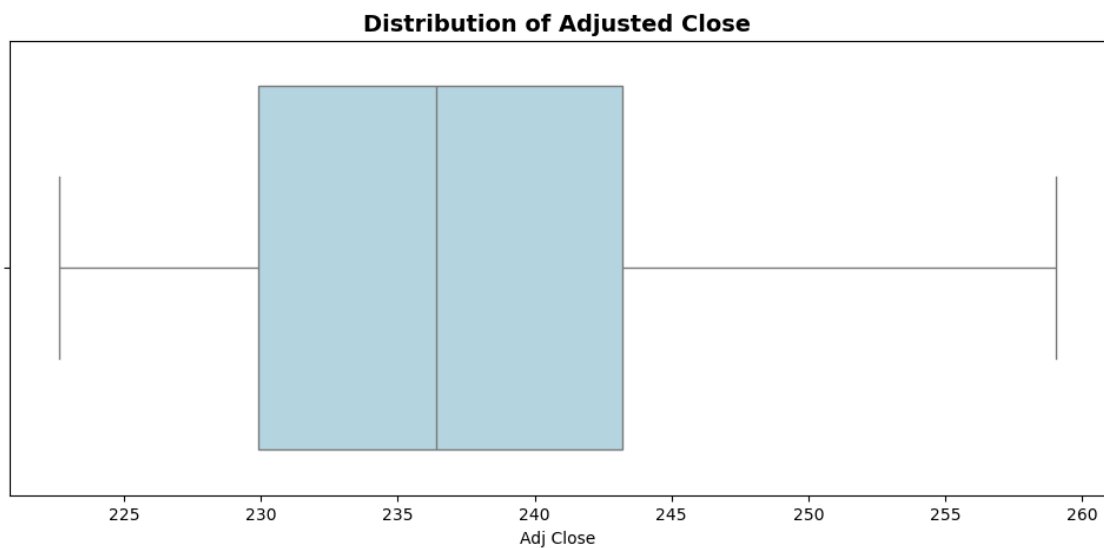
```
[20]: #Adjusted Close Histogram
plt.figure(figsize=(12, 5))
df['Adj Close'].plot(kind='hist', bins=12, color='lightblue', edgecolor='black')
plt.title('Distribution of Adjusted Close', fontsize=14, fontweight='bold')
plt.xlabel('Adjusted Close')
plt.ylabel('Frequency')
plt.show()
```



```
[18]: #Percentage Change Box Plot
plt.figure(figsize=(12, 5))
sns.boxplot(x=df["Percentage Change"], color="lightblue")
plt.title("Distribution of Percentage Change", fontsize=14, fontweight='bold')
plt.show()
```



```
[19]: #Adjusted Close Box Plot
plt.figure(figsize=(12, 5))
sns.boxplot(x=df["Adj Close"], color="lightblue")
plt.title("Distribution of Adjusted Close", fontsize=14, fontweight='bold')
plt.show()
```



AI Statement During this exercise, I utilized multiple sources online to assist in learning and understanding definitions and appropriate Python coding. In my notebook, I cited references to ChatGPT as they were used and the purpose for which they were used. I am developing my Python skills, so I relied on the sample code provided by ChatGPT to give guidance on how to apply it to this dataset. For the visualizations, I utilized my prior assignment code and adjusted it for this dataset. Sample code for that assignment was located through ChatGPT for formatting and other visualization features.