

Business Scenario

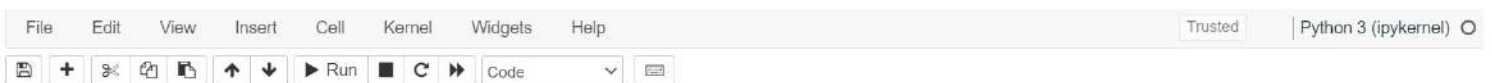
Problem statement:

The Nasdaq-100 is a stock market index comprised of 102 equity securities issued by 101 of the Nasdaq's largest nonfinancial companies. It includes sectors such as manufacturing, technology, retail, telecommunication, biotechnology, health care, transportation, media, and service providers. The cluster trading strategy is used to build a diverse portfolio of investments. This method enables the identification of different company segments. One advantage of this analysis is that it can help to protect an investor's portfolio from risks.

Objective:

You must now create such segments so that customers can identify segments to invest in and segments to avoid. Use cluster analysis techniques to accomplish this task. You will also need to perform time-series forecasting for stock prices.

```
In [1]:  #import required libraries  
import pandas as pd  
import numpy as np
```



```
In [1]: #import required libraries  
import pandas as pd  
import numpy as np  
  
import matplotlib  
import matplotlib.pyplot as plt  
import seaborn as sns  
  
import datetime  
import os  
  
import warnings;  
warnings.simplefilter('ignore')  
pd.set_option('display.max_columns', None)
```

```
In [2]: #import required dataset  
market_cap = pd.read_excel("Nasdaq 100 Market cap.xlsx")  
metric_ratios = pd.read_excel("nasdaq100_metrics_ratios.xlsx")
```

Append all files (imported in the previous step) that contain no more than 10 years of data.

```
In [3]: stocks = list(market_cap["Symbol"].unique())
```

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel) ○

📁 + 🔍 📄 ⬆️ ⬆️ ▶ Run ■ ↺ ▶ Code ▾ 📧

```
In [3]: stocks = list(market_cap["Symbol"].unique())
```

```
In [4]: csv_file = []
        for i in stocks:
            csv_file.append(str(i)+str(".csv"))
```

```
In [5]: csv_file
```

```
Out[5]: ['AAPL.csv',
         'ABNB.csv',
         'ADBE.csv',
         'ADI.csv',
         'ADP.csv',
         'ADSK.csv',
         'AEP.csv',
         'ALGN.csv',
         'AMAT.csv',
         'AMD.csv',
         'AMGN.csv',
         'AMZN.csv',
         'ANSS.csv',
         'ASML.csv',
         'ATVI.csv',
         'AVGO.csv',
         'AZN.csv',
         'BTHI.csv']
```

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel) ○

📄 + 🔍 📌 📄 ⬆️ ⬆️ ▶️ Run ■ ↺️ ▶️ Code ▾ 🗨️

In [6]: `final_data = pd.read_csv("AAPL.csv")`

In [7]: `final_data["Symbol"] = "AAPL"`

In [8]: `stocks.remove("AAPL")
csv_file.remove("AAPL.csv")`

In [9]: `for stock, i in zip(stocks, csv_file):
 data = pd.read_csv(i)
 data["Symbol"] = stock
 final_data = final_data.append(data, ignore_index=True)`


In [10]: `final_data.shape`











Out[10]: (474247, 8)

In [11]: `final_data.tail()`

Out[11]:

	Date	High	Low	Open	Close	Volume	Adj Close	Symbol
474242	2022-10-24	151.460007	144.300003	151.360001	148.669998	1375200.0	148.669998	ZS
474243	2022-10-25	157.580002	150.619995	150.619995	157.089996	1928300.0	157.089996	ZS
474244	2022-10-26	158.679001	149.210007	151.449997	151.029999	1865900.0	151.029999	ZS

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel) 

       Run    Code 

```
In [12]: final_data['Date'] = pd.to_datetime(final_data['Date'])

In [13]: current_date = datetime.datetime.now()

In [14]: max_date_range = current_date - datetime.timedelta(days=365*10)

In [15]: final_data = final_data.loc[final_data['Date'] >= max_date_range]

In [16]: final_data.shape

Out[16]: (230802, 8)
```

Collating Nasdaq 100 market cap.xlsx and nasdaq100_metrics_ratios.xlsx

```
In [17]: market_cap.columns

Out[17]: Index(['Symbol', 'Name', 'Market Cap', 'Last Sale', 'Net Change',
               'Percentage Change'],
              dtype='object')

In [18]: metric_ratios.columns
```

FileEditViewInsertCellKernelWidgetsHelp

TrustedPython 3 (ipykernel)

Code

Collating Nasdaq 100 market cap.xlsx and nasdaq100_metrics_ratios.xlsx

In [17]: market_cap.columns

```
Out[17]: Index(['Symbol', 'Name', 'Market Cap', 'Last Sale', 'Net Change',
               'Percentage Change'],
              dtype='object')
```

In [18]: metric_ratios.columns

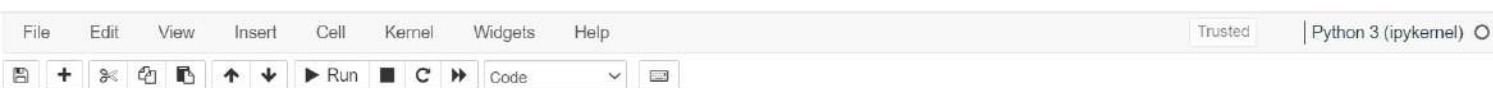
```
Out[18]: Index(['symbol', 'company', 'sector', 'subsector', 'asset_turnover_2017',
               'asset_turnover_2018', 'asset_turnover_2019', 'asset_turnover_2020',
               'asset_turnover_2021', 'asset_turnover_2022',
               ...,
               'yoy_eps_growth_2021', 'yoy_eps_growth_2022', 'yoy_eps_growth_latest',
               'yoy_revenue_growth_2017', 'yoy_revenue_growth_2018',
               'yoy_revenue_growth_2019', 'yoy_revenue_growth_2020',
               'yoy_revenue_growth_2021', 'yoy_revenue_growth_2022',
               'yoy_revenue_growth_latest'],
              dtype='object', length=283)
```

In [19]: metric_ratios.rename(columns={'symbol': 'Symbol'}, inplace=True)

In [20]: market_metric_combined = pd.merge(market_cap, metric_ratios, how='inner')

```
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)
In [20]: market_metric_combined = pd.merge(market_cap, metric_ratios, how='inner')
In [21]: market_metric_combined
Out[21]:
```

	Symbol	Name	Market Cap	Last Sale	Net Change	Percentage Change	company	sector	subsector	asset_turnover_2017	asset_turn
0	AAPL	Apple Inc.	2625740143000	\$151.45	\$2.00	0.0134	Apple Inc.	Information Technology	Technology Hardware, Storage & Peripherals	0.66	
1	ABNB	Airbnb, Inc.	69569944167	\$116.65	\$0.26	0.0022	Airbnb	Consumer Discretionary	Internet & Direct Marketing Retail	NaN	
2	ADBE	Adobe Inc.	149144569000	\$320.81	\$4.59	0.0145	Adobe Inc.	Information Technology	Application Software	0.54	
3	ADI	Analog Devices, Inc.	75484763090	\$146.76	\$2.23	0.0154	Analog Devices	Information Technology	Semiconductors	0.36	
4	ADP	Automatic Data Processing, Inc.	98332762096	\$236.78	\$0.13	-0.0005	ADP	Information Technology	Data Processing & Outsourced Services	NaN	
...
...	...	Walgreens Boots	Walgreens	Consumer



Identifying the variables whose variance is less than 0.005 (as these do not contribute to model building), and eliminating those variables.

```
In [22]: var= market_metric_combined.var()
```


```
In [23]: X=[]
         for i in range(len(var)):
             if var[i] < 0.005:
                 X.append(var.index[i])
```













```
In [24]: market_metric_combined.drop(X, axis=1, inplace = True)
```

```
In [25]: var_final_data = final_data.var()
```

```
In [26]: var_final_data
```

```
Out[26]: High      4.450744e+04
         Low       4.221611e+04
         Open      4.339026e+04
         Close     4.335064e+04
         Volume    8.637191e+14
         Adj Close  4.354625e+04
         dtype: float64
```


File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel) 

          Code  

Missing value treatment

```
In [27]: missing_value = market_metric_combined.isna().sum()
```

```
In [28]: missing_percentage_list = []
for i in range(len(missing_value)):
    percentage_of_missing = (missing_value[i]/102) *100
    if percentage_of_missing > 30:
        missing_percentage_list.append(missing_value.index[i])
```

```
In [29]: len(missing_percentage_list)
```

```
Out[29]: 80
```

```
In [30]: market_metric_combined.drop(missing_percentage_list, axis=1, inplace=True)
```

```
In [31]: columns_for_missing_value_treatment = []
for i in range(len(missing_value)):
    percentage_of_missing = (missing_value[i]/102) *100
    if percentage_of_missing > 0 and percentage_of_missing < 30:
        columns_for_missing_value_treatment.append(missing_value.index[i])
```

```
In [32]: columns_for_missing_value_treatment
```

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel) ○

📄 + 🔍 📌 ⬆️ ⬆️ ▶️ Run ■ ↺ ▶️ Code ▾ 🗨

In [32]: ▶️ columns_for_missing_value_treatment

```
'buyback_yield_2021',
'capex_to_revenue_2017',
'capex_to_revenue_2018',
'capex_to_revenue_2019',
'capex_to_revenue_2020',
'capex_to_revenue_2021',
'cash_ratio_2017',
'cash_ratio_2018',
'cash_ratio_2019',
'cash_to_debt_2018',
'cash_to_debt_2019',
'cash_to_debt_2020',
'cash_to_debt_2021',
'cash_to_debt_latest',
'cogs_to_revenue_2017',
'cogs_to_revenue_2018',
'cogs_to_revenue_2019',
'cogs_to_revenue_2020',
'cogs_to_revenue_2021',
'mscore_2018'
```

In [33]: ▶️ for i in columns_for_missing_value_treatment:
 median = market_metric_combined[i].median()
 market_metric_combined[i].fillna(median, inplace = True)

FileEditViewInsertCellKernelWidgetsHelp

TrustedPython 3 (ipykernel)

+

⌕

📄

📁

⬆️

⬇️

▶️ Run

⏏️

↺️

▶️

Code

In [34]:

market_metric_combined.head()

Out[34]:











	Symbol	Name	Market Cap	Last Sale	Net Change	company	sector	subsector	asset_turnover_2017	asset_turnover_2018	asset_turnover_2019
0	AAPL	Apple Inc.	2625740143000	\$151.45	\$2.00	Apple Inc.	Information Technology	Technology Hardware, Storage & Peripherals	0.66	0.72	0.78
1	ABNB	Airbnb, Inc.	69569944167	\$116.65	\$0.26	Airbnb	Consumer Discretionary	Internet & Direct Marketing Retail	0.60	0.55	0.51
2	ADBE	Adobe Inc.	149144569000	\$320.81	\$4.59	Adobe Inc.	Information Technology	Application Software	0.54	0.54	0.54
3	ADI	Analog Devices, Inc.	75484763090	\$146.76	\$2.23	Analog Devices	Information Technology	Semiconductors	0.36	0.30	0.28
4	ADP	Automatic Data Processing, Inc.	98332762096	\$236.78	\$0.13	ADP	Information Technology	Data Processing & Outsourced Services	0.60	0.34	0.34

In [35]:


start_date = '2020-03-31'

FileEditViewInsertCellKernelWidgetsHelp

TrustedPython 3 (ipykernel)



Code



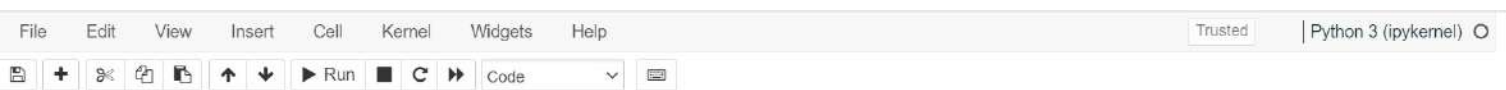
```
In [35]: start_date = '2020-03-31'
end_date = '2020-12-31'
```

```
In [36]: filter=(final_data['Date'] > start_date) & (final_data['Date'] <= end_date)
```

```
In [37]: final_data.loc[filter][0:50]
```

Out[37]:

	Date	High	Low	Open	Close	Volume	Adj Close	Symbol
5034	2020-06-01	74.989998	73.187500	73.447502	74.949997	118387200.0	73.426811	AAPL
5035	2020-07-01	75.224998	74.370003	74.959999	74.597504	108872000.0	73.081505	AAPL
5036	2020-08-01	76.110001	74.290001	74.290001	75.797501	132079200.0	74.257118	AAPL
5037	2020-09-01	77.607498	76.550003	76.809998	77.407501	170108400.0	75.834389	AAPL
5038	2020-10-01	78.167503	77.062500	77.650002	77.582497	140644800.0	76.005829	AAPL
5054	2020-04-02	79.910004	78.407501	78.827499	79.712502	136616400.0	78.092537	AAPL
5055	2020-05-02	81.190002	79.737503	80.879997	80.362503	118826800.0	78.729340	AAPL
5056	2020-06-02	81.305000	80.065002	80.642502	81.302498	105425600.0	79.650223	AAPL
5057	2020-07-02	80.849998	79.500000	80.592499	80.007500	117684000.0	78.567581	AAPL
5058	2020-10-02	80.387497	78.462502	78.544998	80.387497	109348800.0	78.940727	AAPL
5059	2020-11-02	80.974998	79.677498	80.900002	79.902496	94323200.0	78.464462	AAPL



MACHINE LEARNING MODEL

```
In [38]: #importing all packages required
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
from sklearn.preprocessing import LabelEncoder
```

```
In [39]: lab_enc = LabelEncoder()
```

```
In [40]: market_metric = market_metric_combined
```

```
In [41]: categorical_columns = ['Symbol', 'Name', 'company', 'sector', 'subsector']
for column in categorical_columns:
    market_metric_combined[column] = lab_enc.fit_transform(market_metric_combined[column])
```

```
In [42]: market_metric_combined["Last Sale"] = market_metric_combined["Last Sale"].str.replace('$', '')
market_metric_combined["Net Change"] = market_metric_combined["Net Change"].str.replace('$', '')
market_metric_combined["Last Sale"] = market_metric_combined["Last Sale"].str.replace(',', '')
market_metric_combined["Net Change"] = market_metric_combined["Net Change"].str.replace(',', '')
```


```
In [43]: market_metric_combined["Last Sale"] = pd.to_numeric(market_metric_combined["Last Sale"])
market_metric_combined["Net Change"] = pd.to_numeric(market_metric_combined["Net Change"])
```

In [44]:  market_metric_combined.head()

Out[44]:

	Symbol	Name	Market Cap	Last Sale	Net Change	company	sector	subsector	asset_turnover_2017	asset_turnover_2018	asset_turnover_2019	asset_turn
0	0	12	2625740143000	151.45	2.00	14	5	38	0.66	0.72	0.74	
1	1	5	69569944167	116.65	0.26	5	1	23	0.60	0.55	0.64	
2	2	3	149144569000	320.81	4.59	4	5	2	0.54	0.54	0.57	
3	3	11	75484763090	146.76	2.23	12	5	34	0.36	0.30	0.29	
4	4	17	98332762096	236.78	0.13	0	5	10	0.60	0.34	0.34	

In [45]:  *#PCA is sensitive to the scale of the data, so it's also a good practice to scale the data before performing PCA. This can be*
from sklearn.preprocessing import StandardScaler

In [46]:  std_slr = StandardScaler()

In [47]:  std_data = std_slr.fit_transform(market_metric_combined)

In [48]:  pca = PCA(n_components = 21)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel) ○

Run Code

```
In [49]: pca_data = pca.fit_transform(std_data)
```

```
In [50]: kmeans = KMeans(n_clusters=4)
```

```
In [51]: kmeans.fit(pca_data)
```

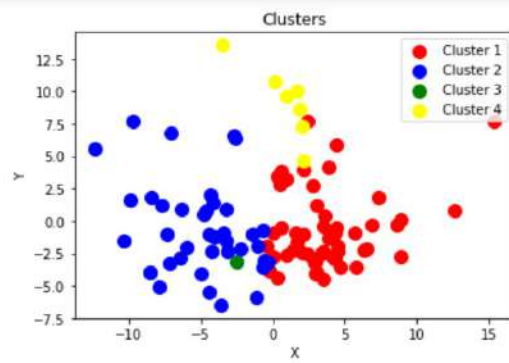
```
Out[51]: KMeans(n_clusters=4)
```


```
In [52]: predictions = kmeans.predict(pca_data)
```

```
In [53]: #market_metric_combined.drop(["cluster"],axis=1,inplace=True)
```

```
In [54]: # Visualize the clusters
plt.scatter(pca_data[predictions == 0, 0], pca_data[predictions == 0, 1], s = 100, c = 'red', label = 'Cluster 1')
plt.scatter(pca_data[predictions == 1, 0], pca_data[predictions == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')
plt.scatter(pca_data[predictions == 2, 0], pca_data[predictions == 2, 1], s = 100, c = 'green', label = 'Cluster 3')
plt.scatter(pca_data[predictions == 3, 0], pca_data[predictions == 3, 1], s = 100, c = 'yellow', label = 'Cluster 4')

#plt.scatter(kmeans.cluster_centers[:, 0], kmeans.cluster_centers[:, 1], s = 300, c = 'yellow', label = 'Centroids')
plt.title('Clusters')
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()
```



In []: 

Apple stock price analysis

In [55]:  `Apple_stock = pd.read_csv('AAPL.csv')`

Apple stock price analysis

```
In [55]: Apple_stock = pd.read_csv('AAPL.csv')

In [56]: Apple_stock["Date"] = pd.to_datetime(Apple_stock['Date'])

In [57]: Apple_stock.index = pd.to_datetime(Apple_stock['Date'])

In [58]: Apple_stock.drop(['Date', 'High', 'Low', 'Volume', 'Adj Close', 'Open'], axis=1, inplace=True)

In [59]: from statsmodels.tsa.seasonal import seasonal_decompose

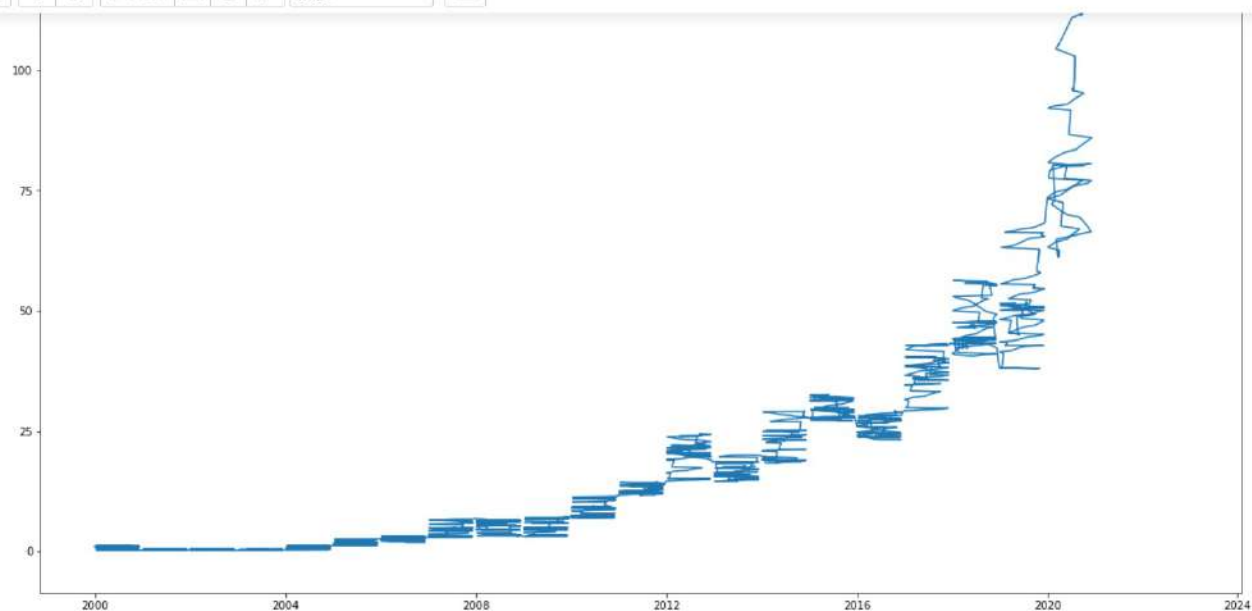
In [60]: result = seasonal_decompose(Apple_stock, model='multiplicative', period=12)


In [61]: plt.figure(figsize=(20,16))
plt.plot(result.trend)
```













Out[61]: [<matplotlib.lines.Line2D at 0x20e77d94e20>]

175



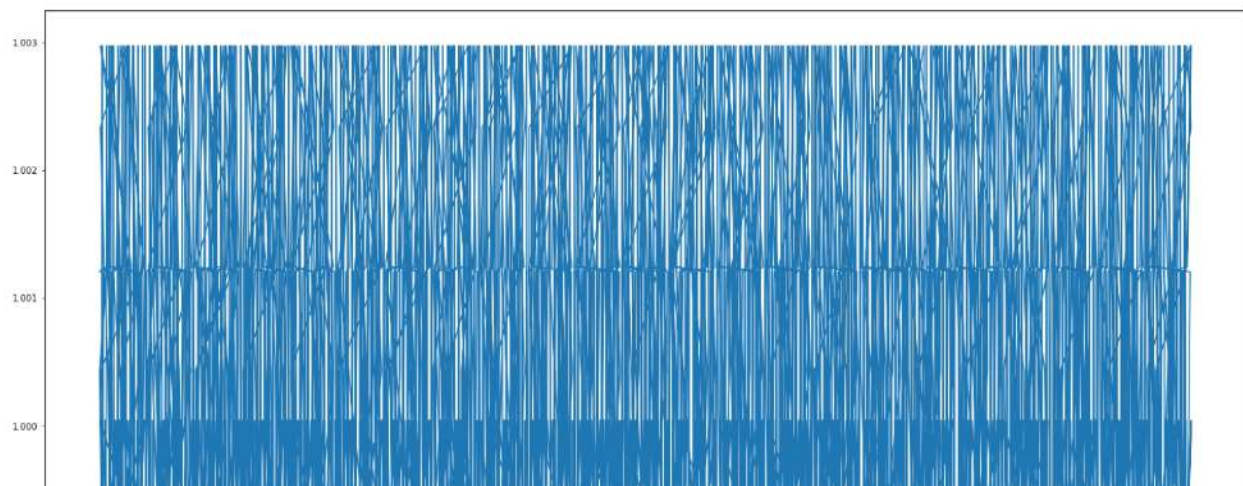



File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel) 













       Run    Code  

```
In [62]: plt.figure(figsize=(24,16))  
plt.plot(result.seasonal)
```

```
Out[62]: [<matplotlib.lines.Line2D at 0x20e77e9dee0>]
```

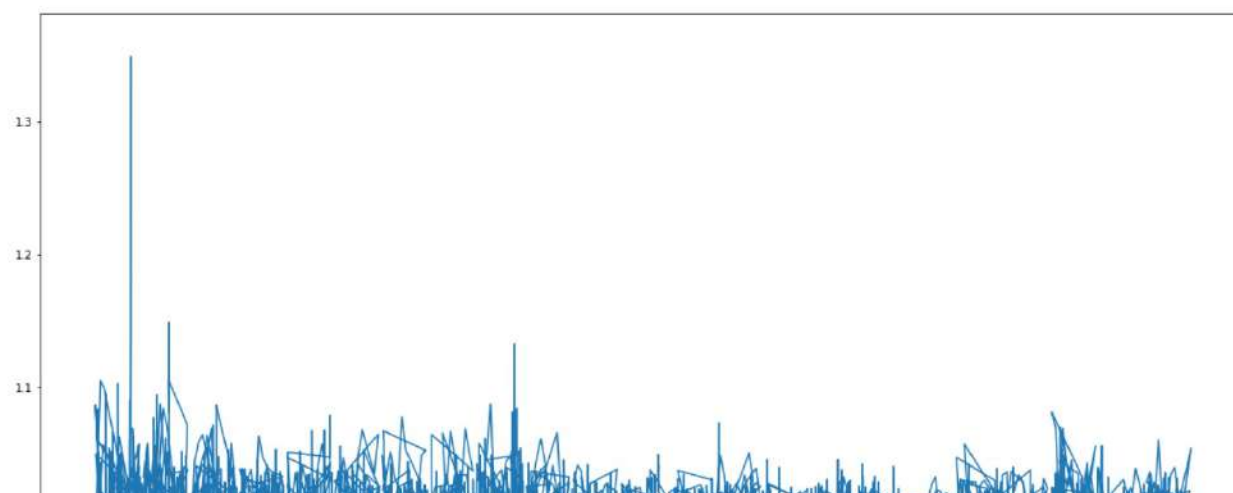


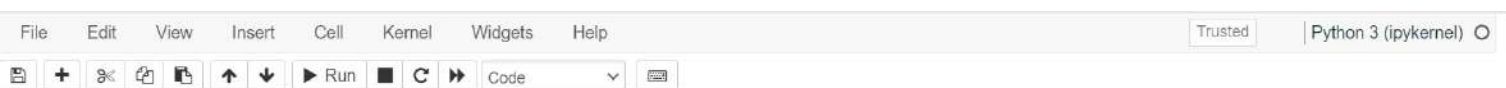
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel) 

        Run    Code  

```
In [63]: plt.figure(figsize=(20,16))  
plt.plot(result.resid)
```

```
Out[63]: [<matplotlib.lines.Line2D at 0x20e77ee0700>]
```





Exponential smoothing method

```
In [64]: from statsmodels.tsa.holtwinters import ExponentialSmoothing
```

```
In [65]: model = ExponentialSmoothing(Apple_stock, seasonal_periods=12, trend='add', seasonal='multiplicative').fit()

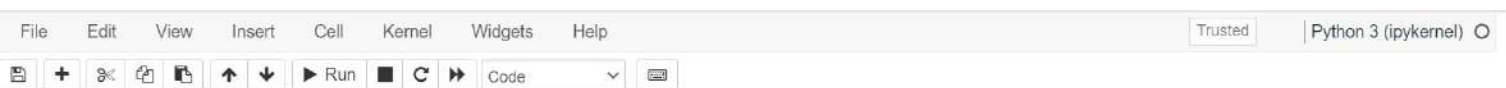
C:\Users\K. SAIKRISHNA REDDY\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: A date index has been provided, but it has no associated frequency information and so will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
C:\Users\K. SAIKRISHNA REDDY\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: A date index has been provided, but it is not monotonic and so will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
```

```
In [66]: predictions = model.forecast(12)

C:\Users\K. SAIKRISHNA REDDY\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:834: ValueWarning: No supported index is available. Prediction results will be given with an integer index beginning at 'start'.
  return get_prediction_index(
```

```
In [67]: print(predictions)
```

```
5745    155.559098
5746    155.535337
5747    155.637159
5748    155.977872
```



Augmented Dickey-Fuller (ADF)

```
In [68]: from statsmodels.tsa.stattools import adfuller
```

```
In [69]: ADF_result = adfuller(Apple_stock)
```

```
In [70]: # print the test statistic and p-value
print("Test statistic: ", ADF_result[0])
print("p-value: ", ADF_result[1])
```

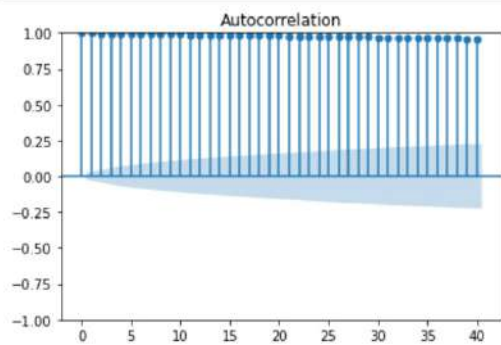
```
Test statistic: 1.7607826150146444
p-value: 0.9982679800859918
```

The ADF test returns a test statistic and a p-value. If the p-value is less than the significance level (typically 0.05), then we can reject the null hypothesis that the time series is non-stationary and conclude that the time series is stationary. If the p-value is greater than the significance level, then we cannot reject the null hypothesis and cannot conclude that the time series is stationary.

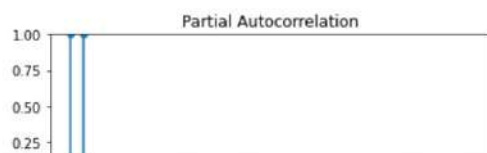
```
In [71]: from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
```

```
In [72]: plot_acf(Apple_stock, lags=40)
plt.show()
```

... Autocorrelation



```
In [73]: plot_pacf(Apple_stock, lags=32)  
plt.show()
```



File
Edit
View
Insert
Cell
Kernel
Widgets
Help

Run
Code

Trusted
Python 3 (ipykernel)

In [75]: market_metric_combined.head()

Out[75]:

	Symbol	Name	Market Cap	Last Sale	Net Change	company	sector	subsector	asset_turnover_2017	asset_turnover_2018	asset_turnover_2019	asset_turnover_2020
0	0	12	2625740143000	151.45	2.00	14	5	38	0.66	0.72	0.74	0.74
1	1	5	69569944167	116.65	0.26	5	1	23	0.60	0.55	0.64	0.64
2	2	3	149144569000	320.81	4.59	4	5	2	0.54	0.54	0.57	0.57
3	3	11	75484763090	146.76	2.23	12	5	34	0.36	0.30	0.29	0.29
4	4	17	98332762086	236.78	0.13	0	5	10	0.60	0.34	0.34	0.34

In [76]: market_metric = pd.merge(market_cap,metric_ratios, how='inner')

In [77]: market_metric.drop(X, axis=1, inplace = True)

In [78]: market_metric.drop(missing_percentage_list, axis=1, inplace=True)

In [79]: for i in columns_for_missing_value_treatment:
median = market_metric[i].median()
market_metric[i].fillna(median, inplace = True)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel) ○

Run Code

```
In [80]: sectors = market_metric["sector"].unique()
```

```
In [81]: sectors
```

```
Out[81]: array(['Information Technology', 'Consumer Discretionary', 'Utilities',  
               'Health Care', 'Communication Services', 'Consumer Staples',  
               'Industrials'], dtype=object)
```

```
In [82]: for i in sectors:  
         dataset = market_metric[market_metric["sector"]==i]  
         market_cap_asc = dataset.sort_values(by='Market Cap', ascending=False, inplace=True)  
         #print("sector: ", i)  
         company = dataset.head(2)[["Symbol"]]  
         #print("top 2 companies based on market capitalization: ", dataset.head(2)[["company"]])  
         L = []  
         L.append(company.values)  
         print(L)
```

```
[array(['AAPL'],  
       ['MSFT']], dtype=object)  
[array(['AMZN'],  
       ['TSLA']], dtype=object)  
[array(['AEP'],  
       ['EXC']], dtype=object)  
[array(['AZN'],  
       ['AMGN']], dtype=object)]
```

```
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

In [127]: AAPL = pd.read_csv("AAPL.csv")
MSFT = pd.read_csv("MSFT.csv")
AMZN = pd.read_csv("AMZN.csv")
TSLA = pd.read_csv("TSLA.csv")
AEP = pd.read_csv("AEP.csv")
EXC = pd.read_csv("EXC.csv")
AZN = pd.read_csv("AZN.csv")
AMGN = pd.read_csv("AMGN.csv")
GOOG = pd.read_csv("GOOG.csv")
GOOGL = pd.read_csv("GOOGL.csv")
PEP = pd.read_csv("PEP.csv")
COST = pd.read_csv("COST.csv")
HON = pd.read_csv("HON.csv")
CSX = pd.read_csv("CSX.csv")

In [128]: list_of_top_companies = [AAPL, MSFT, AMZN, TSLA, AEP, EXC, AZN, AMGN, GOOG, GOOGL, PEP, COST, HON, CSX]

In [95]: ADF_test_list = []
P_value_list = []
def stationarity(data):
    # calculate rolling mean
    data["rolling_mean"] = data["Adj Close"].rolling(window=12).mean()

    # calculate rolling standard deviation
    data["rolling_std"] = data["Adj Close"].rolling(window=12).std()
```



```
# print test statistic and p-value  
  
print("ADF test statistic: ", result[0])  
print("p-value: ", result[1])
```

```
In [96]: > for i in list_of_top_companies:
```

```
    stationarity(i)
```

```
ADF test statistic: 1.8652717020537568  
p-value: 0.998470053348639  
ADF test statistic: 1.3208286140504253  
p-value: 0.9967230480973174  
ADF test statistic: -0.3140803253030758  
p-value: 0.923486327753445  
ADF test statistic: -0.5860791573234889  
p-value: 0.8740966430735331  
ADF test statistic: 0.4522852391686381  
p-value: 0.9833483738436335  
ADF test statistic: -1.060630529563454  
p-value: 0.7305140440741887  
ADF test statistic: 0.6190633626792222  
p-value: 0.9880762034351294  
ADF test statistic: 1.299783913744627  
p-value: 0.9966077257848074  
-----
```

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel) ○

📄 + 🔍 📌 📁 ⬆️ ⬇️ ▶️ Run ■ ↺️ ▶️ Code ▾ 🗨️

```
In [102]: company = ["AAPL", "MSFT", "AMZN", "TSLA", "AEP", "EXC", "AZN", "AMGN", "GOOG", "GOOGL", "PEP", "COST", "HON", "CSX"]
```


```
In [103]: dict = {"Company_symbol" : company,
                 "ADF_test_result": ADF_test_list,
                 "p_value_result": P_value_list}
```












```
In [105]: top_companies_dataframe = pd.DataFrame(dict)
```

```
In [106]: top_companies_dataframe
```

Out[106]:

	Company_symbol	ADF_test_result	p_value_result
0	AAPL	1.865272	0.998470
1	MSFT	1.320829	0.996723
2	AMZN	-0.314080	0.923486
3	TSLA	-0.586079	0.874097
4	AEP	0.452285	0.983348
5	EXC	-1.060631	0.730514
6	AZN	0.619063	0.988076
7	AMGN	1.299784	0.996608
8	GOOG	-0.027612	0.956188

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel) 

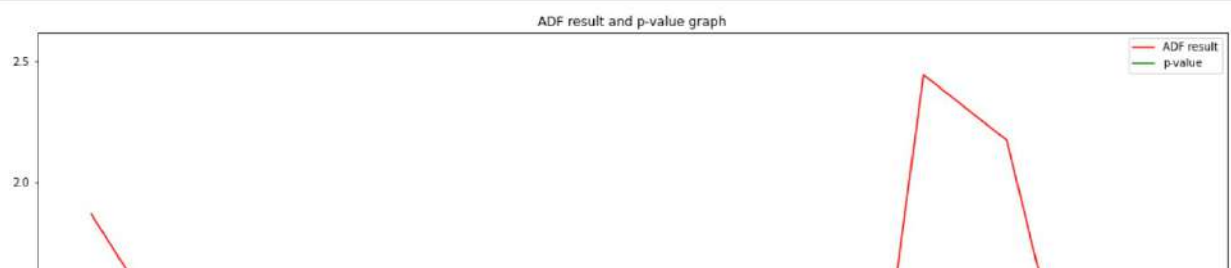
          Code  

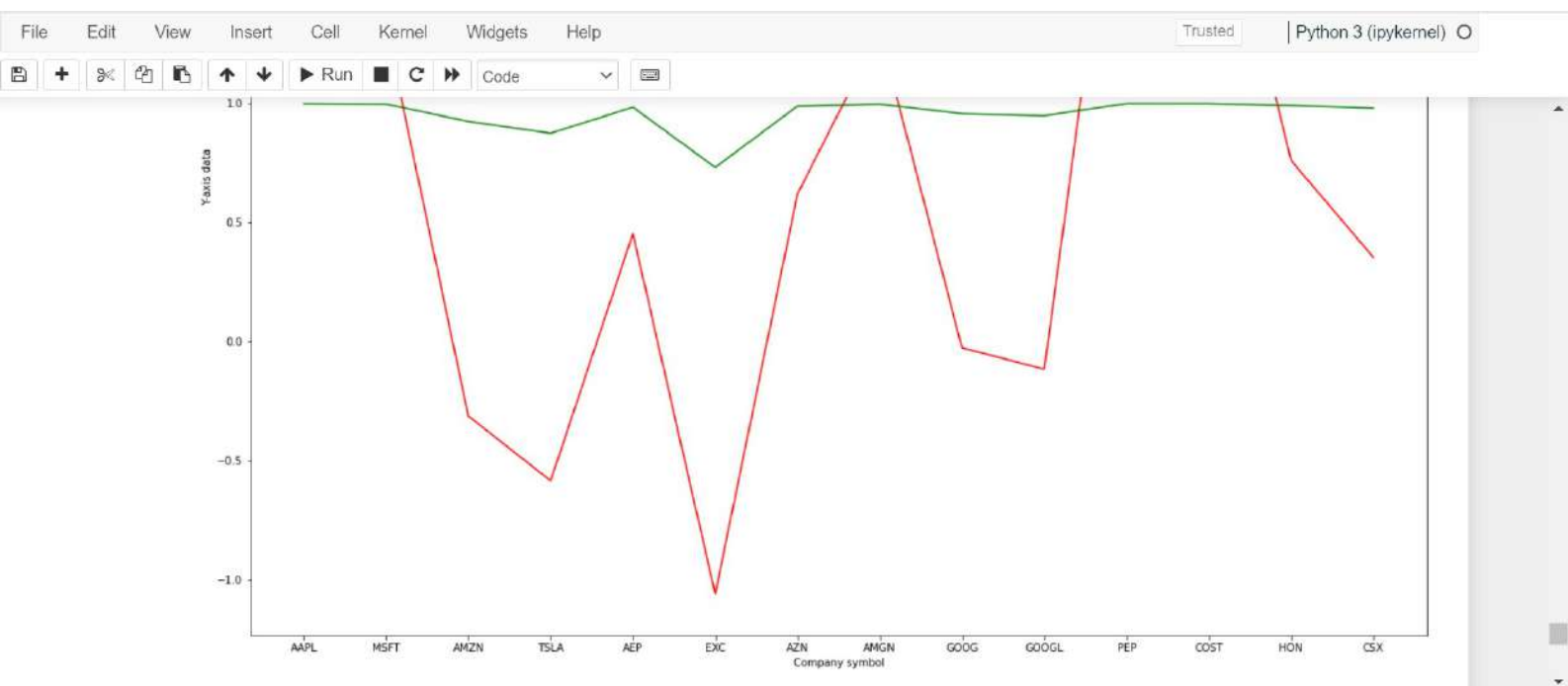
```
In [111]: plt.figure(figsize=(20,16))

x = top_companies_dataframe["Company_symbol"]
y1 = top_companies_dataframe["ADF_test_result"]
y2 = top_companies_dataframe["p_value_result"]

plt.plot(x, y1, label='ADF result',color='red')
plt.plot(x, y2, label='p-value',color='green' )

plt.xlabel("Company symbol")
plt.ylabel("Y-axis data")
plt.legend()
plt.title('ADF result and p-value graph')
plt.show()
```





```
Performing stepwise search to minimize aic
ARIMA(2,2,2)(0,0,0)[0] : AIC=inf, Time=1.69 sec
ARIMA(0,2,0)(0,0,0)[0] : AIC=20811.299, Time=0.09 sec
ARIMA(1,2,0)(0,0,0)[0] : AIC=19034.883, Time=0.14 sec
ARIMA(0,2,1)(0,0,0)[0] : AIC=inf, Time=0.58 sec
ARIMA(2,2,2)(0,0,0)[0] : AIC=inf, Time=1.69 sec
```



FileHomeInsertDrawPage LayoutFormulasDataReviewViewDeveloperHelpTell me what you want to do


Paste

Clipboard

Calibri11A⁺A⁺

B*I*U






Wrap Text

Merge & Center

General



Conditional Formatting


Format as Table


Cell Styles


Insert

Delete




Format

Σ

Sort & Filter

Find & Select

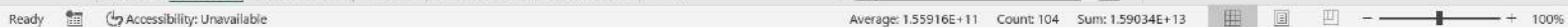
G27

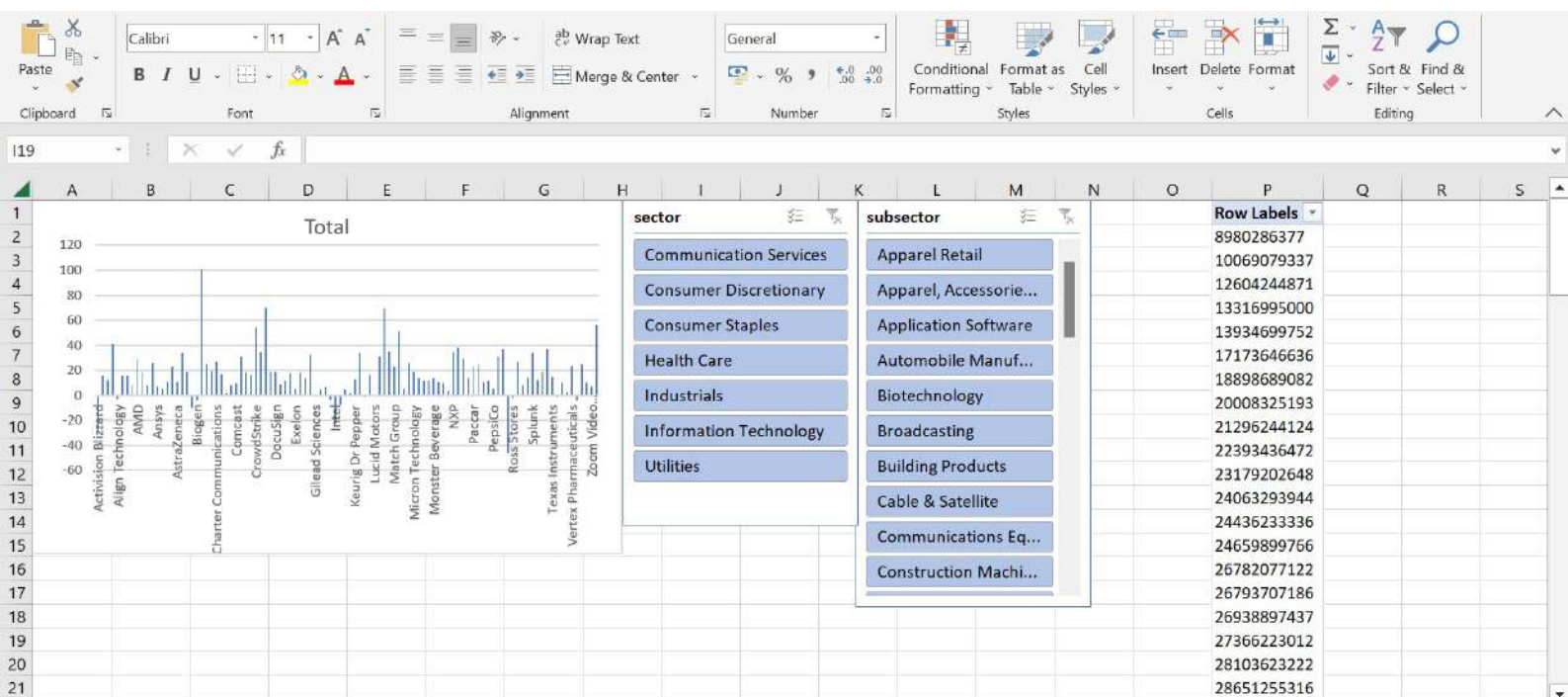


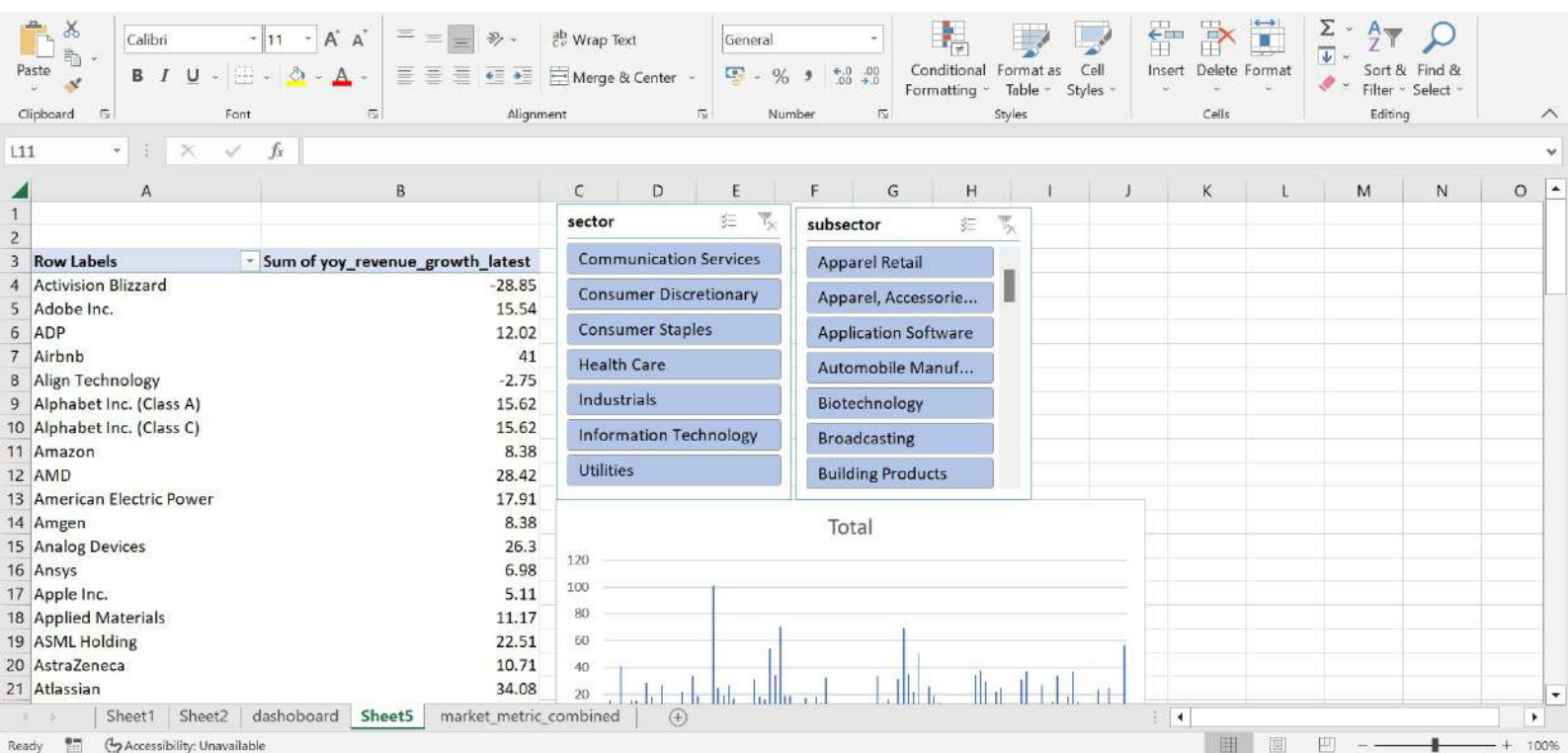
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1															
2															
3	Row Labels														
4	Communication Services														
5	1.3556E+12														
6	Alphabet Inc. (Class C)														
7	1.35142E+12														
8	Alphabet Inc. (Class A)														
9	3.63316E+11														
10	Meta Platforms														
11	1.76958E+11														
12	T-Mobile US														
13	1.38951E+11														
14	Comcast														
15	1.30451E+11														
16	Netflix														
17	70858233390														
18	Charter Communications														
19	56971477562														
20	Activision Blizzard														
21	37568854409														

Sheet1Sheet2dashboardSheet5market_metric_combined

ReadyAccessibility: Unavailable100%







FileEditViewQueryDatabaseServerToolsScriptingHelp

SQL File 3*

Limit to 1000 rows

Jump to

1

•

create database stock_market;

Schemas

Filter objects

customers

employee

health

Tables

hospitalisationdetails

medicalexaminations

names

new_table

Views

Stored Procedures

Functions

sample

students

Administration

Schemas

No object selected

Object Info

Session

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help

Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 1	13:46:29	create database stock_market	1 row(s) affected	0.031 sec

FileEditViewQueryDatabaseServerToolsScriptingHelp

SQL File 3"

Limit to 1000 rows

1 • create database stock_market;

2 • use stock_market;

3 • show tables;

4 • select * from 'nasdaq 100 market cap';

5 • select * from nasdaq100_metrics_ratios;

6

Result Grid

Filter Rows:

Exports:

Wrap Cell Contents:

i=Symbol	company	sector	subsector	asset_turnover_2017
AMGN	Amgen	Health Care	Biotechnology	0.29
ASML	ASML Holding	Information Technology	Semiconductor Equipment	0.54
ATVI	Activision Blizzard	Communication Services	Interactive Home Entertainment	0.39
AVGO	Broadcom Inc.	Information Technology	Semiconductors	0.34
BIIB	Biogen	Health Care	Biotechnology	0.53

nasdaq100_metrics_ratios 4

Read OnlyContext HelpSnippets

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Navigator

SCHEMAS

Filter objects

Last Sale

Net Change

Percentage Change

Indexes

Foreign Keys

Triggers

nasdaq100_metrics_ratios

Views

Stored Procedures

Functions

students

sys

AdministrationSchemas

Information

Column: i=Symbol

Collation: utf8mb4_0900_ai_ci

Definition: i=Symbol text

Object InfoSession

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 17	13:54:37	show tables	2 row(s) returned	0.016 sec / 0.000 sec
✓ 18	13:55:04	select * from 'nasdaq 100 market cap' LIMIT 0, 1000	102 row(s) returned	0.000 sec / 0.000 sec
✓ 19	13:55:41	select * from nasdaq100_metrics_ratios LIMIT 0, 1000	41 row(s) returned	0.031 sec / 0.000 sec
✓ 20	13:55:48	select * from nasdaq100_metrics_ratios LIMIT 0, 1000	41 row(s) returned	0.016 sec / 0.000 sec

FileEditViewQueryDatabaseServerToolsScriptingHelp

Navigator

SCHEMAS

Filter objects

customers

employee

health

sample

stock_market

Tables

market_metric_combined

Columns

MyUnknownColumn

Symbol

Name

Market Cap

AdministrationSchemas

Information

Column: MyUnknownColumn

Definition:
MyUnknownColumn int

Object InfoSession

SQL File 3*

Limit to 1000 rows

1 • create database stock_market;

2 • use stock_market;

3

4 • select * from market_metric_combined;

5

Result Grid

Filter Rows: | Export: | Wrap Cell Contents: |

t Cap	Last Sale	Net Change	Percentage Change	company	sector	subsector
10143000	\$151.45	\$2.00	0.0134	Apple Inc.	Information Technology	Technology H
944167	\$116.65	\$0.26	0.0022	Airbnb	Consumer Discretionary	Internet & Di
1569000	\$320.81	\$4.59	0.0145	Adobe Inc.	Information Technology	Application S
763090	\$146.76	\$2.23	0.0154	Analog Devices	Information Technology	Semiconduct
762096	\$736.78	\$0.13	-0.0005	and	Information Technology	Data Process

market_metric_combined 5 x

Read OnlyContext HelpSnippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 36	21:21:47	CREATE TABLE 'stock_market'.market_metric_combined ('MyUnknow...	OK	0.000 sec
✓ 37	21:21:47	PREPARE stmt FROM 'INSERT INTO 'stock_market'.market_metric_co...	OK	0.000 sec
✓ 38	21:21:54	DEALLOCATE PREPARE stmt	OK	0.000 sec
✓ 39	21:22:45	select * from market_metric_combined LIMIT 0, 1000	102 row(s) returned	0.000 sec / 0.016 sec

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Navigator

SCHEMAS

Filter objects

Last Sale

Net Change

Percentage Change

company

sector

subsector

asset_turnover_2017

asset_turnover_2018

asset_turnover_2019

asset_turnover_2020

asset_turnover_2021

asset turnover latest

Administration

Schemas

Information

Column: MyUnknownColumn

Definition:
MyUnknownColumn int

Object Info

Session

SQL File 3*

Limit to 1000 rows

6

•

select 'Market Cap', company, sector from market_metric_combined

7

where sector="Information Technology" ORDER BY 'Last Sale' DESC

8

LIMIT 1;

9

10

11

12

Result Grid

Filter Rows:

Exports

Wrap Cell Contents

Fetch rows:

Result Grid

Form Editor

Market Cap	company	sector
63386006809	Fiserv	Information Technology

market_metric_combined 7

Read Only

Context Help

Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
39	21:22:45	select * from market_metric_combined LIMIT 0, 1000	102 row(s) returned	0.000 sec / 0.016 sec
40	21:25:47	select 'Market Cap', company from market_metric_combined where secto...	Error Code: 1111. Invalid use of group function	0.000 sec
41	21:34:15	select 'Market Cap', company from market_metric_combined where secto...	1 row(s) returned	0.000 sec / 0.000 sec
42	21:34:33	select 'Market Cap', company, sector from market_metric_combined wher...	1 row(s) returned	0.000 sec / 0.000 sec

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

FileEditViewQueryDatabaseServerToolsScriptingHelp

Navigator

SCHEMAS

Filter objects

customers

employee

health

sample

stock_market

Tables

market_metric_combined

nasdaq 100 market cap

Views

Stored Procedures

Functions

students

svs

Administration

Schemas

Information

Schema: stock_market

SQL File 3*

Limit to 1000 rows

SQL

11

12 • SELECT sector, COUNT(*) as NumCompanies

13 FROM market_metric_combined

14 GROUP BY sector;

15

16

17

Result Grid

Filter Rows:

Exports

Wrap Cell Contents

Result Grid

Form Editor

sector	NumCompanies
Information Technology	42
Consumer Discretionary	15
Utilities	4
Health Care	13
Communication Services	13
Consumer Staples	7

Result 9 x

Read Only

Context Help

Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 48	21:39:34	DEALLOCATE PREPARE stmt	OK	0.000 sec
✓ 49	21:44:23	select * from 'nasdaq 100 market cap' LIMIT 0, 1000	102 row(s) returned	0.000 sec / 0.000 sec
✗ 50	21:47:36	select count(company) from market_metrics_combined group by sector LI...	Error Code: 1146. Table 'stock_market.market_metrics_combined' doesn't...	0.000 sec
✓ 51	21:50:52	SELECT sector, COUNT(*) as NumCompanies FROM market_metric_com...	7 row(s) returned	0.015 sec / 0.000 sec

Object InfoSession

SQLAdditions

< > | | Jump to

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

FileEditViewQueryDatabaseServerToolsScriptingHelp

Navigator

SCHEMAS

Filter objects

market_metric_combined

Columns

MyUnknownColumn

Symbol

Name

Market Cap

Last Sale

Net Change

Percentage Change

company

sector

subsector

Administration

Schemas

Information

Schema: stock_market

SQL File 3"

Limit to 1000 rows

12

•

SELECT sector, COUNT(*) as NumCompanies

13

FROM market_metric_combined

14

GROUP BY sector;

15

16

17

•

select company from market_metric_combined order by 'Market Cap' limit 5;

18

Result Grid

Filter Rows:

Exports

Wrap Cell Contents

Fetch rows:

Result Grid

Form Editor

market_metric_combined 10

Read Only

Context Help

Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
49	21:44:23	select * from 'nasdaq 100 market cap' LIMIT 0, 1000	102 row(s) returned	0.000 sec / 0.000 sec
50	21:47:36	select count(company) from market_metrics_combined group by sector LI...	Error Code: 1146. Table 'stock_market.market_metrics_combined' doesn't...	0.000 sec
51	21:50:52	SELECT sector, COUNT(*) as NumCompanies FROM market_metric_com...	7 row(s) returned	0.015 sec / 0.000 sec
52	21:52:36	select company from market_metric_combined order by 'Market Cap' limit 5	5 row(s) returned	0.000 sec / 0.000 sec

Object Info

Session

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- market_metric_combined
 - Columns
 - MyUnknownColumn
 - Symbol
 - Name
 - Market Cap
 - Last Sale
 - Net Change
 - Percentage Change
 - company
 - sector
 - subsector

Administration Schemas

Information

Schema: stock_market

SQL File 3*

```
20 • create table sector_subsector_count
21 SELECT sector, subsector, COUNT(*) as NumCompanies
22 FROM market_metric_combined
23 GROUP BY sector, subsector;
24
25 • select * from sector_subsector_count;
26
```

Limit to 1000 rows

Result Grid

sector	subsector	NumCompanies
Information Technology	Technology Hardware, Storage & Peripherals	1
Consumer Discretionary	Internet & Direct Marketing Retail	7
Information Technology	Application Software	17
Information Technology	Semiconductors	11
Information Technology	Data Processing & Outsourced Services	4
Utilities	Electric Utilities	1

sector_subsector_count 12 x

Read Only Context Help Snippets

Output

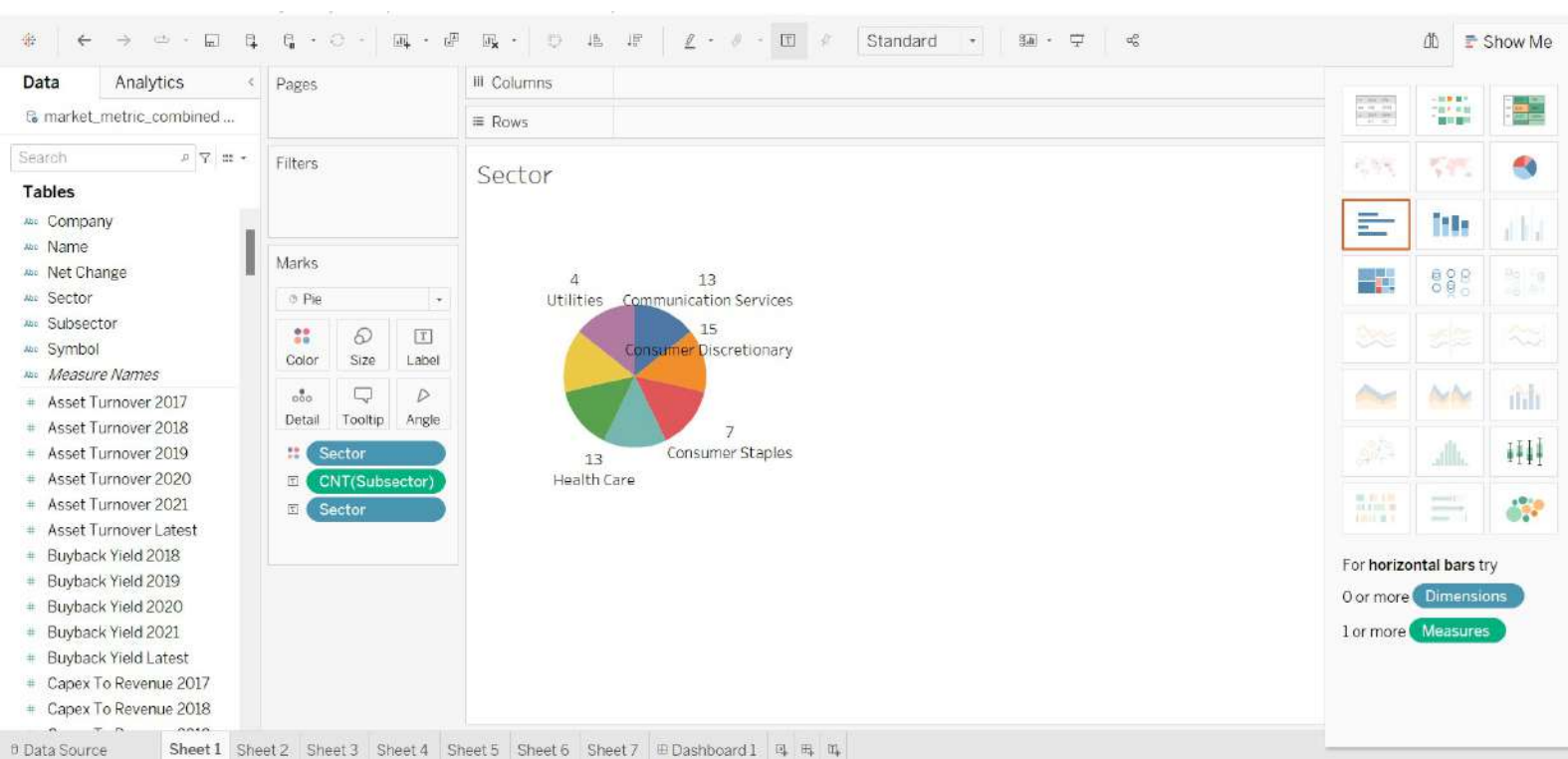
Action Output

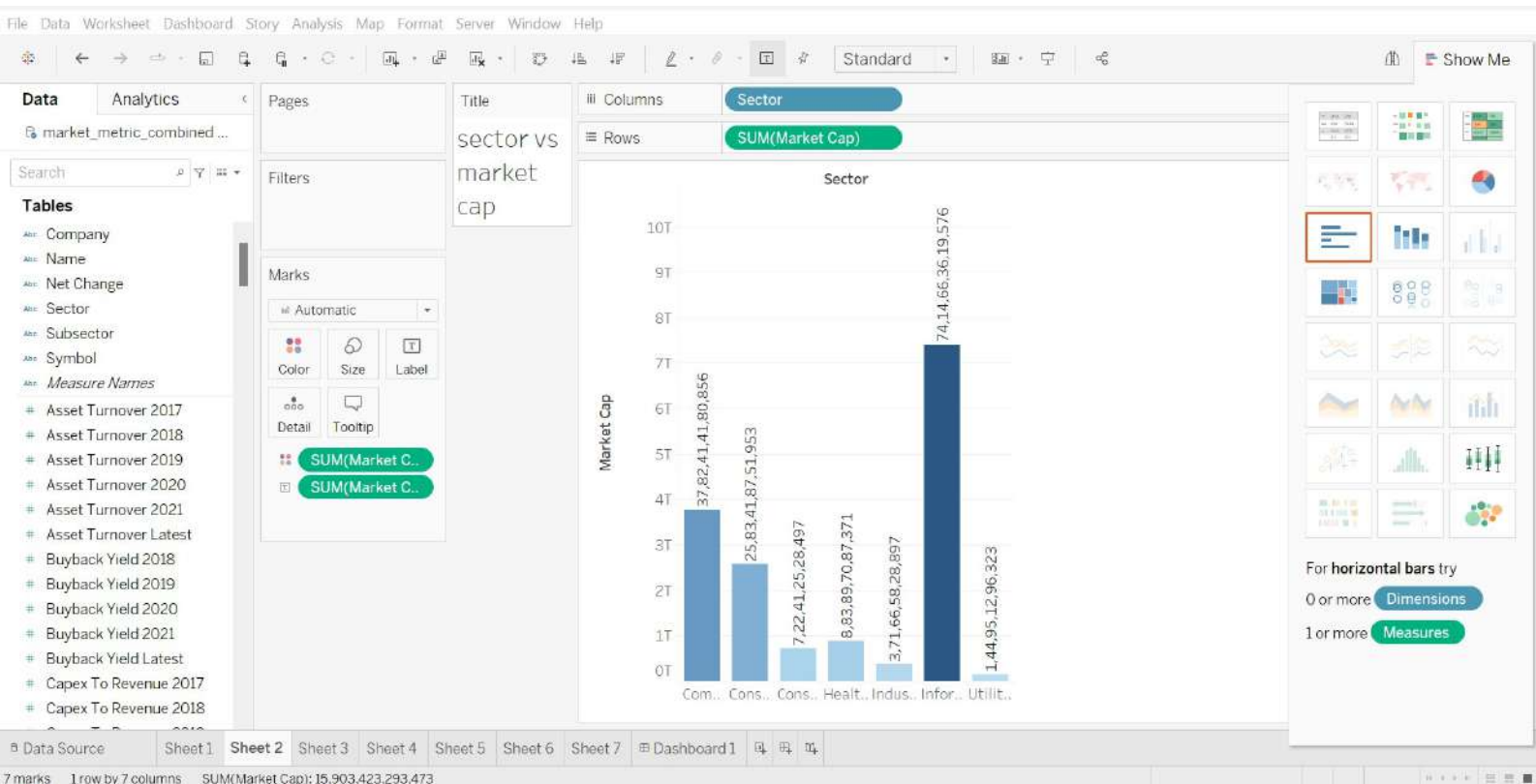
#	Time	Action	Message	Duration / Fetch
55	21:59:12	INSERT INTO sector_subsector_count SELECT sector, subsector, COU...	Error Code: 1146. Table 'stock_market.sector_subsector_count' doesn't e...	0.000 sec
56	21:59:28	SELECT sector, subsector, COUNT(*) as NumCompanies FROM market_...	41 row(s) returned	0.016 sec / 0.000 sec
57	21:59:53	create table sector_subsector_count SELECT sector, subsector, COUNT(...	41 row(s) affected Records: 41 Duplicates: 0 Warnings: 0	0.078 sec
58	22:00:08	select * from sector_subsector_count LIMIT 0, 1000	41 row(s) returned	0.000 sec / 0.000 sec

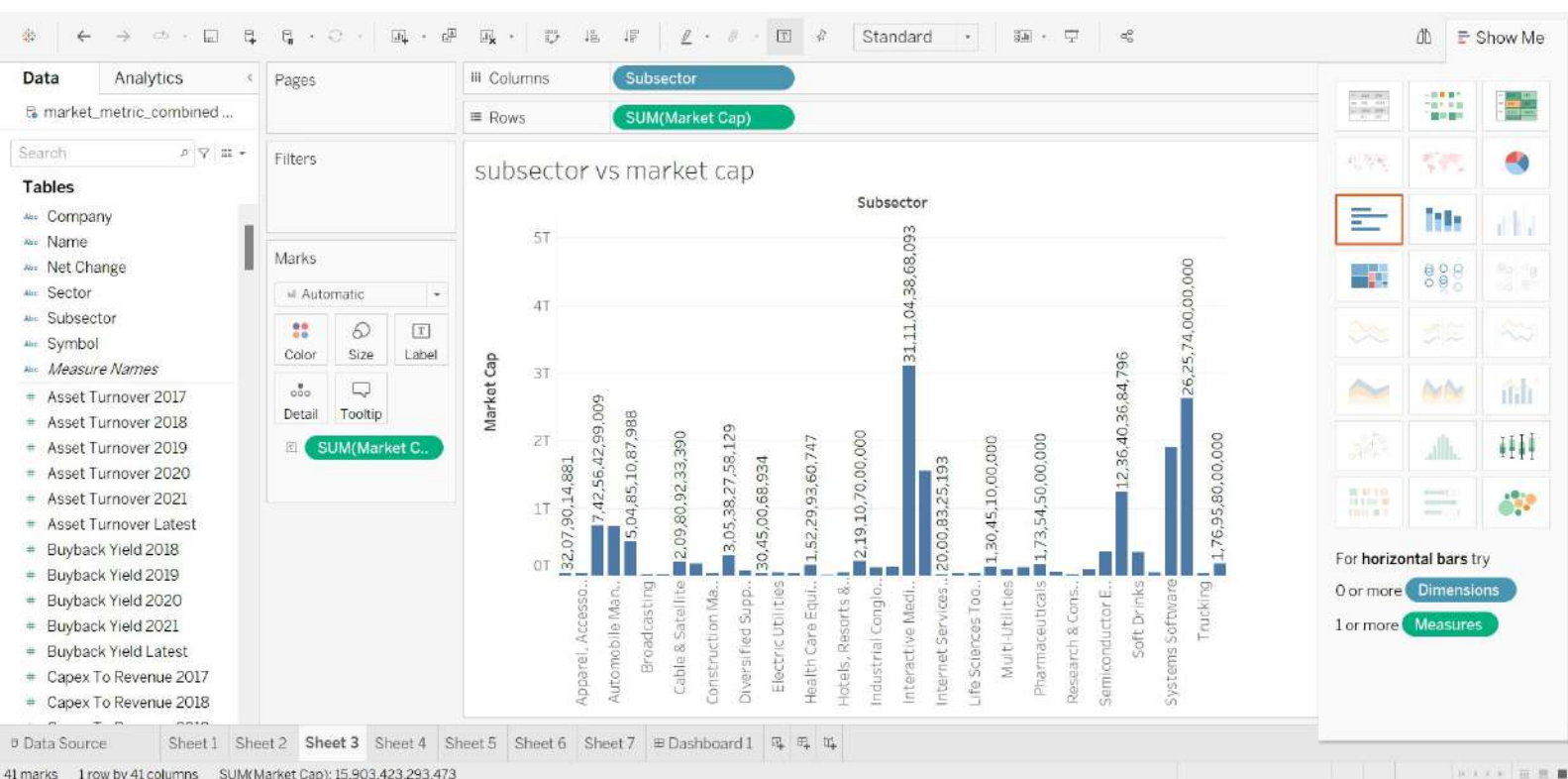
Object Info Session

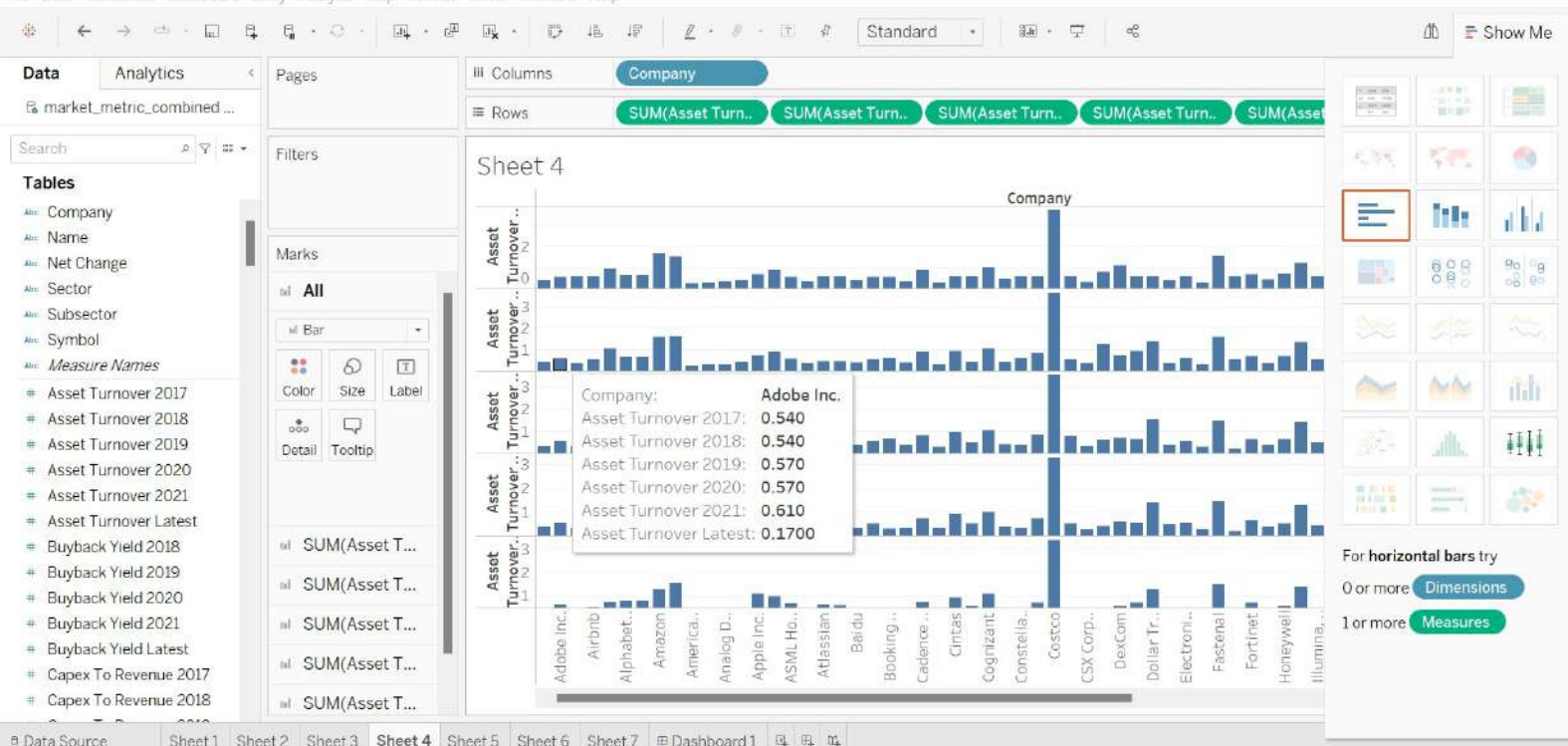
SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

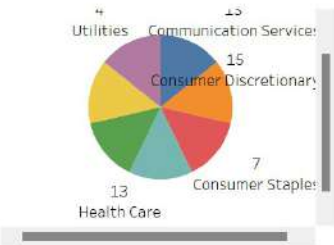




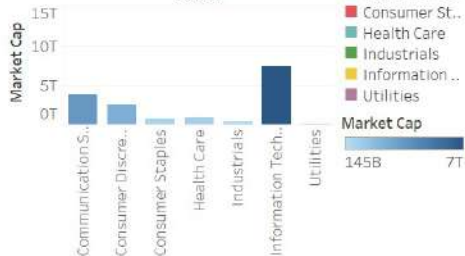




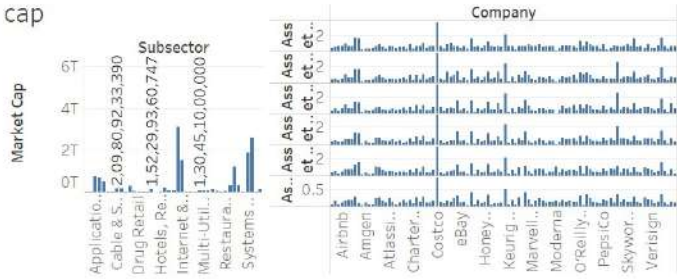
Sector



sector vs market cap



subsector vs market cap Sheet 4



▢ Data Source
Sheet 1
Sheet 2
Sheet 3
Sheet 4
Sheet 5
Sheet 6
Sheet 7
Dashboard 1
🔍
🔍
🔍

