Module 1 -By Suprateek Halsana In [1]: import numpy as np import pandas as pd import warnings warnings.filterwarnings('ignore') **Problem [1.1] Reading the Dataset** In [2]: data=pd.read csv(r"C:\Users\Suprateek Halsana\Documents\Python Scripts\Aspiring Mind Internship\LALPATH LAB [INDUSTRY - CHOSEN] \LALPATHLAB.csv") data Out[2]: Total Prev Open High Low Last Close Average No. of Deliverab Symbol Series Date Traded Turnover Close **Price Price Price Price Price Price Trades** Q Quantity 15-LALPATHLAB EQ May-891.15 895.0 881.00 912.00 900.60 889.35 257655 2.291453e+08 15451 17477 914.15 2017 16-1 LALPATHLAB 914.57 EQ May-900.60 910.0 925.00 895.05 909.40 910.95 104896 9.593430e+07 13001 758 2017 17-2 LALPATHLAB May-910.95 913.0 925.00 909.00 912.05 911.70 917.19 72460 6.645960e+07 2802 5382 2017 18-3 LALPATHLAB May-909.75 EQ 911.70 908.0 919.35 903.05 906.00 914.12 37707 3.446889e+07 2731 2483 2017 19-LALPATHLAB May-909.75 917.00 905.80 910.00 910.25 910.61 81082 7.383375e+07 4430 6992 2017 07-490 LALPATHLAB EQ May-1024.95 1015.7 1031.95 1006.90 1013.00 1013.10 1020.66 20113 2.052851e+07 3628 1158 2019 08-LALPATHLAB EQ 1006.10 1120 491 May-1013.10 1014.0 1019.85 1001.90 1002.00 1010.90 18048 1.824473e+07 4175 2019 09-492 LALPATHLAB May-1006.10 1014.0 1014.00 977.70 979.90 982.90 987.36 60539 5.977355e+07 8169 3730 2019 10-493 LALPATHLAB EQ May-982.90 989.9 994.30 963.60 980.00 980.65 980.71 30568 2.997840e+07 4618 1909 2019 13-LALPATHLAB EQ May-980.65 970.8 996.00 960.00 975.00 976.25 974.21 37187 3.622810e+07 6798 237 2019 495 rows × 15 columns In [3]: data.describe() Out[3]: **Total Traded** Average **Prev Close** Open Price **High Price Low Price Last Price** Close Price Turnover No. of Tr Price Quantity 495.000000 count 495.000000 495.000000 495.000000 495.000000 495.000000 495.000000 4.950000e+02 4.950000e+02 81.540021 79.871031 83.917032 77.560519 80.225784 80.180802 81.178892 1.389009e+05 1.172821e+08 4103.64 std 550.000000 min 748.450000 749.000000 723.050000 739.000000 737.350000 734.620000 3.903000e+03 3.460254e+06 1.00 2.948100e+04 2.734100e+07 25% 857.300000 857.750000 872.300000 844.625000 854.375000 857.925000 859.495000 2855.50 908.000000 897.150000 4.739300e+04 4.479719e+07 50% 896.200000 898.000000 882.050000 896.000000 895.280000 4470.00 953.550000 75% 956.400000 973.950000 940.475000 955.250000 958.300000 957.885000 9.157100e+04 8.213189e+07 7027.50 1089.500000 1100.000000 1071.300000 1089.500000 42803.00 max 1125.000000 1096.000000 1096.910000 2.047290e+06 1.603652e+09 In [4]: data.isnull().sum() Out[4]: Symbol 0 Series 0 Date 0 0 Prev Close Open Price 0 High Price Low Price 0 Last Price Close Price 0 Average Price Total Traded Quantity 0 Turnover 0 No. of Trades 0 Deliverable Qty 0 % Dly Qt to Traded Qty 0 dtype: int64 In [5]: data['Series'].isnull().sum() Out[5]: 0 In [6]: (data['Series'] == 'EQ').sum() Out[6]: 494 (data['Series'] == 'BL').sum() Out[7]: 1 Removal of non EQ Series In [8]: data=data[data['Series']=='EQ'] In [9]: data.shape (494, 15) Out[9]: In [10]: data.describe() Out[10]: Average **Total Traded Close Price** Turnover No. of Tr **Prev Close Open Price High Price Low Price Last Price Price** Quantity 494.000000 494.000000 494.000000 494.000000 494.000000 494.000000 494.000000 4.940000e+02 4.940000e+02 494.00 count 907.374494 5471.98 908.446457 909.126113 922.827834 909.155061 8.664805e+04 7.770704e+07 mean 894.349798 908.618725 std 80.013571 79.758601 83.856648 77.389389 80.109065 80.067847 1.390295e+05 1.173757e+08 4100.42 737.350000 748.450000 749.000000 723.050000 739.000000 737.350000 734.620000 3.903000e+03 3.460254e+06 284.00 min 859.447500 2.944900e+04 2.731598e+07 857.762500 857.625000 872.050000 844.437500 854.062500 857.762500 2863.50 25% 895.120000 4.722250e+04 4.462565e+07 4475.50 50% 896.650000 897.975000 907.950000 882.025000 896.000000 897.125000 957.050000 953.037500 973.562500 939.675000 955.000000 957.962500 956.885000 9.080075e+04 8.202401e+07 7041.25 1071.300000 1089.500000 1125.000000 1096.000000 1089.500000 2.047290e+06 1.603652e+09 42803.00 1100.000000 1096.910000 max Problem [1.2] last 90 days data [11]: 190data=data.iloc[404:495,:] 190data.describe() Out[11]: **Total Traded** Average **Prev Close Open Price Low Price Last Price Close Price** Turnover No. of **High Price Price** Quantity 90.000000 90.000000 90.000000 90.0 90.000000 90.000000 90.000000 90.000000 90.000000 9.000000e+01 count 1026.846667 1028.133889 1046.501667 1011.066111 1026.322778 1027.599444 1029.301667 7.299330e+07 6945.0 mean 71588.833333 42.463655 43.942561 45.082417 39.397593 41.507389 42.441683 67742.618587 4203.9 std 40.914339 6.831324e+07 908.500000 908.000000 918.000000 890.950000 909.950000 906.080000 7704.000000 8.185802e+06 1875.(min 910.300000 27594.000000 25% 1013.975000 1014.000000 1029.000000 1001.962500 1012.325000 1013.975000 1017.620000 2.878656e+07 3955.2 1017.700000 1029.500000 1032.450000 1035.450000 1051.850000 1032.450000 1033.480000 47222.500000 4.999364e+07 5694.0 50% 1079.875000 75% 1056.837500 1057.812500 1038.000000 1052.250000 1056.837500 1059.605000 87892.000000 8.547156e+07 8329.5 1089.500000 1100.000000 1125.000000 1071.300000 1096.000000 1089.500000 1096.910000 341688.000000 3.521682e+08 20729.0 Problem [1.3] Date converted to datetime64[ns] data['Date'] = pd.to datetime(data['Date']) In [12]: In [13]: data Out[13]: Total High Average Deliveral Prev Open Close No. of Low Last Symbol Series Date Traded **Turnover** Close Price **Price Price** Price Price Price **Trades** Quantity 2017-0 LALPATHLAB EQ 891.15 895.0 914.15 881.00 912.00 900.60 889.35 257655 2.291453e+08 15451 1747 05-15 2017-LALPATHLAB EQ 900.60 910.0 925.00 895.05 909.40 910.95 914.57 104896 9.593430e+07 13001 758 05-16 2017-2 LALPATHLAB EQ 910.95 913.0 925.00 909.00 912.05 911.70 917.19 72460 6.645960e+07 2802 538 05-17 2017-3 LALPATHLAB EQ 911.70 908.0 919.35 903.05 906.00 909.75 914.12 37707 3.446889e+07 2731 248 05-18 2017-LALPATHLAB EQ 909.75 917.0 917.00 905.80 910.00 910.25 910.61 81082 7.383375e+07 4430 699 05-19 2019-490 LALPATHLAB EQ 1024.95 1015.7 1031.95 1006.90 1013.00 1013.10 1020.66 20113 2.052851e+07 3628 115 2019-491 LALPATHLAB EQ 1013.10 1014.0 1019.85 1001.90 1002.00 1006.10 1010.90 18048 1.824473e+07 4175 112 05-08 2019-492 LALPATHLAB EQ 1006.10 1014.0 1014.00 977.70 979.90 982.90 987.36 60539 5.977355e+07 8169 373 05-09 2019-493 LALPATHLAB EQ 982.90 989.9 994.30 963.60 980.00 980.65 980.71 30568 2.997840e+07 4618 190 05-10 2019-494 LALPATHLAB EQ 980.65 970.8 996.00 960.00 975.00 976.25 974.21 37187 3.622810e+07 6798 237 494 rows × 15 columns Last 90 days Data in General (Another Approach) In [14]: max d=max(data['Date']) ndf=pd.DataFrame() df=pd.DataFrame() d = 90**while** (ndf.shape[0]<90): m=max d-pd.to timedelta(d,unit='days') r=(data['Date']>m) & (data['Date']<=max d)</pre> df=data.loc[r] ndf=pd.concat([df,ndf],axis=0) d=90-ndf.shape[0] $max_d=m$ del df ndf Out[14]: Total Prev Open High Low Last Close Average No. of Delivera Symbol Series Date **Traded Turnover** Close Price Price Price Price Price Price **Trades** Quantity 2018-**404** LALPATHLAB EQ 908.50 911.25 922.00 890.95 909.95 910.30 906.08 341688 3.095975e+08 20516 286 12-31 2019-LALPATHLAB EQ 910.30 909.95 928.00 902.45 925.00 924.80 912.79 1.438439e+08 4646 405 157587 112 01-01 2019-EQ LALPATHLAB 924.80 924.00 900.00 916.00 915.20 910.93 94182 8.579362e+07 4481 406 924.00 50 01-02 2019-LALPATHLAB EQ 915.20 909.70 918.00 897.10 910.00 913.05 907.57 6.512356e+07 3283 30 407 71756 01-03 2019-EQ 1.468668e+08 LALPATHLAB 913.05 908.00 935.45 940.10 937.58 7745 102 408 908.00 944.85 156645 01-04 2019-1020.66 LALPATHLAB 1024.95 1006.90 1013.00 1013.10 3628 490 1015.70 1031.95 20113 2.052851e+07 11 05-07 2019-1006.10 LALPATHLAB EQ 1013.10 1014.00 1019.85 1001.90 1002.00 1010.90 4175 491 18048 1.824473e+07 11 05-08 2019-LALPATHLAB 977.70 982.90 8169 492 1006.10 1014.00 1014.00 979.90 987.36 60539 5.977355e+07 37 05-09 2019-LALPATHLAB EQ 982.90 989.90 994.30 963.60 980.00 980.65 980.71 2.997840e+07 4618 493 30568 19 05-10 2019-LALPATHLAB EQ 974.21 6798 23 980.65 970.80 996.00 960.00 975.00 976.25 37187 3.622810e+07 05-13 90 rows × 15 columns In [15]: d=data.copy() d['Month'] = pd. DatetimeIndex (d['Date']) . month d['Year']=pd.DatetimeIndex(d['Date']).year d Out[15]: Total High Close **Average** Open Deliveral Prev Last No. of Low Symbol Series Date **Traded** Turnover Close Price **Price Price Price Price** Price **Trades** Quantity 2017-0 LALPATHLAB 912.00 891.15 895.0 914.15 881.00 900.60 889.35 257655 2.291453e+08 15451 1747 05-15 2017-909.40 910.95 LALPATHLAB EQ 900.60 910.0 925.00 895.05 914.57 104896 9.593430e+07 13001 758 05-16 2017-925.00 909.00 912.05 911.70 2 LALPATHLAB EQ 910.95 913.0 917.19 72460 6.645960e+07 2802 538 05-17 2017-919.35 914.12 3.446889e+07 LALPATHLAB EQ 911.70 908.0 903.05 906.00 909.75 37707 2731 248 05-18 2017-917.00 910.25 910.61 LALPATHLAB EQ 909.75 917.0 905.80 910.00 81082 7.383375e+07 4430 699 05-19 490 LALPATHLAB 1024.95 1015.7 1031.95 1006.90 1013.00 1013.10 1020.66 20113 2.052851e+07 3628 115 05-07 2019-LALPATHLAB 1013.10 1014.0 1019.85 1001.90 1002.00 1006.10 1010.90 18048 1.824473e+07 4175 112 491 05-08 LALPATHLAB 1014.0 492 EQ 1006.10 1014.00 977.70 979.90 982.90 987.36 60539 5.977355e+07 8169 373 05-09 2019-493 LALPATHLAB 982.90 989.9 994.30 963.60 980.00 980.65 980.71 30568 2.997840e+07 4618 190 05-10 LALPATHLAB 494 EQ 980.65 970.8 996.00 960.00 975.00 976.25 974.21 37187 3.622810e+07 6798 237 05-13 494 rows × 17 columns Problem [1.4] Grouped data according to Month and calculated **VWAP** In [16]: g=d.groupby(['Month']) In [17]: #shows only first value of each group g.first() #shows only last value of each group #g.last() Out[17]: Total Prev Open High Low Last Close Average No. of Deliverable Symbol Series Date Traded **Turnover Price Trades** Close Price Price Price Price Price Qty Quantity Month 2018-1 LALPATHLAB 28022 2.478059e+07 EQ 881.30 882.05 897.05 873.00 881.45 884.65 884.33 1536 13422 01-01 2018-2 LALPATHLAB 913.95 915.00 901.00 905.00 910.45 907.70 3.208438e+07 3342 20473 02-01 2018-LALPATHLAB EQ 908.00 914.90 908.00 910.08 416646 3.791796e+08 409538 1120 03-01 2018-1.107144e+07 4 LALPATHLAB 876.30 884.80 867.00 868.00 871.65 876.74 2439 6667 04-02 5 LALPATHLAB 895.00 914.15 881.00 912.00 889.35 2.291453e+08 174775 257655 15451 2017-LALPATHLAB 916.00 919.80 897.10 898.25 900.60 904.02 108277 9.788415e+07 4930 71398 06-01 2017-762.15 775.00 7 LALPATHLAB 781.95 783.00 787.45 219597 1.695540e+08 97471 772.11 07-03 2017-8 LALPATHLAB 836.90 838.00 821.00 822.00 824.10 825.40 39485 3.259108e+07 1863 22099 08-01 2017-804.00 805.00 LALPATHLAB 812.30 820.00 812.58 6.141965e+07 62733 09-01 2017-LALPATHLAB 795.05 801.85 775.00 776.95 790.15 794.33 41099 3.264616e+07 4021 30946 10-03 790.00 781.40 LALPATHLAB 772.55 787.90 775.25 779.90 5.629937e+07 5270 36399 2017-12 LALPATHLAB 889.80 886.00 890.00 882.00 890.00 889.25 889.26 75759 6.736913e+07 3573 38057 12-01 In [18]: #to get the values of a certain group #g.get_group((1,2018)).sample(10) In [19]: y=[] for n, group in g: y.append(n) x.append(sum(group['Turnover'])/sum(group['Total Traded Quantity'])) print('\t\tList of VWAP of each group of month and Year\t\n\n',x) List of VWAP of each group of month and Year [938.8828664708917, 1002.8435482070836, 978.448699257408, 949.8413467831679, 893.8529829569709, 846. 0772160043015, 861.7780411959719, 917.6575062053062, 894.6386215974317, 817.4930415398811, 866.855816 3056941, 915.2418851408958] In [20]: l=pd.DataFrame() 1 ['Month'] = y l['VWAP']=x In [21]: print('\n\t\t Monthly Value Weighted Average Price ') Monthly Value Weighted Average Price Out[21]: **VWAP** Month 0 1 938.882866 2 1002.843548 1 978.448699 4 949.841347 3 5 893.852983 6 846.077216 5 7 861.778041 6 8 917.657506 7 9 894.638622 8 9 10 817.493042 10 11 866.855816 12 915.241885 11 Problem [1.5 (a)]: Function to calculate the Average price of last N **Days** In [22]: **def** f(d,n): df=d.iloc[d.shape[0]-n:d.shape[0]+1,:] return df n=int(input('enter last n days : ')) ndf=f(data,n) # mean of the close price of last n days ndf['Close Price'].mean() enter last n days: 90 Out[22]: 1027.5994444444448 Problem [1.5 (b)]: Function to calculate Profit / Loss % for last n days In [23]: import statistics # to get only last n days data we will call f function d=f(d,n)lcp=list(d['Close Price']) l=len(lcp) lper=[] **if** 1%2!=0:1=1-1 for i in range (1-1): lper.append(((lcp[i+1]-lcp[i])/lcp[i])*100)#print(lper) if(statistics.mean(lper)>0):print('Net Profit Obtained : ' ,statistics.mean(lper)) else:print('Net Loss Obtained :' , statistics.mean(lper)) Function to create Days from months and year In [24]: def createdays(s): s=s.split() days=int(s[0]) if(s[1] == 'year' or s[1] == 'years'): days=days*365 elif(s[1] == 'month' or s[1] == 'months'):days=days*30 **elif**(s[1] == 'week' **or** s[1] == 'weeks'): days=days*7 return days In [25]: print('Enter the period') s=input() days=createdays(s) profloss (data, days) Enter the period 3 months Net Profit Obtained: 0.09716290845092859 Problem [1.6]: Adding Feature Day_Perc_Change In [26]: # Here we make the data with month and year as new dataframe ndata ndata=d.copy() ndata['Day_Perc_change']=ndata['Close Price'].pct_change() ndata # giving nan value 0.000 value ndata['Day_Perc_change'][0]=0 ndata['Day_Perc_change'] = ndata['Day_Perc_change'] *100 ndata.head() Out[26]: % D Total Prev Open High Low Last Close Average No. of Deliverable Qt Symbol Series Date Traded Turnover **Price** Close Price Price **Price** Price Price **Trades** Qty Trade Quantity Q 2017-0 LALPATHLAB 891.15 895.0 914.15 881.00 912.00 900.60 889.35 257655 2.291453e+08 15451 174775 67. 05-15 2017-1 LALPATHLAB 910.0 925.00 895.05 909.40 910.95 900.60 914.57 104896 9.593430e+07 13001 75813 72.: 05-16 910.95 913.0 925.00 909.00 912.05 911.70 2 LALPATHLAB 917.19 72460 6.645960e+07 2802 53829 74.: 3 LALPATHLAB 908.0 919.35 903.05 906.00 909.75 911.70 914.12 37707 3.446889e+07 2731 24836 65. 05-18 2017-4 LALPATHLAB 909.75 917.0 917.00 905.80 910.00 910.25 910.61 81082 7.383375e+07 4430 69926 86. In [27]: print('Enter the period') s=input() days=createdays(s) # to get data for last n days lndata=f(ndata,days) lndata Enter the period 3 months Out[27]: Total Prev Open High No. of Delivera Low Last Close Average Symbol Series **Traded** Date Turnover Close Price **Price** Price **Price** Price **Price Trades** Quantity 2018-404 LALPATHLAB EQ 890.95 909.95 910.30 906.08 908.50 911.25 922.00 341688 3.095975e+08 20516 286 12-31 2019-LALPATHLAB EQ 910.30 909.95 157587 405 928.00 902.45 925.00 924.80 912.79 1.438439e+08 4646 112 01-01 2019-EQ 915.20 406 LALPATHLAB 924.80 924.00 924.00 900.00 916.00 910.93 94182 8.579362e+07 4481 50 01-02 2019-LALPATHLAB EQ 407 915.20 909.70 918.00 897.10 910.00 913.05 907.57 71756 6.512356e+07 3283 30 01-03 2019-940.10 937.58 408 LALPATHLAB 913.05 908.00 944.85 908.00 935.45 156645 1.468668e+08 7745 102 01-04 2019-490 LALPATHLAB 1024.95 1015.70 1031.95 1006.90 1013.00 1013.10 1020.66 20113 2.052851e+07 3628 11 2019-491 LALPATHLAB EQ 1013.10 1014.00 1019.85 1001.90 1002.00 1006.10 1010.90 18048 1.824473e+07 4175 11 05-08 2019-982.90 987.36 5.977355e+07 492 LALPATHLAB EQ 1006.10 1014.00 1014.00 977.70 979.90 60539 8169 37 05-09 2019-963.60 980.65 493 LALPATHLAB EQ 982.90 989.90 994.30 980.00 980.71 30568 2.997840e+07 4618 19 05-10 2019-494 LALPATHLAB 975.00 EQ 980.65 970.80 996.00 960.00 976.25 974.21 37187 3.622810e+07 6798 23 05-13 90 rows × 18 columns In [28]: mean per=Indata['Day Perc change'].mean() if (mean_per>0):print('Net Profit Obtained : ' , mean_per) else:print('Net Loss Obtained : ' , mean per) Net Profit Obtained: 0.09828475150935694 **Problem [1.7]: Adding Feature Trend** In [29]: trend=[] d=list(ndata['Day_Perc_change']) for i in d: if(i>-0.5 and i<=0.5):ans='Slight or No change'elif(i>0.5 and i<=1):ans='Slight positive'</pre> elif(i>-1 and i<=-0.5):ans='Slight negative'</pre> elif(i>1 and i<=3):ans='positive'</pre> elif(i>-3 and i<=-1):ans='negative'</pre> elif(i>3 and i<=7):ans='Among top gainers'</pre> elif(i>-7 and i<=-3):ans='Among top losers'</pre> elif(i>7):ans='Bull run' elif(i<-7):ans='Bear drop'</pre> trend.append(ans) ndata['Trend']=trend In [30]: ndata Out[30]: Total Open High Prev Low Last Close Average No. of Deliveral Symbol Series Date Traded Turnover **Trades** Close **Price Price Price** Price Price Price Quantity 0 LALPATHLAB 881.00 891.15 895.0 914.15 912.00 900.60 889.35 257655 2.291453e+08 15451 1747 2017-1 LALPATHLAB EQ 900.60 910.0 925.00 895.05 909.40 910.95 104896 13001 914.57 9.593430e+07 758 05-16 2017-2 LALPATHLAB EQ 910.95 913.0 925.00 909.00 912.05 911.70 917.19 72460 6.645960e+07 2802 538 05-17 2017-3 LALPATHLAB 911.70 908.0 919.35 903.05 906.00 909.75 914.12 37707 3.446889e+07 2731 248 05-18 2017-910.61 4 LALPATHLAB EQ 910.00 910.25 909.75 917.0 917.00 905.80 81082 7.383375e+07 4430 699 05-19 2019-1006.90 1013.10 LALPATHLAB EQ 1024.95 1015.7 1031.95 1013.00 1020.66 20113 2.052851e+07 3628 490 115 05-07 1002.00 1006.10 LALPATHLAB 1014.0 1019.85 1001.90 1010.90 491 1013.10 18048 1.824473e+07 4175 112 05-08 2019-LALPATHLAB EQ 1006.10 982.90 987.36 492 1014.0 1014.00 977.70 979.90 60539 5.977355e+07 8169 373 05-09 2019-493 LALPATHLAB EQ 982.90 980.00 980.65 4618 989.9 994.30 963.60 980.71 30568 2.997840e+07 190 05-10 2019-494 LALPATHLAB 960.00 EQ 980.65 970.8 996.00 975.00 976.25 974.21 37187 3.622810e+07 6798 237 05-13 494 rows × 19 columns In [31]: ndata['Trend'].unique() Out[31]: array(['Slight or No change', 'positive', 'negative', 'Slight positive', 'Slight negative', 'Among top gainers', 'Among top losers', 'Bull run'], dtype=object) In [32]: df=ndata.copy() In [33]: df Out[33]: Total Open High Close Average No. of Deliveral Prev Low Last **Traded Turnover** Symbol Series Date Price **Trades** Close Price **Price Price Price Price** Quantity 2017-912.00 889.35 LALPATHLAB EQ 891.15 895.0 914.15 881.00 900.60 257655 2.291453e+08 15451 1747 05-15 2017-1 LALPATHLAB EQ 900.60 910.0 925.00 909.40 910.95 895.05 914.57 104896 9.593430e+07 13001 758 05-16 2017-2 LALPATHLAB 910.95 912.05 911.70 913.0 925.00 909.00 917.19 72460 6.645960e+07 2802 538 05-17 2017-3 LALPATHLAB EQ 911.70 919.35 903.05 906.00 909.75 908.0 914.12 37707 3.446889e+07 2731 248 2017-917.00 905.80 81082 7.383375e+07 LALPATHLAB EQ 909.75 917.0 910.00 910.25 910.61 4430 699 05-19 2019-LALPATHLAB EQ 1015.7 1031.95 1006.90 1013.00 1013.10 490 1024.95 1020.66 20113 2.052851e+07 3628 115 05-07 2019-1006.10 LALPATHLAB EQ 1013.10 1014.0 1019.85 1001.90 1002.00 491 1010.90 18048 1.824473e+07 4175 112 05-08 2019-492 LALPATHLAB EQ 60539 5.977355e+07 1006.10 1014.0 1014.00 977.70 979.90 982.90 987.36 8169 373 2019-05-10 2019-**494** LALPATHLAB 980.65 996.00 960.00 6798 237 970.8 975.00 976.25 974.21 37187 3.622810e+07 05-13 494 rows × 19 columns In [34]: gd=df.groupby('Trend') gd.first() In [35]: Out[35]: Total Prev Open High Low Last Close Average No. of Deliverable Symbol Series Date Traded Turnover Close Price **Price** Price Price Price **Trades** Price Qty Trend **Among** 2017-LALPATHLAB 799.85 790.00 847.75 783.80 833.4 834.80 809.38 177028 1.432833e+08 16859 79503 top gainers Among 2017-LALPATHLAB 834.80 834.00 834.00 780.00 780.1 781.95 783.30 2047290 1.603652e+09 23920 1226065 top 06-30 losers 2018-LALPATHLAB EQ 807.10 828.00 932.50 825.00 892.0 887.70 861.50 1136557 9.791453e+08 Bull run 42803 545124 05-15 Slight 2017-898.25 903.45 907.90 880.00 888.0 893.70 LALPATHLAB 898.82 75168 6.756225e+07 7428 57741 negative 05-26 Slight or 2017-889.35 LALPATHLAB 891.15 895.00 914.15 881.00 912.0 900.60 257655 2.291453e+08 15451 174775 No 05-15 change Slight 2017-886.90 886.90 903.05 883.80 893.0 894.70 895.74 **LALPATHLAB** 84241 7.545772e+07 67593 positive 2017negative LALPATHLAB 909.85 913.95 913.95 875.00 890.0 886.90 899.58 201776 1.815139e+08 7913 167269 2017positive LALPATHLAB 900.60 910.00 925.00 895.05 909.4 910.95 914.57 104896 9.593430e+07 13001 75813 Problem [1.8]: Mean and Median of Total Traded Quantity of each Trend In [36]: for n, group in gd: print(' Mean : ' , group['Total Traded Quantity'].mean()) print(' Median : ' , group['Total Traded Quantity'].median()) Among top gainers : 186722.96 Mean Median : 110923.0 Among top losers Mean : 258606.0 Median : 63332.0 Bull run Mean : 722199.0 Median : 722199.0 Slight negative : 63297.0 Mean Median: 41099.0 Slight or No change : 67316.4444444444 Mean Median : 40710.0 Slight positive Mean : 87609.45238095238 Median: 45531.0 negative : 83712.23762376238 Median : 53858.0 positive Mean : 82095.8493150685 Median : 55105.0 Problem [1.9]: Saving Dataframe to csv file In [37]: | ndata.to csv(r"C:\Users\Suprateek Halsana\Documents\Python Scripts\Aspiring Mind Internship\LALPATHLAB [INDUSTRY - CHOSEN]\week2.csv",index=False)