Name : Dhaval Gogri

Course : Advanced Application Programming

Quest No : Quest 5

In [381]:

```
import pandas as pd
import matplotlib.pyplot as plt
```

Part A - States and area codes

In [373]:

```
# Reading the csv file and storing it in part1 variable
part1 = pd.read_csv("/Users/dhavalgogri/Documents/Dhaval old computer/000 SMU study materials/Fall2018/AdvancedApplProg
```

In [376]:

```
# Removing the rows where "Wasington,DC" has been included as a State
part1 = part1[part1.State != 'Washington,DC']
```

In [375]:

```
# Counting the frequency of each state with respect to its state code
countFrequency = part1.set_index(["State", "State code"]).count(level="State")

# Only keeping the States which have only 1 state code. Checking Area Code condition with 1
countFrequencyWithOnlyOneAreaCode = countFrequency[countFrequency['Area code'] == 1]

# Printing the States which have only 1 State Code
print(countFrequencyWithOnlyOneAreaCode)
```

Out[375]:

```
                Area code
          State
-------------------------
         Alaska         1
       Delaware         1
         Hawaii         1
          Idaho         1
          Maine         1
        Montana         1
  New Hampshire         1
     New Mexico         1
   North Dakota         1
   Rhode Island         1
   South Dakota         1
        Vermont         1
  West Virginia         1
        Wyoming         1
```

In [ ]:

In [ ]:

PART B - Zipcodes and States

In [377]:

```
# Reading the csv file and storing it in part2 variable
part2 = pd.read_csv("/Users/dhavalgogri/Documents/Dhaval old computer/000 SMU study materials/Fall2018/AdvancedApplProg
```

In [363]:

```
# Counting the frequency of each state with respect to its zip codes
countFrequency2 = part2.set_index(["state", "city"]).count(level="state")

# Sorting the state with respect to the number of zip codes it has
countFrequency2 = countFrequency2.sort_values('zip_code', ascending = False)

# Printing the state and zip code information below
countFrequency2['zip_code']
```

Out[363]:

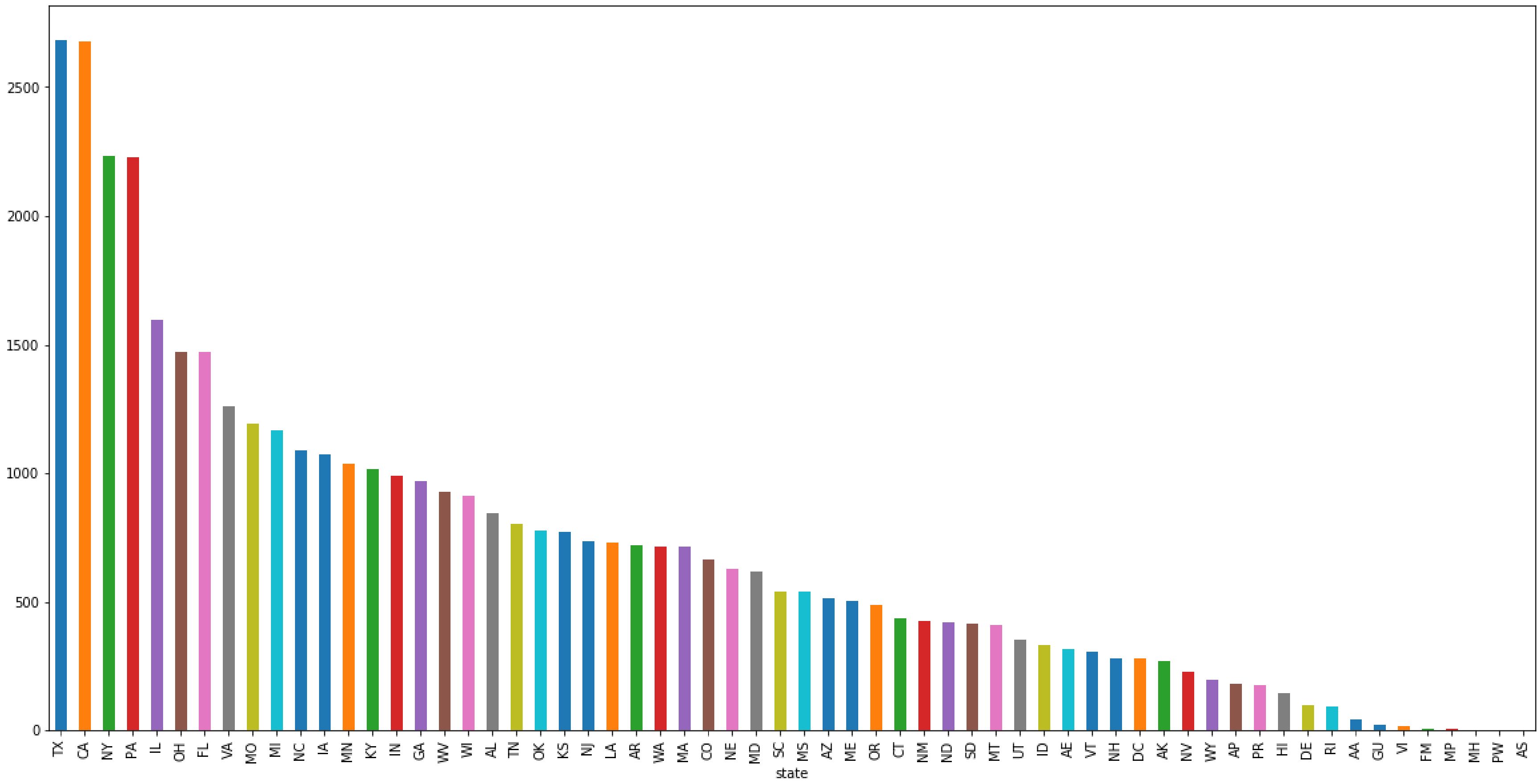
```
state
TX    2682
CA    2678
NY    2233
PA    2226
IL    1596
OH    1472
FL    1470
VA    1257
MO    1192
MI    1167
NC    1087
IA    1071
MN    1036
KY    1016
IN     991
GA     967
WV     930
WI     913
AL     847
TN     805
OK     778
KS     772
NJ     737
LA     731
AR     720
WA     716
MA     713
CO     662
NE     628
MD     619
      ... 
AZ     515
ME     505
OR     485
CT     436
NM     427
ND     418
SD     416
```

```
MT     411
UT     352
ID     332
AE     317
VT     308
NH     281
DC     278
AK     269
NV     227
WY     197
AP     181
PR     177
HI     143
DE      98
RI      91
AA      41
GU      21
VI      16
FM       4
MP       3
MH       2
PW       1
AS       1
Name: zip_code, Length: 62, dtype: int64
```

```
In [ ]:
```

In [364]:

```
# Graph to plot the State vs Total Zip Code for that state
my_plot = countFrequency2['zip_code'].plot(kind='bar', width = 0.5, figsize = (20,10))
```



In [ ]:

```
```

In [ ]:

```
```

PART C - Films

In [ ]:

```
```

In [378]:

```
# Reading the csv file and storing it in part3 variable
part3 = pd.read_csv("/Users/dhavalgogri/Documents/Dhaval old computer/000 SMU study materials/Fall2018/AdvancedApplProg

# Removing unwanted tables from the dataframe
del part3['Year']
del part3['Length']
del part3['Title']
del part3['Actress']
del part3['Director']
del part3['Popularity']
del part3['Awards']
del part3['*Image,,,,,']
```

In [366]:

```
# replacing the value "Westerns" by "Western" in the "Subject" column
part3.loc[part3['Subject'] == 'Westerns'] = 'Western'
```

In [367]:

```
# group subject and actor and finding their total count and storing it in the "Count" column
part3['Count']=part3.groupby(['Subject','Actor']).Subject.transform('size')

# grouping the same and only putting Max values as true if the actor has the most films for
# that subject
part3['Max']=part3.groupby(['Subject'])['Count'].transform('max')==part3['Count']

# Only keeping the rows which have the Max values as true as it is the highest combination
# for that subject
groupBySubject_Actor = part3[part3['Max'] == True]

# Removing all the values where the Max Subject and Actor count is 1.
groupBySubject_Actor = groupBySubject_Actor[groupBySubject_Actor['Count'] > 1]

# Grouping all the remaining data into subject, actor and count
groupBySubject_Actor_Count = groupBySubject_Actor.groupby(['Subject','Actor','Count'])['Actor','Count']

# storing the final result in the finalAnswer dataFrame
finalAnswer = pd.DataFrame(groupBySubject_Actor_Count.size())

# Printing the dataframe
finalAnswer
```

Out[367]:

```
                                             0
        Subject             Actor  Count
         Action     Connery, Sean   15.0    15
         Comedy     Sellers, Peter  22.0    22
          Drama    Brando, Marlon   17.0    17
         Horror     Ford, Wallace    3.0     3
        Mystery     Connery, Sean    3.0     3
                  Douglas, Michael   3.0     3
                        Grant, Cary  3.0     3
Science Fiction       Hamill, Mark   3.0     3
            War        Wayne, John  10.0    10
        Western        Wayne, John  46.0    46
```

In [ ]:

PART D - Stock Trading Algorithm with Moving Averages (MA)

In [368]:

```
# Reading the csv file and storing it in part4 variable
part4 = pd.read_csv('/Users/dhavalgogri/Documents/Dhaval old computer/000 SMU study materials/Fall2018/AdvancedApplProg

# Sorting data with date. Lowest date first
part4 = part4.sort_values(by='Date', ascending = True)
```

In [369]:

```
# Taking the moving average of the stock price and storing it in "MA100" column
part4['MA100'] = part4['SPY'].rolling(100).mean()
```

In [379]:

```
# Dropping all rows which have the value "NAN"
part4 = part4.dropna(axis=0)
```

Buy and Sell Stock based on Moving Average

In [371]:

```
# Buy and Sell stocks based on moving avergaes

initialCashInHand = 1000.0
cashInHand = 1000.0
stocksInHand = 0.0
buyCount = 0
sellCount = 0

# interate the data for 18 years of stock
for index, row in part4.iterrows():
    # Check when the value of the stock is greater than moving average
    # then we buy the stocks. We also check if we have cash in hand to buy
    # the stocks
    if (cashInHand > 0 and row['SPY'] > row['MA100']) :
        stocksInHand = cashInHand/row['SPY']
        cashInHand = 0
        buyCount = buyCount + 1
    # Check when the value of the stock is less than moving average
    # then we sell the stocks. We also check if we have stocks to sell
    elif (stocksInHand > 0 and row['SPY'] < row['MA100']) :
        cashInHand = stocksInHand * row['SPY']
        stocksInHand = 0
        sellCount = sellCount + 1
    # We sell all the available stocks in hand on the last day
    elif (index == 0 and stocksInHand > 0) :
        cashInHand = stocksInHand * row['SPY']
        stocksInHand = 0
        sellCount = sellCount + 1



# Calculating the percentage profit/loss
percentageIncrease = ((cashInHand - initialCashInHand)/initialCashInHand)*100
print("Stocks Buy Count in Moving Average Strategy = ", buyCount)
print("Stocks Sell Count Hand in Moving Average Strategy = ", sellCount)
print("Cash In Hand in Moving Average Strategy = $", cashInHand)
print("Percentage Increase in Moving Average Strategy = ", percentageIncrease)
```

```
Stocks Buy Count in Moving Average Strategy =  107
Stocks Sell Count Hand in Moving Average Strategy =  107
Cash In Hand in Moving Average Strategy = $ 2169.0758636030896
Percentage Increase in Moving Average Strategy =  116.90758636030895
```

In [ ]:

In [ ]:

Buy And Hold Strategy

In [372]:

```
# Buy and Hold Strategy

initialCashInHandDuringBuyAndHold = 1000.0
cashInHandDuringBuyAndHold = 1000.0
stocksInHandDuringBuyAndHold = 0.0
flag = 0

# Buy stocks at the first day of moving average
stocksInHandDuringBuyAndHold = cashInHandDuringBuyAndHold/part4.loc[len(part4)-1,'SPY']
cashInHandDuringBuyAndHold = 0

# Sell stocks at the last day of moving average
cashInHandDuringBuyAndHold = stocksInHandDuringBuyAndHold * part4.loc[0,'SPY']

# Calculating the percentage profit/loss
percentageIncreaseBuyAndHold = ((cashInHandDuringBuyAndHold - initialCashInHandDuringBuyAndHold)/initialCashInHandDurin
print("Cash In Hand in Buy and Hold Strategy = $", cashInHandDuringBuyAndHold)
print("Percentage Increase in Buy and Hold Strategy = ", percentageIncreaseBuyAndHold)
```

```
Cash In Hand in Buy and Hold Strategy = $ 2895.1340444047846
Percentage Increase in Buy and Hold Strategy =  189.51340444047847
```

In [ ]:

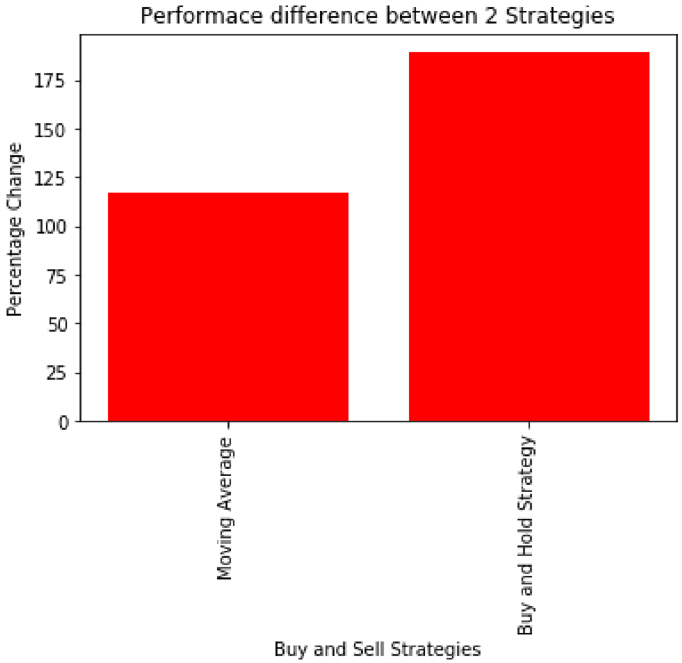
## Comaprision between Moving Average VS Buy and Hold

In [388]:

```
x = ['Moving Average', 'Buy and Hold Strategy']
yval  = [percentageIncrease, percentageIncreaseBuyAndHold]

x_pos = [0,1]

plt.bar(x_pos, yval, color='red')
plt.xticks(x_pos, x, rotation="vertical")
plt.xlabel("Buy and Sell Strategies")
plt.ylabel("Percentage Change")
plt.title("Performace difference between 2 Strategies")
plt.show()
```



From the above comparision we can see that the Buy and Hold Strategy was better for this Stock

In [ ]:

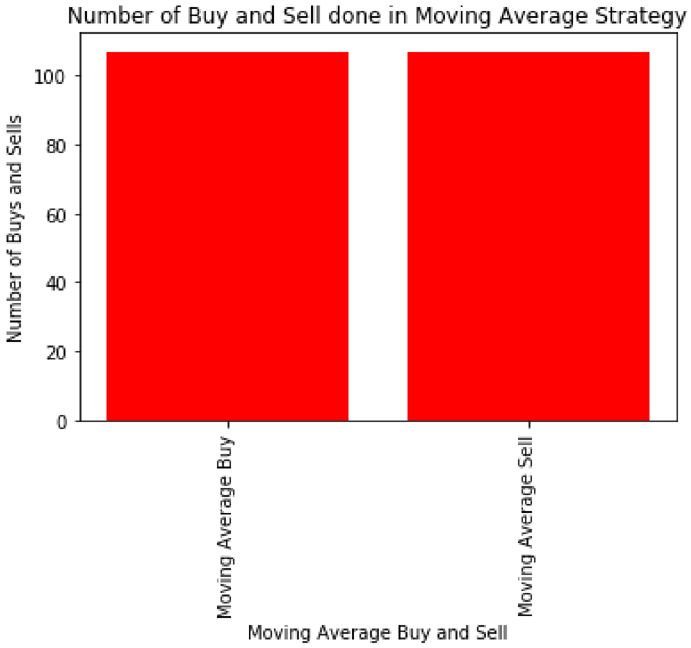
Comaprision between Moving Average BUY and SELL

In [391]:

```
x = ['Moving Average Buy', 'Moving Average Sell']
yval  = [buyCount, sellCount]

x_pos = [0,1]

plt.bar(x_pos, yval, color='red')
plt.xticks(x_pos, x, rotation="vertical")
plt.xlabel("Moving Average Buy and Sell")
plt.ylabel("Number of Buys and Sells")
plt.title("Number of Buy and Sell done in Moving Average Strategy")
plt.show()
```



From the above graph we can see that there were same number of buys and sell in Moving Average Strategy

In [ ]:

Present Slides Themes Help