ABInBev

<Ale-Go:Rhythm>:

- Candidate Name: Sonal Agrawal
- College Name: IIT Kanpur
- Email: sagarwal20@iitk.ac.in
- Phone No.: 9971514521

Chosen Problem statement: Stock Balancing

Problem Statement

Out of Stock at point of purchase due to lack of inventory

Excessive inventory at depot leading to expiry and wastage of product

Revenue loss to business

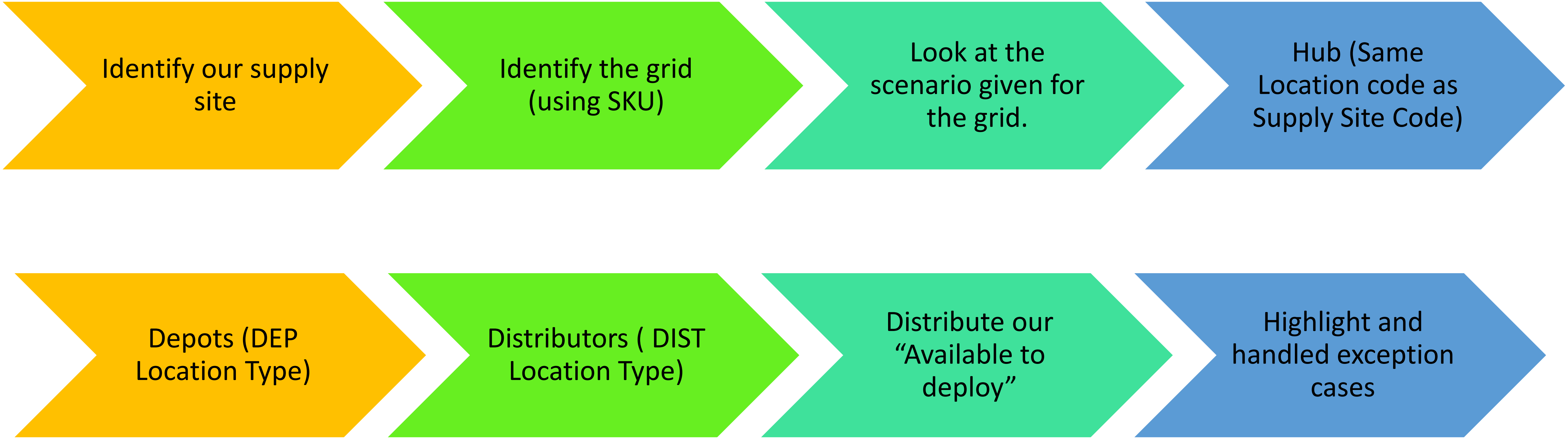
Requirement analysis

- We have to develop an optimization algorithm for stock distribution between breweries and depots so as to maintain a balanced inventory in all the depots in a region/country.

- We need to distribute the “Available to Deploy” among the depots/distributors according to the scenarios defined.

- The expected output is suggesting an optimal amount of stock that needs to go to each depot/distributor as the final output following the business rules.

Approach



Identify our supply site

Identify the grid (using SKU)

Look at the scenario given for the grid.

Hub (Same Location code as Supply Site Code)

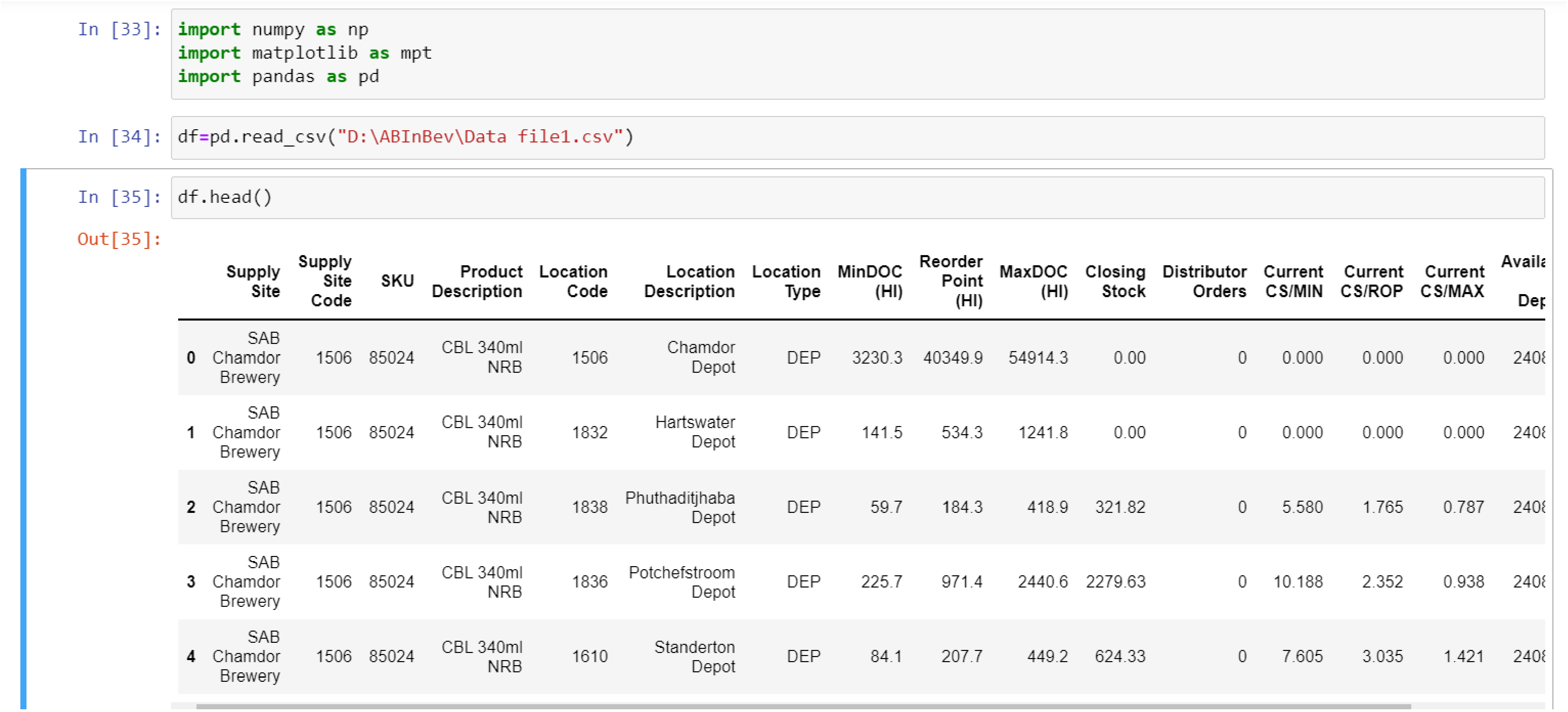
Depots (DEP Location Type)

Distributors (DIST Location Type)

Distribute our “Available to deploy”

Highlight and handled exception cases

Importing dataset using pandas in Jupyter Notebook



```
In [33]: import numpy as np
         import matplotlib as mpt
         import pandas as pd
```

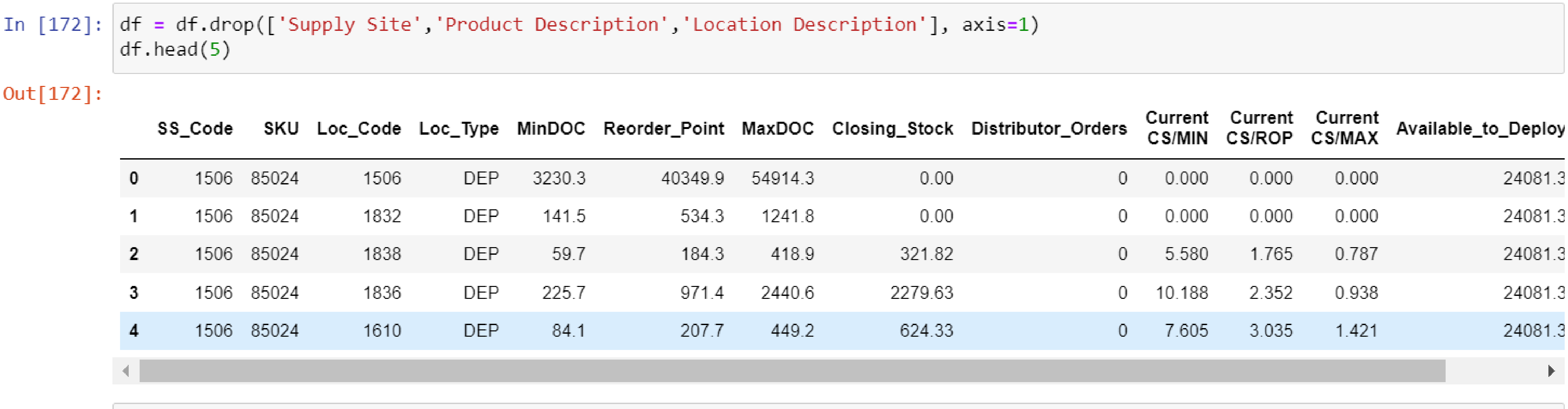
```
In [34]: df=pd.read_csv("D:\ABInBev\Data file1.csv")
```

```
In [35]: df.head()
```

Out[35]:

	Supply Site	Supply Site Code	SKU	Product Description	Location Code	Location Description	Location Type	MinDOC (HI)	Reorder Point (HI)	MaxDOC (HI)	Closing Stock	Distributor Orders	Current CS/MIN	Current CS/ROP	Current CS/MAX	Availa Dep
0	SAB Chamdor Brewery	1506	85024	CBL 340ml NRB	1506	Chamdor Depot	DEP	3230.3	40349.9	54914.3	0.00	0	0.000	0.000	0.000	2408
1	SAB Chamdor Brewery	1506	85024	CBL 340ml NRB	1832	Hartswater Depot	DEP	141.5	534.3	1241.8	0.00	0	0.000	0.000	0.000	2408
2	SAB Chamdor Brewery	1506	85024	CBL 340ml NRB	1838	Phuthaditjhaba Depot	DEP	59.7	184.3	418.9	321.82	0	5.580	1.765	0.787	2408
3	SAB Chamdor Brewery	1506	85024	CBL 340ml NRB	1836	Potchefstroom Depot	DEP	225.7	971.4	2440.6	2279.63	0	10.188	2.352	0.938	2408
4	SAB Chamdor Brewery	1506	85024	CBL 340ml NRB	1610	Standerton Depot	DEP	84.1	207.7	449.2	624.33	0	7.605	3.035	1.421	2408

Dropped unnecessary columns



```
In [172]: df = df.drop(['Supply Site','Product Description','Location Description'], axis=1)
          df.head(5)
```

Out[172]:

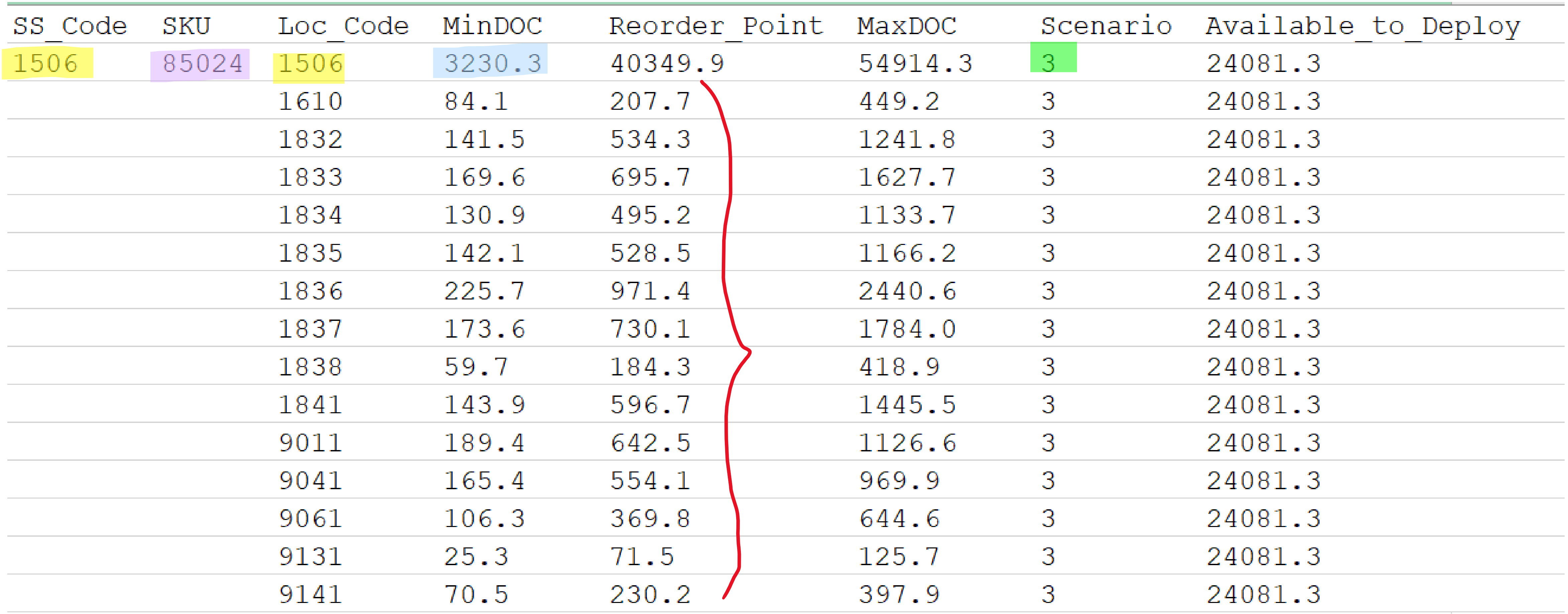
	SS_Code	SKU	Loc_Code	Loc_Type	MinDOC	Reorder_Point	MaxDOC	Closing_Stock	Distributor_Orders	Current CS/MIN	Current CS/ROP	Current CS/MAX	Available_to_Deploy
0	1506	85024	1506	DEP	3230.3	40349.9	54914.3	0.00	0	0.000	0.000	0.000	24081.3
1	1506	85024	1832	DEP	141.5	534.3	1241.8	0.00	0	0.000	0.000	0.000	24081.3
2	1506	85024	1838	DEP	59.7	184.3	418.9	321.82	0	5.580	1.765	0.787	24081.3
3	1506	85024	1836	DEP	225.7	971.4	2440.6	2279.63	0	10.188	2.352	0.938	24081.3
4	1506	85024	1610	DEP	84.1	207.7	449.2	624.33	0	7.605	3.035	1.421	24081.3

Handled Rows with NULL values

GRID 1

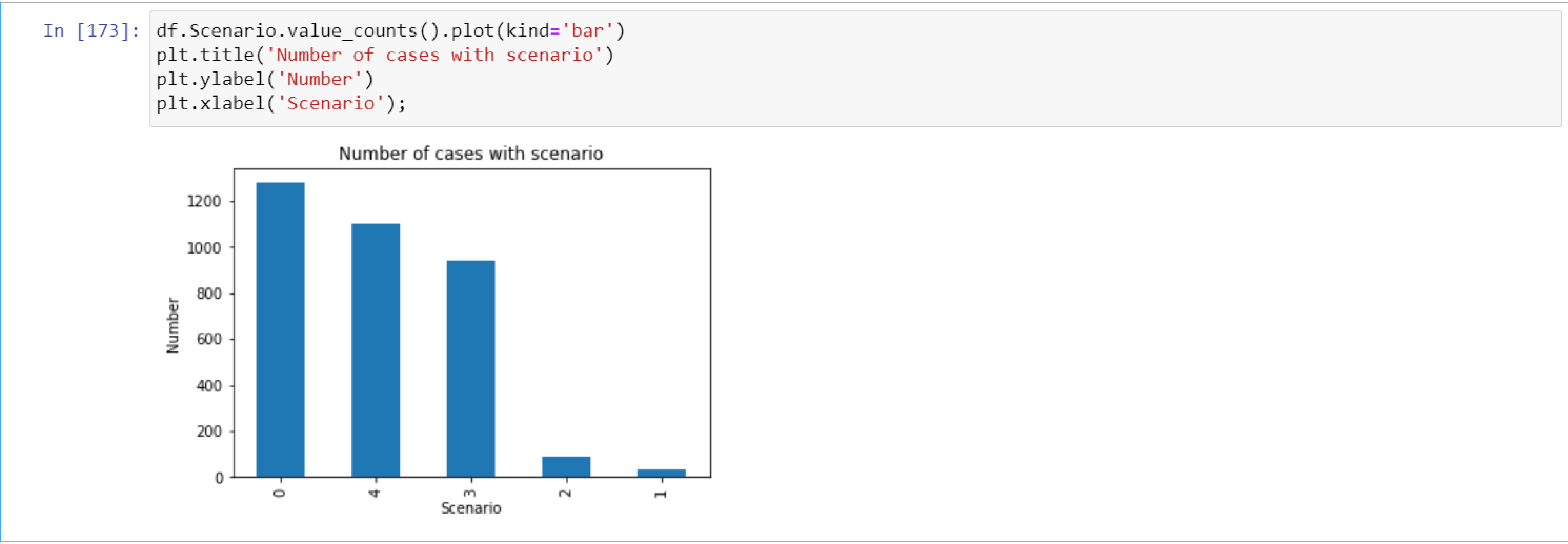
🟪 - SKU for GRID

🟨 - Hub for GRID Since
SS_Code = Loc_Code



SS_Code	SKU	Loc_Code	MinDOC	Reorder_Point	MaxDOC	Scenario	Available_to_Deploy
1506	85024	1506	3230.3	40349.9	54914.3	3	24081.3
		1610	84.1	207.7	449.2	3	24081.3
		1832	141.5	534.3	1241.8	3	24081.3
		1833	169.6	695.7	1627.7	3	24081.3
		1834	130.9	495.2	1133.7	3	24081.3
		1835	142.1	528.5	1166.2	3	24081.3
		1836	225.7	971.4	2440.6	3	24081.3
		1837	173.6	730.1	1784.0	3	24081.3
		1838	59.7	184.3	418.9	3	24081.3
		1841	143.9	596.7	1445.5	3	24081.3
		9011	189.4	642.5	1126.6	3	24081.3
		9041	165.4	554.1	969.9	3	24081.3
		9061	106.3	369.8	644.6	3	24081.3
		9131	25.3	71.5	125.7	3	24081.3
		9141	70.5	230.2	397.9	3	24081.3

Scenario Distribution



```
In [173]: df.Scenario.value_counts().plot(kind='bar')
          plt.title('Number of cases with scenario')
          plt.ylabel('Number')
          plt.xlabel('Scenario');
```

Understanding our business rules

1

DEP : MaxDOC

DIST : Distributor Orders

Hub : Available to Deploy - ∑ (MaxDOC (only for depot)+ Distributor Orders)

2

DIST : Distributor Orders

DEP + Hub : (ROP*remaining_amt/∑ROP)

Remaining_amt = Available to Deploy - ∑ Distributor Orders

3

Hub : MinDOC

DEP + DIST : (ROP*remaining_amt /∑ROP)

Remaining_amt = Available to Deploy – MinDOC for Hub

4

Hub + DEP + DIST (all sites in the grid) :(ROP*Available to Deploy /∑ROP)

```
#pseudo code for identifying grid
#Identifying the start and end for our grid using the SKU code
#skulist contains the SKU code for all the Products


start = 0
end = 0

container = []

for i in range(len(skulist)):
    if (skulist[start] == skulist[end]):
        end += 1
    else:
        container.append([start,end])
        start = end+1
        end = start

# [(1, 16), (17, 60), ...]
```

```
#pseudo code for scenario 3
#Container looks like this [(1, 16), (17, 60), ...]

n = len(container)

for i in range(n):
    batch_start = container[i][0]   #index for start of grid
    batch_end = container[i][1]    #index for end of grid

    hub = df[batch_start, :]  #we notice the first row in each grid corresponds to the hub
    Scnr =hub[Scenario]  #identifying our scenario

if Scnr==3:
    remaining_amt = hub[Available_to_Deploy] - hub[MinDOC]

    ratio_sum = 0
    for j in range(batch_end - batch_start):
        ratio_sum += df[batch_start+1+j][Render_Point]

    division_ratio = []

    for j in range(batch_end - batch_start):
        division_ratio.append(df[batch_start+1+j][] * remaining_amt/ratio_sum)
```

```
# Last step will be to check that Total stock at site = supplied stock + closing stock < MaxDOC for that particular site
```

Shortcomings

- Dropped columns that have null CS/MIN or CS/MAX or CS/ROP.
- Have to deal with scenario 0 cases.

Workings and progress an be checked out here:

Jupyter Notebook : *Exploratory Data Analysis + Groupings + pseudo codes*

Thank You ABInBev for this opportunity!