Drive Medical Data Assessment

In [1]:
```
import pandas as pd
```

In [2]:
```
pd.set_option('display.max_colwidth', None)
```

In [3]:
```
instructions = pd.read_excel("Excel Aptitude Assessment - 20-May-2021 (RD).xlsx",sheet_name = 0,skiprows=1)
```

In [4]:
```
instructions['Please refer to the "Open PO Data" tab to complete the tasks/questions below. You may provide the answ
ers in any format (and on any tab, including creating new tabs) you prefer.']
```

Out[4]:
```
0                                         1. How many *unique* materials are on open PO?
1                                      2. What is the total value of open POs for plant BP02?
2                            3. How many PO line items have a delivery date in August of 2021?
3    4. What percent of total open PO value is for items with an "A" indicator for ABC?
4                                    5. What is the average cost per unit of material 15528?
5                             6. Create a pivot table showing total open PO value by category
6                                                           7. Freeze panes on the top row
7                                                        8. Complete a VLOOKUP or XLOOKUP
8                                                         9. Complete a SUMIF calculation
9                                                  10. Complete a text-to-columns function
Name: Please refer to the "Open PO Data" tab to complete the tasks/questions below. You may provide the answers in an
y format (and on any tab, including creating new tabs) you prefer., dtype: object
```

In [5]:
```
df = pd.read_excel("Excel Aptitude Assessment - 20-May-2021 (RD).xlsx", sheet_name = 1)
```

In [6]:
```
df.head()
```

Out[6]:

	PO Number	Material	ABC	Category	Plant	Quantity	Value	Delivery Date
0	291114	10210-4ASM	A	Walkers, All Types	BP01	343	12691.00	2021-05-20
1	290718	10210-4ASM	A	Walkers, All Types	BP01	558	20646.00	2021-05-20
2	287832	10210-4ASM	A	Walkers, All Types	BP03	505	19578.85	2021-05-20
3	287489	RTL10320BKB	D	Canes & Crutches	BP01	20	104.00	2021-05-20
4	286517	10210-4ASM	A	Walkers, All Types	BP01	505	19578.85	2021-05-20

In [7]:
```
df.describe()
```

Out[7]:

	PO Number	Quantity	Value
count	11635.000000	11635.000000	11635.000000
mean	295593.810915	168.160808	7264.512486
std	7904.179130	273.447218	11111.755576
min	226764.000000	1.000000	0.000000
25%	291483.000000	22.000000	1118.600000
50%	296890.000000	70.000000	3268.320000
75%	301614.500000	184.000000	9734.000000
max	305710.000000	4500.000000	196946.400000

In [8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11635 entries, 0 to 11634
Data columns (total 8 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   PO Number      11635 non-null  int64
 1   Material       11635 non-null  object
 2   ABC            11635 non-null  object
 3   Category       11635 non-null  object
 4   Plant          11635 non-null  object
 5   Quantity       11635 non-null  int64
 6   Value          11635 non-null  float64
 7   Delivery Date  11635 non-null  datetime64[ns]
dtypes: datetime64[ns](1), float64(1), int64(2), object(4)
memory usage: 727.3+ KB
```

1. How many *unique* materials are on open PO?

- There are 11538 unique materials on open PO

In [9]:
```
len(df['PO Number'].unique())
```

Out[9]:
```
11538
```

In [10]:
```
print(f"There are {len(df['PO Number'].unique())} unique materials on open PO.")
```

```
There are 11538 unique materials on open PO.
```

2. What is the total value of open POs for plant BP02?

- There are 5928 different BP02's with a sum of $47,581,180.73

In [11]:
```
bp02 = df[df['Plant'] == 'BP02']
```

In [12]:
```
bp02
```

Out[12]:

	PO Number	Material	ABC	Category	Plant	Quantity	Value	Delivery Date
25	280214	10226-1	B	Walkers, All Types	BP02	1350	14350.5	2021-05-20
26	279756	369352	B	Bath Safety	BP02	4500	28485.0	2021-05-20
27	275167	MS391520	C	Patient Room	BP02	170	8964.1	2021-05-20
28	275168	MS391520	C	Patient Room	BP02	170	8964.1	2021-05-20
29	274670	12036	B	Bath Safety	BP02	386	19801.8	2021-05-20
...	...	...	...	...	...	...	...	...
11630	304222	13244	B	Patient Room	BP02	44	23515.8	2021-12-08
11631	304469	SFPRO317FS-21	D	Power	BP02	80	29420.0	2021-12-10
11632	304009	SFPRO417FS-12	C	Power	BP02	80	28348.0	2021-12-10
11633	305034	SFPRO417FS-21	C	Power	BP02	80	31308.0	2021-12-11
11634	289206	H19E008BK	D	Personal Care	BP02	800	7840.0	2022-02-21

5928 rows × 8 columns

In [13]:
```
len(bp02)
```

Out[13]:
```
5928
```

In [14]:
```
sum(bp02['Value'])
```

Out[14]:
```
47581180.72999994
```

In [15]:
```
print(f"There are {len(bp02)} different BP02's with a sum of ${sum(bp02['Value']):.2f}")
```

```
There are 5928 different BP02's with a sum of $47581180.73
```

3. How many PO line items have a delivery date in August of 2021?

- There are 2496 PO line items in Aug 2021

In [16]:
```
aug_2021 = df[(df['Delivery Date'].dt.month == 8) & (df['Delivery Date'].dt.year == 2021)]
```

In [17]:
```
aug_2021
```

Out[17]:

	PO Number	Material	ABC	Category	Plant	Quantity	Value	Delivery Date
8305	305379	15216P	A	Beds	BP03	48	9893.76	2021-08-01
8306	305257	STDS1094	C	Transport/Wheelchair	BP03	100	3569.00	2021-08-01
8307	304616	AP-K316DDA	D	Transport/Wheelchair	BP01	37	1843.71	2021-08-01
8308	304612	AP-K316DDA	D	Transport/Wheelchair	BP02	63	3139.29	2021-08-01
8309	304611	AP-K320DDA	D	Transport/Wheelchair	BP02	39	1943.37	2021-08-01
...	...	...	...	...	...	...	...	...
10796	302780	15216P	A	Beds	BP01	128	31037.44	2021-08-31
10797	301401	10257BL-1	A	Rollators	BP01	199	4545.16	2021-08-31
10798	301364	11148-1	A	Commodes	BP03	253	2818.42	2021-08-31
10799	299085	15211P	B	Beds	BP03	128	13314.56	2021-08-31
10800	280899	BLS16FBD-ELR	C	Transport/Wheelchair	BP02	21	1051.26	2021-08-31

2496 rows × 8 columns

In [18]:
```
len(aug_2021)
```

Out[18]:
```
2496
```

In [19]:
```
print(f"There are {len(aug_2021)} PO line items in Aug 2021.")
```

```
There are 2496 PO line items in Aug 2021.
```

4. What percent of total open PO value is for items with an "A" indicator for ABC?

- Of the total open PO Values for ABC, 21.76% comes from A.

In [20]:
```
df.groupby('ABC').count()
```

Out[20]:

	PO Number	Material	Category	Plant	Quantity	Value	Delivery Date
ABC							
A	2532	2532	2532	2532	2532	2532	2532
B	3673	3673	3673	3673	3673	3673	3673
C	3317	3317	3317	3317	3317	3317	3317
D	2094	2094	2094	2094	2094	2094	2094
S	19	19	19	19	19	19	19

In [21]:
```
df['ABC'].value_counts(normalize = True)
```

Out[21]:
```
B    0.315685
C    0.285088
A    0.217619
D    0.179974
S    0.001633
Name: ABC, dtype: float64
```

In [22]:
```
item_a = df['ABC'].value_counts(normalize = True)['A'] * 100
```

In [23]:
```
print(f'Of the total open PO Values for ABC, {item_a:.2f}% comes from A.')
```

```
Of the total open PO Values for ABC, 21.76% comes from A.
```

5. What is the average cost per unit of material 15528?

- The average cost per unit for material 15528 is $85.07.

In [24]:
```
material_15528 = df[df['Material'] == '15528']
```

In [25]:
```
material_15528
```

Out[25]:

	PO Number	Material	ABC	Category	Plant	Quantity	Value	Delivery Date
166	283927	15528	A	Beds	BP03	134	11353.82	2021-05-21
167	283934	15528	A	Beds	BP03	23	1948.79	2021-05-21
564	283174	15528	A	Beds	BP02	157	13302.61	2021-05-26
652	283929	15528	A	Beds	BP02	134	11353.82	2021-05-27
653	283933	15528	A	Beds	BP02	23	1948.79	2021-05-27
...	...	...	...	...	...	...	...	...
10600	283188	15528	A	Beds	BP01	157	13302.61	2021-08-26
10616	283898	15528	A	Beds	BP03	157	13302.61	2021-08-27
10788	285594	15528	A	Beds	BP02	157	13302.61	2021-08-30
10991	281160	15528	A	Beds	BP01	157	13302.61	2021-09-05
11094	283879	15528	A	Beds	BP01	157	13302.61	2021-09-07

123 rows × 8 columns

In [26]:
```
avg_15528 = (material_15528['Value'] / material_15528['Quantity']).mean()
```

In [27]:
```
print(f'The average cost per unit for material 15528 is ${avg_15528:.2f}.')
```

```
The average cost per unit for material 15528 is $85.07.
```

6. Create a pivot table showing total open PO value by category

In [28]:
```
df.groupby('Category').sum()['Value']
```

Out[28]:
```
Category
Bath Safety               6650143.04
Beds                     18443524.04
Canes & Crutches          2130865.84
Commodes                  4029057.04
Components                  27323.00
Electrotherapy             162554.40
Inspired                    77658.41
Patient Room              6780782.64
Personal Care              938806.38
Power                     5400788.70
Pressure Prevention       4524723.05
Respiratory               4377304.52
Rollators                 8128698.98
Sleep                      209442.20
Transport/Wheelchair     16804197.36
Walkers, All Types        5493727.45
Wenzelite                  343005.73
Name: Value, dtype: float64
```

7. Freeze panes on the top row

8. Complete a VLOOKUP or XLOOKUP

- Hypothetically if we had 2 datasets, 1 with just PO NUMBER and Quantity and the other with PO NUMBER and value we can merge these two sets by using the PO NUMBER as the matching key.

In [29]:
```
po_value = df.groupby("PO Number").sum().reset_index()[['PO Number','Value']]
```

In [30]:
```
po_quantity = df.groupby("PO Number").sum().reset_index()[['PO Number','Quantity']]
```

In [31]:
```
po_value.head()
```

Out[31]:

	PO Number	Value
0	226764	16184.70
1	228190	5446.28
2	236423	2523.06
3	236611	19133.52
4	237686	150.00

In [32]:
```
po_quantity.head()
```

Out[32]:

	PO Number	Quantity
0	226764	630
1	228190	212
2	236423	107
3	236611	588
4	237686	1

In [33]:
```
merged_po = po_quantity.merge(po_value, on = "PO Number")
```

In [34]:
```
merged_po
```

Out[34]:

	PO Number	Quantity	Value
0	226764	630	16184.70
1	228190	212	5446.28
2	236423	107	2523.06
3	236611	588	19133.52
4	237686	1	150.00
...	...	...	...
11533	305706	658	7330.12
11534	305707	602	6706.28
11535	305708	4	1568.00
11536	305709	107	13935.68
11537	305710	13	634.92

11538 rows × 3 columns

In [35]:
```
merged_po.equals(df.groupby("PO Number").sum().reset_index())
```

Out[35]:
```
True
```

9. Complete a SUMIF Calculation

- if they come from A and if they are beds, what is the total quantity/value per Plant

In [36]:
```
df[(df['ABC'] == 'A') & (df['Category'] == 'Beds')].groupby('Plant').sum().iloc[:,1:]
```

Out[36]:

	Quantity	Value
Plant		
BP01	75791	5939673.55
BP02	58452	6888851.84
BP03	32744	2430947.79

10. Complete a text-to-columns function

- I will divide the Material column and separate it by "-"

In [37]:
```
material_to_column = df['Material'].str.split('-',expand = True)
```

In [38]:
```
material_to_column = material_to_column.join(df['Material'])
```

In [39]:
```
material_to_column.columns = ['code1','code2','code3','code4','code5','full_material_code']
```

In [40]:
```
material_to_column
```

Out[40]:

	code1	code2	code3	code4	code5	full_material_code
0	10210	4ASM	None	None	None	10210-4ASM
1	10210	4ASM	None	None	None	10210-4ASM
2	10210	4ASM	None	None	None	10210-4ASM
3	RTL10320BKB	None	None	None	None	RTL10320BKB
4	10210	4ASM	None	None	None	10210-4ASM
...	...	...	...	...	...	...
11630	13244	None	None	None	None	13244
11631	SFPRO317FS	21	None	None	None	SFPRO317FS-21
11632	SFPRO417FS	12	None	None	None	SFPRO417FS-12
11633	SFPRO417FS	21	None	None	None	SFPRO417FS-21
11634	H19E008BK	None	None	None	None	H19E008BK

11635 rows × 6 columns