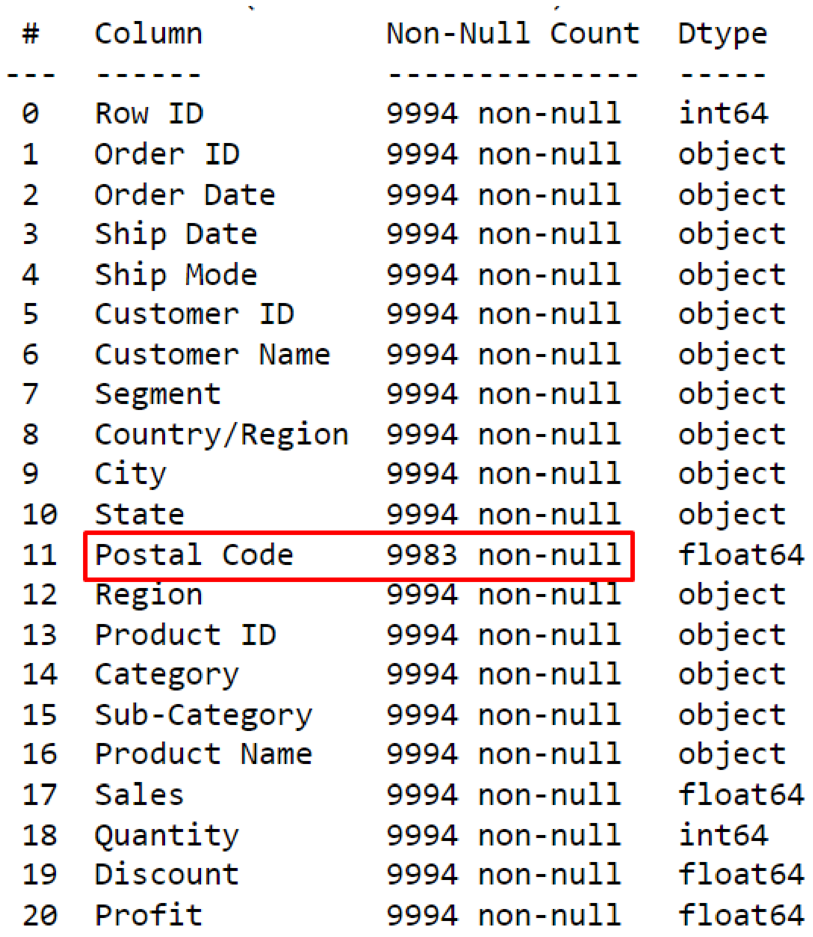
1. Analysis

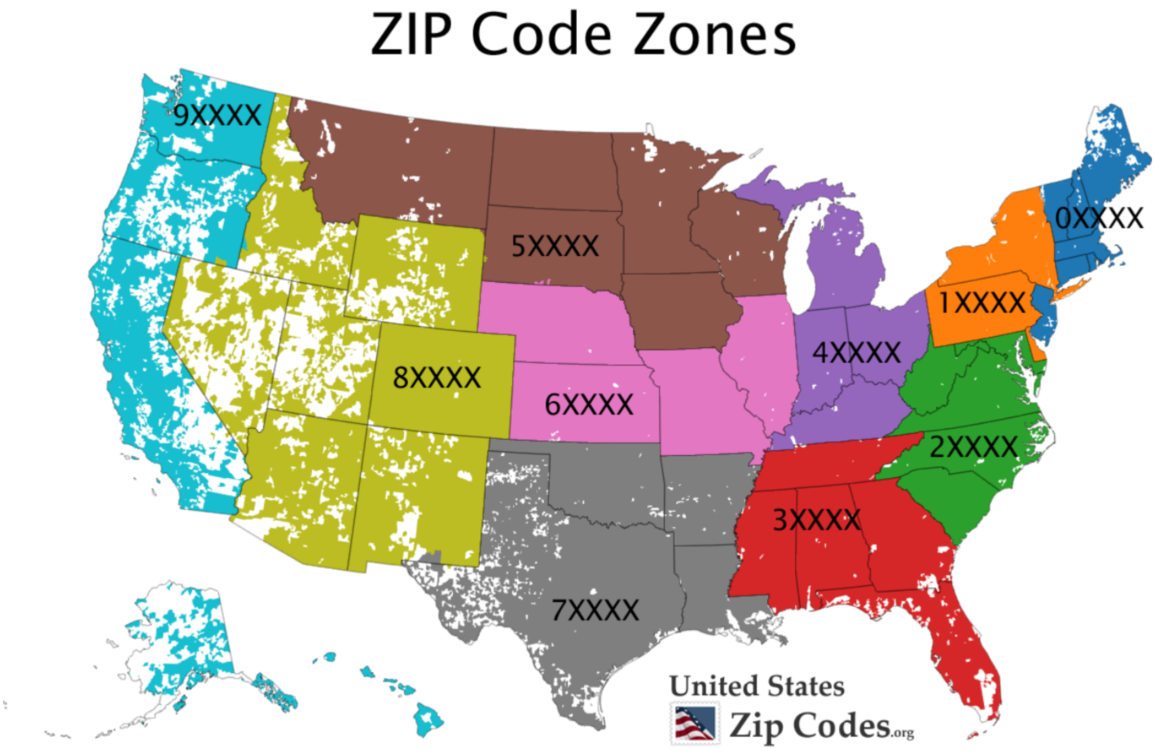
Postal Code

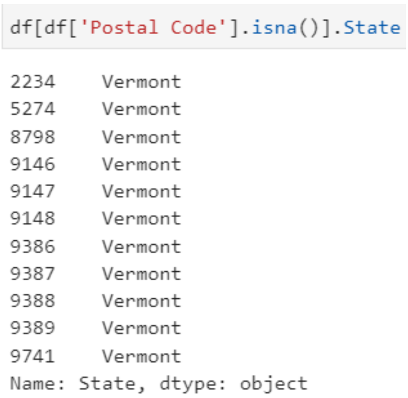
- We detected 11 null values in this column. We have to analyze the null rows and decide if they can be filled instead of dropping them:



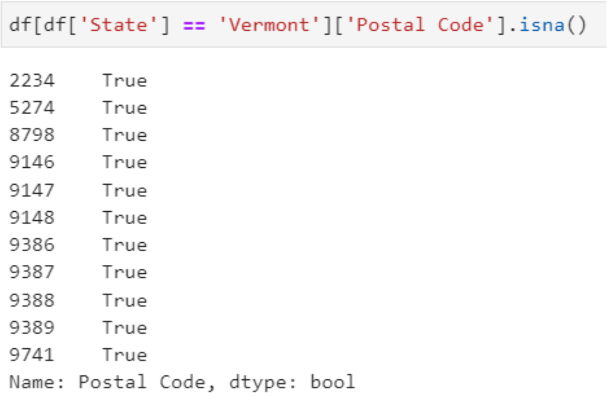
```
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Row ID          9994 non-null   int64
 1   Order ID        9994 non-null   object
 2   Order Date      9994 non-null   object
 3   Ship Date       9994 non-null   object
 4   Ship Mode       9994 non-null   object
 5   Customer ID     9994 non-null   object
 6   Customer Name   9994 non-null   object
 7   Segment         9994 non-null   object
 8   Country/Region  9994 non-null   object
 9   City            9994 non-null   object
 10  State           9994 non-null   object
 11  Postal Code     9983 non-null   float64
 12  Region          9994 non-null   object
 13  Product ID      9994 non-null   object
 14  Category        9994 non-null   object
 15  Sub-Category    9994 non-null   object
 16  Product Name    9994 non-null   object
 17  Sales           9994 non-null   float64
 18  Quantity        9994 non-null   int64
 19  Discount        9994 non-null   float64
 20  Profit          9994 non-null   float64
```

- Check for 5 digits in Postal Code:
  We found some Postal Code values with 4 digits while other Postal Code values were 5 digits. We need to figure out if there is a missing digit which is “0” or if it is a typo error for the 4 digit Postal Codes. If there is a missing zero value, it might belong to the northeast coast states.





```
df[df['Postal Code'].isna()].State

2234     Vermont
5274     Vermont
8798     Vermont
9146     Vermont
9147     Vermont
9148     Vermont
9386     Vermont
9387     Vermont
9388     Vermont
9389     Vermont
9741     Vermont
Name: State, dtype: object
```



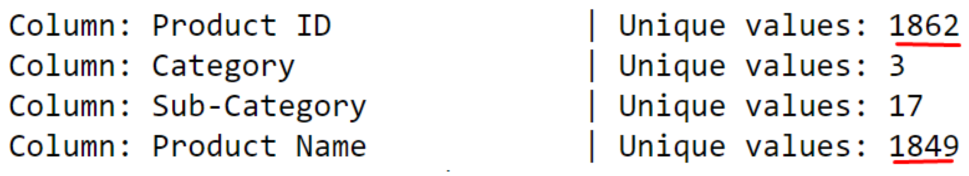
```
df[df['State'] == 'Vermont']['Postal Code'].isna()

2234     True
5274     True
8798     True
9146     True
9147     True
9148     True
9386     True
9387     True
9388     True
9389     True
9741     True
Name: Postal Code, dtype: bool
```

We checked the States for the NaN Postal Code rows and discovered that all of the NaN values belong to Vermont state. To double-check our findings we also checked the Postal Code values for every Vermont State entry. By doing that we discovered that for every row that State value is Vermont, the Postal Code is NaN. To solve this we can simply replace NaN values with Vermont Postal Code.

Product Name and Product ID (Uniqueness and Redundancy)

In this dataset, we observed inconsistencies in two of the columns i.e. **Product ID** and **Product Name**. Upon making an assumption that “Product IDs” are unique and that each “Product Name” has its own unique id, then based on this assumption there are few errors that are observed. In total there are found to be **1894 unique combinations** of product ID and product names.

However, analyzing these columns separately, it is observed that the number of Product IDs unique is 1862 and number of unique names is 1849, which is different and should not be the case.



```
Column: Product ID                  | Unique values: 1862
Column: Category                    | Unique values: 3
Column: Sub-Category                | Unique values: 17
Column: Product Name                | Unique values: 1849
```

Number of Unique Product IDs	1862
Number of Unique Product Names	1849

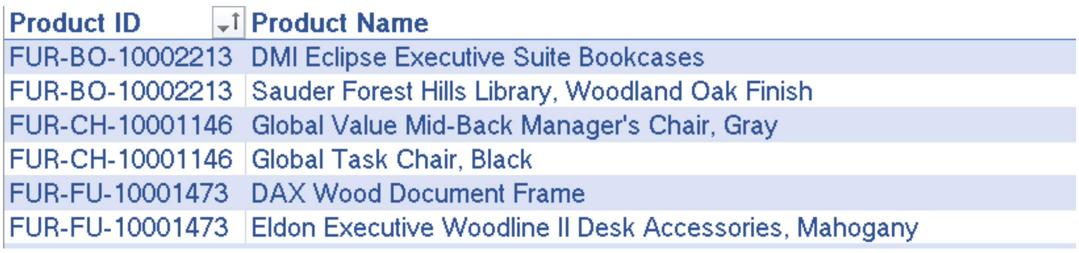
This means that either we have fewer "Product Name"s than "Product ID"s or number of “Product ID”s is more than required.

On performing further analysis using Excel, it was observed that:

1. There are **32 erroneous product ids** that belong to more than one product name
2. There are **17 erroneous product names** that belong to more than one product id

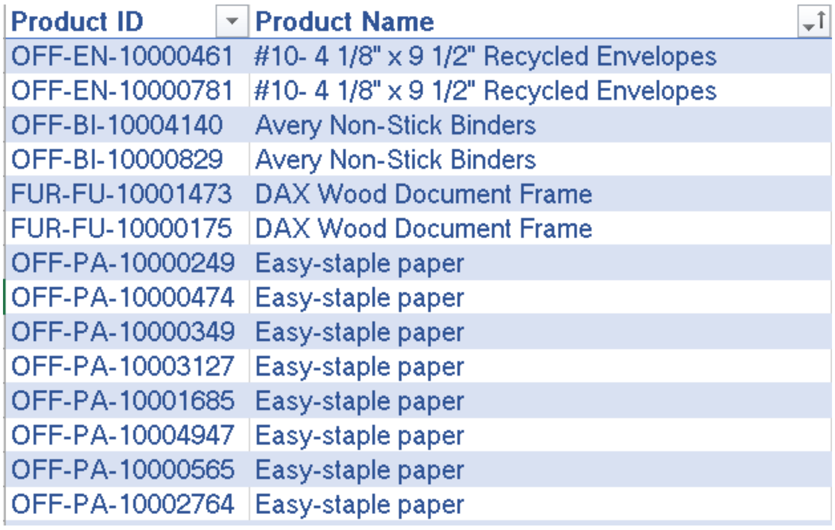
Number of Product IDs that belong to more than one product name	32
Number of Product Names that belong to more than one product id	17

Following are some example product-ids belonging to more than one product names:



Product ID	Product Name
FUR-BO-10002213	DMI Eclipse Executive Suite Bookcases
FUR-BO-10002213	Sauder Forest Hills Library, Woodland Oak Finish
FUR-CH-10001146	Global Value Mid-Back Manager's Chair, Gray
FUR-CH-10001146	Global Task Chair, Black
FUR-FU-10001473	DAX Wood Document Frame
FUR-FU-10001473	Eldon Executive Woodline II Desk Accessories, Mahogany

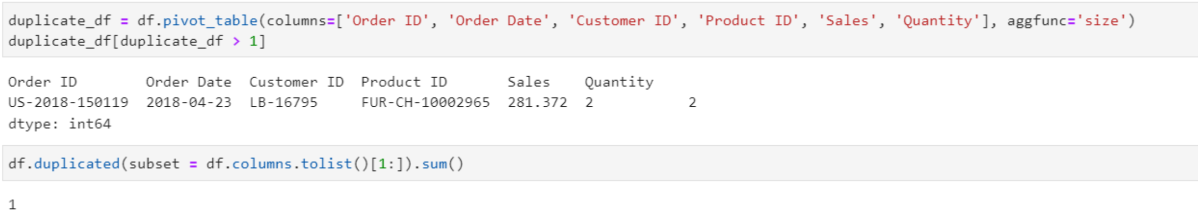
Following are some example product-names that belong to more than one product ids:



Product ID	Product Name
OFF-EN-10000461	#10- 4 1/8" x 9 1/2" Recycled Envelopes
OFF-EN-10000781	#10- 4 1/8" x 9 1/2" Recycled Envelopes
OFF-BI-10004140	Avery Non-Stick Binders
OFF-BI-10000829	Avery Non-Stick Binders
FUR-FU-10001473	DAX Wood Document Frame
FUR-FU-10000175	DAX Wood Document Frame
OFF-PA-10000249	Easy-staple paper
OFF-PA-10000474	Easy-staple paper
OFF-PA-10000349	Easy-staple paper
OFF-PA-10003127	Easy-staple paper
OFF-PA-10001685	Easy-staple paper
OFF-PA-10004947	Easy-staple paper
OFF-PA-10000565	Easy-staple paper
OFF-PA-10002764	Easy-staple paper

In order to maintain the uniqueness and reduce redundancy within the dataset, in our next task, we prefer taking the approach of reassigning product names and ids with the most common occurrences amongst them.

Duplicates



```
duplicate_df = df.pivot_table(columns=['Order ID', 'Order Date', 'Customer ID', 'Product ID', 'Sales', 'Quantity'], aggfunc='size')
duplicate_df[duplicate_df > 1]
```

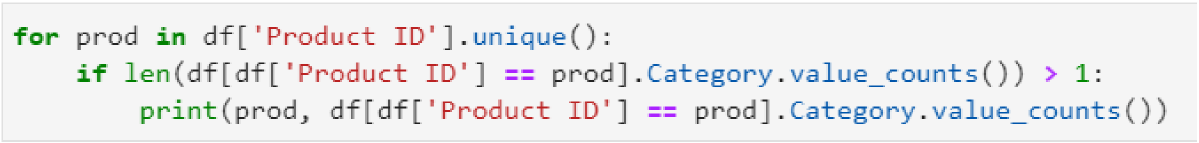
```
Order ID         Order Date  Customer ID  Product ID       Sales    Quantity
US-2018-150119   2018-04-23  LB-16795     FUR-CH-10002965  281.372  2           2
dtype: int64
```

```
df.duplicated(subset = df.columns.tolist()[1:]).sum()
```

```
1
```

We created a pivot table by filtering out dataframe with ['Order ID', 'Order Date', 'Customer ID', 'Product ID', 'Sales', 'Quantity'] columns. It appears that we have 2 entries for the shown row. To double check we used the pandas duplicated function by discluding the rowid column.

Product ID and Category:



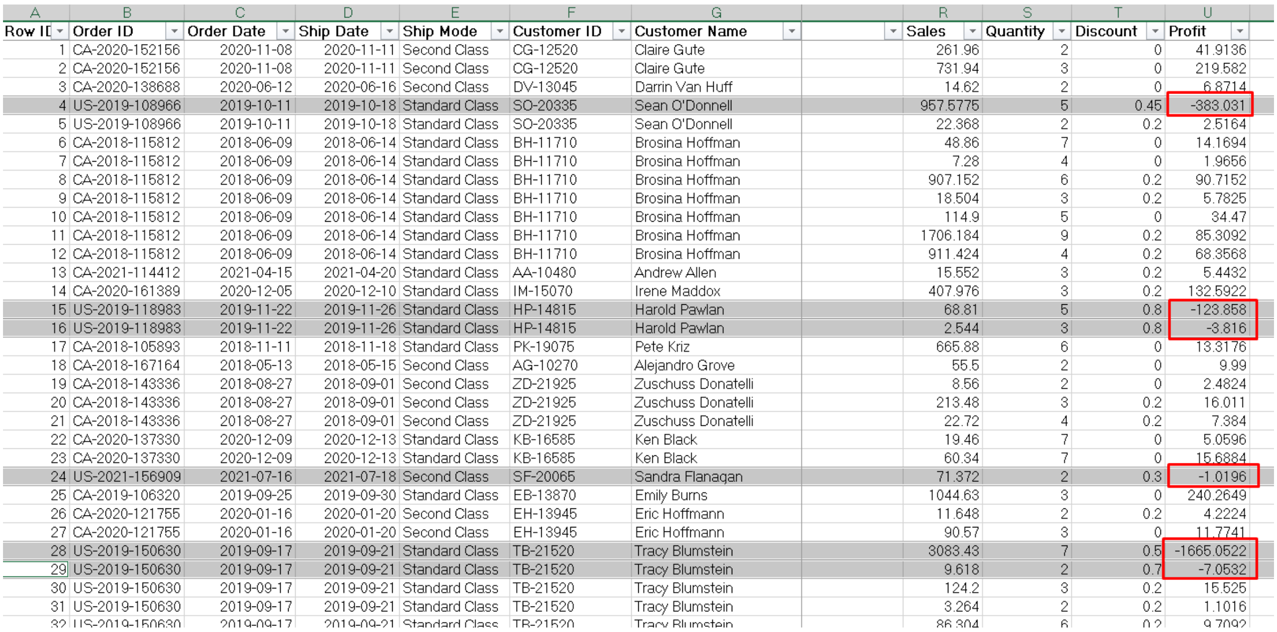
```
for prod in df['Product ID'].unique():
    if len(df[df['Product ID'] == prod].Category.value_counts()) > 1:
        print(prod, df[df['Product ID'] == prod].Category.value_counts())
```

All products are assigned to a single category, since there are no printed outputs from the code block above.

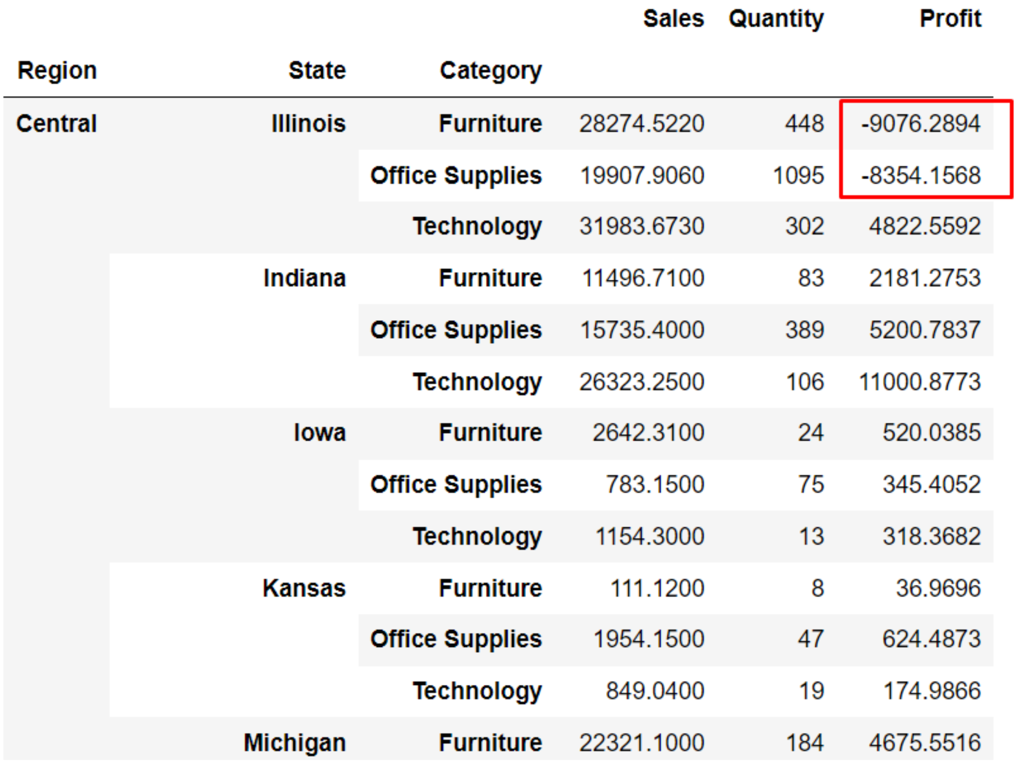
Profit

We found potential inconsistencies in the profit as is negative in many cases; we need to investigate why this is happening.

We calculated the Cost of Sales as explained in item 4.



A	B	C	D	E	F	G		R	S	T	U
Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name		Sales	Quantity	Discount	Profit
1	CA-2020-152156	2020-11-08	2020-11-11	Second Class	CG-12520	Claire Gute		261.96	2	0	41.9136
2	CA-2020-152156	2020-11-08	2020-11-11	Second Class	CG-12520	Claire Gute		731.94	3	0	219.582
3	CA-2020-138688	2020-06-12	2020-06-16	Second Class	DV-13045	Darrin Van Huff		14.62	2	0	6.8714
4	US-2019-108966	2019-10-11	2019-10-18	Standard Class	SO-20335	Sean O'Donnell		957.5775	5	0.45	-383.031
5	US-2019-108966	2019-10-11	2019-10-18	Standard Class	SO-20335	Sean O'Donnell		22.368	2	0.2	2.5164
6	CA-2018-115812	2018-06-09	2018-06-14	Standard Class	BH-11710	Brosina Hoffman		48.86	7	0	14.1694
7	CA-2018-115812	2018-06-09	2018-06-14	Standard Class	BH-11710	Brosina Hoffman		7.28	4	0	1.9656
8	CA-2018-115812	2018-06-09	2018-06-14	Standard Class	BH-11710	Brosina Hoffman		907.152	6	0.2	90.7152
9	CA-2018-115812	2018-06-09	2018-06-14	Standard Class	BH-11710	Brosina Hoffman		18.504	3	0.2	5.7825
10	CA-2018-115812	2018-06-09	2018-06-14	Standard Class	BH-11710	Brosina Hoffman		114.9	5	0	34.47
11	CA-2018-115812	2018-06-09	2018-06-14	Standard Class	BH-11710	Brosina Hoffman		1706.184	9	0.2	85.3092
12	CA-2018-115812	2018-06-09	2018-06-14	Standard Class	BH-11710	Brosina Hoffman		911.424	4	0.2	68.3568
13	CA-2021-114412	2021-04-15	2021-04-20	Standard Class	AA-10480	Andrew Allen		15.552	3	0.2	5.4432
14	CA-2020-161389	2020-12-05	2020-12-10	Standard Class	IM-15070	Irene Maddox		407.976	3	0.2	132.5922
15	US-2019-118983	2019-11-22	2019-11-26	Standard Class	HP-14815	Harold Pawlan		68.81	5	0.8	-123.858
16	US-2019-118983	2019-11-22	2019-11-26	Standard Class	HP-14815	Harold Pawlan		2.544	3	0.8	-3.816
17	CA-2018-105893	2018-11-11	2018-11-18	Standard Class	PK-19075	Pete Kriz		665.88	6	0	13.3176
18	CA-2018-167164	2018-05-13	2018-05-15	Second Class	AG-10270	Alejandro Grove		55.5	2	0	9.99
19	CA-2018-143336	2018-08-27	2018-09-01	Second Class	ZD-21925	Zuschuss Donatelli		8.56	2	0	2.4824
20	CA-2018-143336	2018-08-27	2018-09-01	Second Class	ZD-21925	Zuschuss Donatelli		213.48	3	0.2	16.011
21	CA-2018-143336	2018-08-27	2018-09-01	Second Class	ZD-21925	Zuschuss Donatelli		22.72	4	0.2	7.384
22	CA-2020-137330	2020-12-09	2020-12-13	Standard Class	KB-16585	Ken Black		19.46	7	0	5.0596
23	CA-2020-137330	2020-12-09	2020-12-13	Standard Class	KB-16585	Ken Black		60.34	7	0	15.6884
24	US-2021-156909	2021-07-16	2021-07-18	Second Class	SF-20065	Sandra Flanagan		71.372	2	0.3	-1.0196
25	CA-2019-106320	2019-09-25	2019-09-30	Standard Class	EB-13870	Emily Burns		1044.63	3	0	240.2649
26	CA-2020-121755	2020-01-16	2020-01-20	Second Class	EH-13945	Eric Hoffmann		11.648	2	0.2	4.2224
27	CA-2020-121755	2020-01-16	2020-01-20	Second Class	EH-13945	Eric Hoffmann		90.57	3	0	11.7741
28	US-2019-150630	2019-09-17	2019-09-21	Standard Class	TB-21520	Tracy Blumstein		3083.43	7	0.5	-1665.0522
29	US-2019-150630	2019-09-17	2019-09-21	Standard Class	TB-21520	Tracy Blumstein		9.618	2	0.7	-7.0532
30	US-2019-150630	2019-09-17	2019-09-21	Standard Class	TB-21520	Tracy Blumstein		124.2	3	0.2	15.525
31	US-2019-150630	2019-09-17	2019-09-21	Standard Class	TB-21520	Tracy Blumstein		3.264	2	0.2	1.1016
32	US-2019-150630	2019-09-17	2019-09-21	Standard Class	TB-21520	Tracy Blumstein		86.304	6	0.2	9.7092



			Sales	Quantity	Profit
Region	State	Category			
Central	Illinois	Furniture	28274.5220	448	-9076.2894
		Office Supplies	19907.9060	1095	-8354.1568
		Technology	31983.6730	302	4822.5592
	Indiana	Furniture	11496.7100	83	2181.2753
		Office Supplies	15735.4000	389	5200.7837
		Technology	26323.2500	106	11000.8773
	Iowa	Furniture	2642.3100	24	520.0385
		Office Supplies	783.1500	75	345.4052
		Technology	1154.3000	13	318.3682
	Kansas	Furniture	111.1200	8	36.9696
		Office Supplies	1954.1500	47	624.4873
		Technology	849.0400	19	174.9866
	Michigan	Furniture	22321.1000	184	4675.5516

2. Target Audience

Report	Target audience	Explanation	Use
Operational	1. Product suppliers 2. Sales team manager 3. Tax accountants 4. Logistics team 5. HR team	1. Buy wholesale based on sell data this quarter/month 2. Update the sale performance info 3. Calculating tax 4. Assign more drivers to the busy area / reschedule delivery shift once we get sales data grouped by region 5. Recruit more operators and drivers to busy areas	Monitor and control: Monitor the current sales performance in by KPIs
Executive	1. CEO 2. Sales team manager 3. Stakeholders	1. Update sale performance info 2. Making sale strategies 3. Seeking investment	Decision-making: making decisions on product management, importing new products, etc. Performance improvement: Build sales strategies or promotions. Research analysis: Test the hypothesis and big deeper into the data.

3. Context and additional Assumptions

Presenting **metadata as the context** of the given spreadsheet or raw data:

Descriptive Metadata

File Name	Sample - Superstore
Type of File	Excel (.xls)
Size of File	3.22 MB
Number of Rows	9993
Number of columns	21
Column Names	Order Id, Order Date, Ship Date, Ship Mode, Customer ID, Customer Name, Segment, Country/ Region, City, State, Postal Code, Region, Product ID, Category, Sub-category, Product Name, Sales, Quantity, Discount, Profit
Geographic scope	United States
Inconsistency	Found between Product ID and Product Name
Null Values	Found in Postal Code

Some points to add -

- This **data is about** the sales records of a company that buys products from different suppliers and delivers them to customers across the United States.
- During analysis, the raw data has been **checked** in terms of completeness, consistency and uniqueness and **processed** to achieve those levels.

- The **purpose** of the operational report is to monitor current sales performance and quickly spot areas that need further attention in a day-to-day basis. The Executive Report can be used for decision making regarding products.

- KPIs for Operational Report:

  We have chosen GroupBy sales, profit, expenses columns and created several new columns - total sales, total cost, total profit, disc%, gross profit margin% as our KPIs. As for the target audiences, we keep manager for the HR team. For the product suppliers and sales managers, our focus is to provide performance based on product categories.

- KPIs for Executive Report:
  To decide on sales strategies, we report on sales column by order date(group in year). Three new columns have been introduced 2018 vs. 2019, 2019 vs. 2020 and 2020 vs. 2021 which can provide trending information regarding sales.

4. Operational and Executive Reports

Operational report

We are going to list nine columns, partially using original names from the data, partially derived from a few columns.

manager	use the original column name.	We join the sheets “orders” and “people” together by matching “region”, by adding one more column “manager”. This will keep 1-1 relationship between “manager” and “region” column in the following table creation.
region	use the original column name.	We groupby the data first by their regions(total four).
state	use the original column name.	We groupby the data secondly by their states, under each region.
category	use the original column name.	We groupby the data the third time, by each category, under state.
total sales($)	newly created column	Since the “sales” column refers to unit price; we multiply the “sales” with

		“quantity” to get the total sale for each row first. Then, we aggregate(sum) all the total sales under each category.
total profit($)	use the original column name “profit”	We aggregate(sum) all profits under each category.
disc.(%)	newly created column	To see how much these products are discounted, we multiply the disc. rate by corresponding total sales (calculated above) for each row, to get the exact discount amount. Then, we sum them up under each category. Finally, we divide this total discount amount by the total sales(calculated above), under each category.
total expenses($)	newly created column	we minus the “total profit”(calculated above) as well as the total discount (calculated above) from “total sales”(calculated above), under each category.
gross profit margin(%)	newly created column	To oversee how much the company earned, we divide the “total profit”(calculated above) by “total sales”(calculated above), under each category.

Executive Report

category	Original column from Orders tab	Sum sales under each product type
sub-category	Original column from Orders tab	Sum sales under each product sub-type of each category
year 2018($)	Order date(original column from orders tab) between 2018-01-01 and 2018-12-31	Sum sales of 2018 only

year 2019($)	Order date(original column from orders tab) between 2019-01-01 and 2019-12-31	Sum sales of 2019 only
year 2020($)	Order date(original column from orders tab) between 2020-01-01 and 2020-12-31	Sum sales of 2020 only
year 2021($)	Order date(original column from orders tab) between 2021-01-01 and 2021-12-31	Sum sales of 2021 only
yearly percentage(%) from 2018 to 2019 [“+” is increasing, “-” is decreasing]	Newly created column	Comparison of gross sales between these 2 years
yearly percentage(%) from 2019 to 2020 [“+” is increasing, “-” is decreasing]	Newly created column	Comparison of gross sales between these 2 years
yearly percentage(%) from 2020 to 2021 [“+” is increasing, “-” is decreasing]	Newly created column	Comparison of gross sales between these 2 years

5. Empty templates

Order ID	Region	State	Category	Total Sales ($)	Quantity	Discount (%)	Total Profit ($)	Total expenses ($)	Gross Profit Margin (%)
1	abc			0	35	35	20	0	#DIV/0!
				0				0	#DIV/0!
				0		0		0	#DIV/0!
				0		0		0	#DIV/0!
				0		0		0	#DIV/0!
				0		0		0	#DIV/0!
				0		0		0	#DIV/0!
				0		0		0	#DIV/0!
				0		0		0	#DIV/0!
				0		0		0	#DIV/0!
				0		0		0	#DIV/0!
				0		0		0	#DIV/0!
				0		0		0	#DIV/0!
				0		0		0	#DIV/0!
				0		0		0	#DIV/0!
				0		0		0	#DIV/0!
				0		0		0	#DIV/0!
				0		0		0	#DIV/0!
				0		0		0	#DIV/0!
				0		0		0	#DIV/0!
				0		0		0	#DIV/0!
				0		0		0	#DIV/0!

Subtotal					**0**	**35**	**35**	**20**	**#VALUE!**	**#VALUE!**

Executive Report :

	Category	**Sub - Category**	**Year 2018 ($)**	**Year 2019 ($)**	**Year 2020 ($)**	**Year 2021 ($)**	**Yearly (%) from 2018 to 2019**	**Yearly (%) from 2019 to 2020**	**Yearly (%) from 2020 to 2021**
Subtotal									