

SALES PROJECT

Business store large amounts of data in order to make better decisions and they do this through the process of data analytics. Data Analytics help businesses optimize their performance, analyze customer trends and satisfaction. In this project. We will give a report on how well their products are doing which in turn will help business reduce cost and identify more efficient ways of doing things.

Data analysts spend more time on cleaning data to make the data perfect for analysis to begin. Time spent more data cleansing becomes more tiring when business needs report urgently hence for this project we make use of **Alteryx** to perform our data cleaning then load to our **MS SQL Server** for later visualization in **Tableau**.

Analysis performed on this dataset:

1. Which product line are giving the most sales and least sales?
2. Which year made most sales?
3. Which Deal Size had the highest revenue?
4. What was the best month for sales?
5. Which product generate highest revenue in the best month?
6. What products are often sold together?
7. Which Customers should be targeted to generate revenue ?

We begin by cleaning our data with alteryx then connecting it to our MS SQL Server



1. From this analysis we could see classic cars is the best product line they have followed by vintage cars because they make most of their sales under classic cars. This is very useful as more attention can be given to classic cars to in order to meet the demands of that and develop strategies in increasing sales for other products. For this analysis, Trains have the least sales

SQLQuery1.sql - K...KYE\Nana Ama (60))

```
---INSPECTING DATA
select * from [dbo].[sales_data_sample]

---CHECKING UNIQUE VALUES
select distinct status from [dbo].[sales_data_sample] --Nice one to plot in tableau
select distinct year_id from [dbo].[sales_data_sample]
select distinct PRODUCTLINE from [dbo].[sales_data_sample] ---Nice to plot in Tableau
select distinct COUNTRY from [dbo].[sales_data_sample] ---Nice to plot in Tableau
select distinct DEALSIZE from [dbo].[sales_data_sample] ---Nice to plot in Tableau
select distinct TERRITORY from [dbo].[sales_data_sample] ---Nice to plot in Tableau

---ANALYSIS
----Let's start by grouping sales by productline
select PRODUCTLINE, sum(sales) Revenue
from [dbo].[sales_data_sample]
group by PRODUCTLINE
order by 2 desc
```

100 %

Results Messages

	PRODUCTLINE	Revenue
1	Classic Cars	3919615.6607666
2	Vintage Cars	1903150.83557129
3	Motorcycles	1166388.3392334
4	Trucks and Buses	1127789.84326172
5	Planes	975003.571350098
6	Ships	714437.130126953
7	Trains	226243.468994141

Query executed successfully.

2. Now lets check the sales across the year as we know which product sells the most.

SQLQuery1.sql - K...KYE\Nana Ama (60))

```
--ANALYSIS
----Let's start by grouping sales by productline
select PRODUCTLINE, sum(sales) Revenue
from [dbo].[sales_data_sample]
group by PRODUCTLINE
order by 2 desc

--WHICH YEAR MADE MOST SALES
select YEAR_ID, sum(sales) Revenue
from [dbo].[sales_data_sample]
group by YEAR_ID
order by 2 desc
```

100 %

	YEAR_ID	Revenue
1	2004	4724162.59338379
2	2003	3516979.54724121
3	2005	1791486.7086792

Query executed successfully.

It appears in 2004, they made their most sales followed by 2003. The company can identify what they did in 2004 that made them make more sales and replicate in other years. Company can also find out what made them make less sales in the other years and work on improving their sales strategy.

```

group by YEAR_ID
order by 2 desc

select distinct MONTH_ID from [dbo].[sales_data_sample]
where year_id = 2003

select distinct MONTH_ID from [dbo].[sales_data_sample]
where year_id = 2004

select distinct MONTH_ID from [dbo].[sales_data_sample]
where year_id = 2005

```

MONTH_ID
9
3
12
6
7
1
10
4

MONTH_ID
1
2
3
4
5

Let's find out what happened in 2005. Maybe the company did not operate throughout the year.

When comparing 2005 year to the rest of the years, we could see that in 2005, the company operated in few months(5 months) and did a full year operation in 2003 and 2004. Other factors could account for why it wasn't a full year operation but we do know it had the lowest revenue due to this.

3. Which Deal Size had the highest revenue ?

SQLQuery1.sql - K...KYE\Nana Ama (60)*

```

--WHICH YEAR MADE MOST SALES
select YEAR_ID, sum(sales) Revenue
from [dbo].[sales_data_sample]
group by YEAR_ID
order by 2 desc

select DEALSIZE, sum(sales) Revenue
from [dbo].[sales_data_sample]
group by DEALSIZE
order by 2 desc

```

DEALSIZE	Revenue
Medium	6087432.23974609
Small	2643077.34881592
Large	1302119.26074219

Query executed successfully.

The medium size deals generate more revenues. This is very useful as production can be increased in order to meet the demand for such deals. And could use for some marketing strategies for small and large deal sizes to also make more revenue .

4. What was the best month for sales?

SQLQuery1.sql - K...KYE\Nana Ama (60))*

```

from [dbo].[sales_data_sample]
group by YEAR_ID
order by 2 desc

---Which deal size generate the highest revenue
select DEALSIZE, sum(sales) Revenue
from [dbo].[sales_data_sample]
group by DEALSIZE
order by 2 desc

---What was the best month for sales in a specific year? How much was earned that month?
select MONTH_ID, sum(sales) Revenue, count(ORDERNUMBER) Frequency
from [dbo].[sales_data_sample]
where YEAR_ID = 2004 --change year to see the rest
group by MONTH_ID
order by 2 desc

```

100 %

Results Messages

	MONTH_ID	Revenue	Frequency
1	11	1089048.00762939	301
2	10	552924.250793457	159
3	8	461501.267944336	133
4	12	372802.658325195	110
5	7	327144.090942383	91
6	9	320750.908569336	95
7	1	316577.418762207	91
8	2	311419.530029297	86
9	6	286674.221130371	85
10	5	273438.389648438	74
11	4	206148.119750977	64
12	3	205733.729858398	56

Query executed successfully.

```

from [dbo].[sales_data_sample]
group by YEAR_ID
order by 2 desc

---Which deal size generate the highest revenue
select DEALSIZE, sum(sales) Revenue
from [dbo].[sales_data_sample]
group by DEALSIZE
order by 2 desc

---What was the best month for sales in a specific year? How much was earned that month?
select MONTH_ID, sum(sales) Revenue, count(ORDERNUMBER) Frequency
from [dbo].[sales_data_sample]
where YEAR_ID = 2003 --change year to see the rest
group by MONTH_ID
order by 2 desc

```

100 %

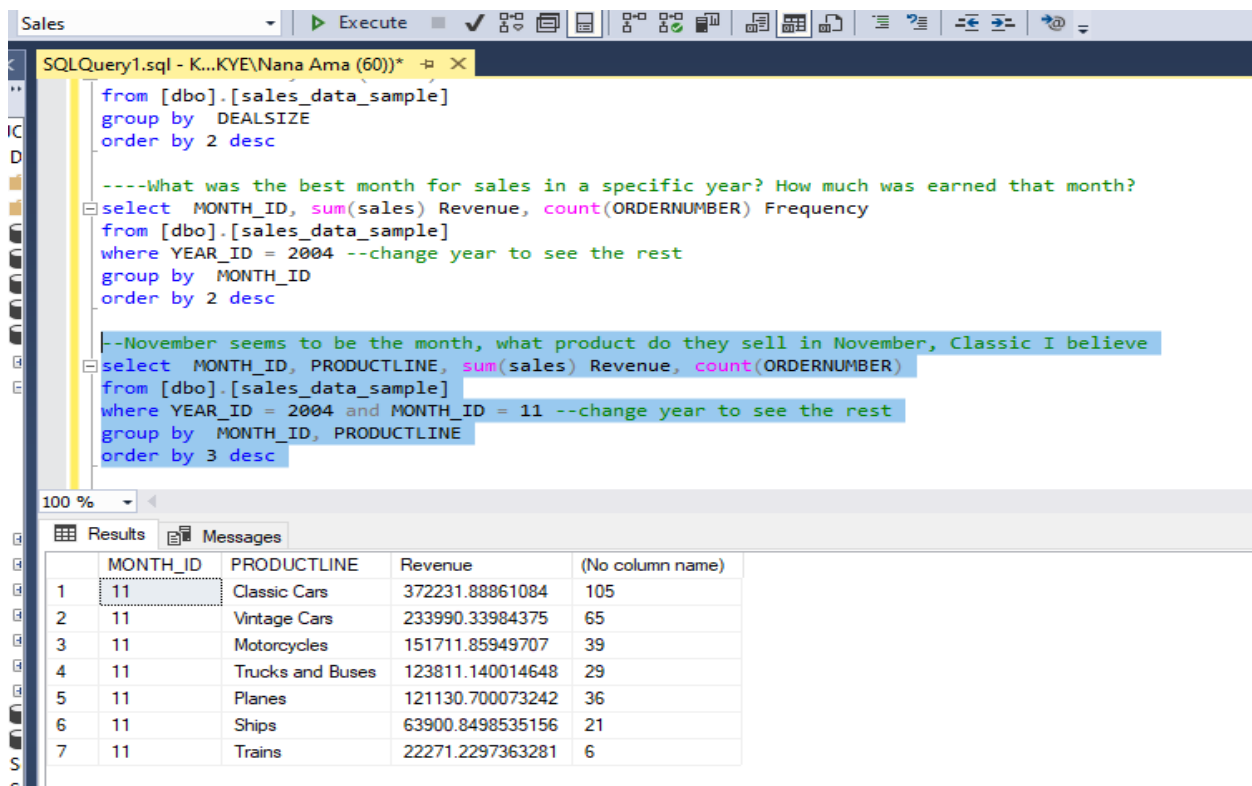
Results Messages

	MONTH_ID	Revenue	Frequency
1	11	1029837.66271973	296
2	10	568290.971557617	158
3	9	263973.360961914	76
4	12	261876.459899902	70
5	4	201609.551086426	58
6	8	197809.300048828	58
7	5	192673.109985352	58
8	7	187731.880493164	50
9	3	174504.900085449	50
10	6	168082.56072998	46
11	2	140836.189941406	41
12	1	129753.599731445	39

Comparing both 2003 and 2004 as the top two revenue generating years, November is the best month sales for 2004 having 301 orders and 2003 having 296 orders. The total revenue is again more than twice of the second best month. (We can also find out which city or country has the highest revenue by changing the "DEALSIZE" in the code to country or city)

Since November seems to be the best month, let's find out what products are sold in November. Classic Cars I believe. Let's find out.

5. What products are sold in November ?



The screenshot shows a SQL query window with the following code:

```

from [dbo].[sales_data_sample]
group by DEALSIZE
order by 2 desc

----What was the best month for sales in a specific year? How much was earned that month?
select MONTH_ID, sum(sales) Revenue, count(ORDERNUMBER) Frequency
from [dbo].[sales_data_sample]
where YEAR_ID = 2004 --change year to see the rest
group by MONTH_ID
order by 2 desc

--November seems to be the month, what product do they sell in November, Classic I believe
select MONTH_ID, PRODUCTLINE, sum(sales) Revenue, count(ORDERNUMBER)
from [dbo].[sales_data_sample]
where YEAR_ID = 2004 and MONTH_ID = 11 --change year to see the rest
group by MONTH_ID, PRODUCTLINE
order by 3 desc

```

The Results pane shows the following data:

	MONTH_ID	PRODUCTLINE	Revenue	(No column name)
1	11	Classic Cars	372231.88861084	105
2	11	Vintage Cars	233990.33984375	65
3	11	Motorcycles	151711.85949707	39
4	11	Trucks and Buses	123811.140014648	29
5	11	Planes	121130.700073242	36
6	11	Ships	63900.8498535156	21
7	11	Trains	22271.2297363281	6

Our guess was right. Classic Cars are the products sold in November and generates the highest revenue for both 2003 and 2004.

Now let's get more interesting and find out who our best customer is. (This could be answered with an RFM). An RFM analysis is a marketing technique used to quantitatively rank and group customers based on the recency, frequency, and monetary total of their recent transactions to identify the best customers and perform targeted marketing campaigns. The system assigns each customer numerical scores based on these factors to provide an objective analysis. RFM analysis is based on the marketing adage that "80% of your business comes from 20% of your customers."

```
Sales
SQLQuery1.sql - K...KYE\Nana Ama (60)) *
----Who is our best customer (this could be best answered with RFM)
DROP TABLE IF EXISTS #rfm
;with rfm as
(
    select
        CUSTOMERNAME,
        sum(sales) MonetaryValue,
        avg(sales) AvgMonetaryValue,
        count(ORDERNUMBER) Frequency,
        max(ORDERDATE) last_order_date,
        (select max(ORDERDATE) from [dbo].[sales_data_sample]) max_order_date,
        DATEDIFF(DD, max(ORDERDATE), (select max(ORDERDATE) from [dbo].[sales_data_sample])) Recency
    from [dbo].[sales_data_sample]
    group by CUSTOMERNAME
),
rfm_calc as
(
    select r.*,
        NTILE(4) OVER (order by Recency desc) rfm_recency,
        NTILE(4) OVER (order by Frequency) rfm_frequency,
        NTILE(4) OVER (order by MonetaryValue) rfm_monetary
    from rfm r
)
select
    c.*, rfm_recency+ rfm_frequency+ rfm_monetary as rfm_cell,
    cast(rfm_recency as varchar) + cast(rfm_frequency as varchar) + cast(rfm_monetary as varchar) rfm_cell_string
into #rfm
from rfm_calc c

select CUSTOMERNAME , rfm_recency, rfm_frequency, rfm_monetary,
    case
        when rfm_cell_string in (111, 112 , 121, 122, 123, 132, 211, 212, 114, 141) then 'lost_customers' --lost customers
        when rfm_cell_string in (133, 134, 143, 244, 334, 343, 344, 144) then 'slipping away, cannot lose' -- (Big spenders who haven't purchased lately) slipping away
        when rfm_cell_string in (311, 411, 331) then 'new customers'
        when rfm_cell_string in (222, 223, 233, 322) then 'potential churners'
        when rfm_cell_string in (323, 333, 321, 422, 332, 432) then 'active' --(Customers who buy often & recently, but at low price points)
        when rfm_cell_string in (433, 434, 443, 444) then 'loyal'
    end rfm_segment
```

We wrote case statements to help us identify lost customers, slipping away customers who we cannot afford to lose (Big spenders who have not purchased lately), new customers, potential churners, Active customers (They do buy often but at low prices) and the loyal (They often, buy big spenders). This is going to help the marketing team when sending marketing campaigns /advertisement to know which customers to target for which program.

SQLQuery1.sql - K...KYE\Nana Ama (60))*

```

cast(rfm_recency as varchar) + cast(rfm_frequency as varchar) + cast(rfm_monetary as varchar) rfm_cell_string
into #rfm
from rfm_calc c

select CUSTOMERNAME, rfm_recency, rfm_frequency, rfm_monetary,
case
when rfm_cell_string in (111, 112, 121, 122, 123, 132, 211, 212, 114, 141) then 'lost_customers' --lost customers
when rfm_cell_string in (133, 134, 143, 244, 334, 343, 344, 144) then 'slipping away, cannot lose' -- (Big spenders who haven't
when rfm_cell_string in (311, 411, 331) then 'new customers'
when rfm_cell_string in (222, 223, 233, 322) then 'potential churners'
when rfm_cell_string in (323, 333, 321, 422, 332, 432) then 'active' --(Customers who buy often & recently, but at low price p
when rfm_cell_string in (433, 434, 443, 444) then 'loyal'
end rfm_segment
from #rfm

```

100 %

Results Messages

	CUSTOMERNAME	rfm_recency	rfm_frequency	rfm_monetary	rfm_segment
76	Rovelli Gifts	2	4	4	slipping away...
77	L'ordine Souvenirs	4	4	4	loyal
78	Saveley & Henriot...	1	4	4	slipping away...
79	Danish Wholesal...	4	4	4	loyal
80	Salzburg Collecta...	4	4	4	loyal
81	Corporate Gift Ide...	3	4	4	slipping away...
82	Souvenirs And T...	4	4	4	loyal
83	Anna's Decoratio...	3	4	4	slipping away...
84	AV Stores, Co.	2	4	4	slipping away...
85	The Sharp Gifts ...	4	4	4	loyal
86	Land of Toys Inc.	2	4	4	slipping away...
87	Dragon Souvenie...	3	4	4	slipping away...
88	La Rochelle Gifts	4	4	4	loyal
89	Muscle Machine I...	3	4	4	slipping away...
90	Australian Collect...	3	4	4	slipping away...
91	Mini Gifts Distribut...	4	4	4	loyal
92	Euro Shopping C...	4	4	4	loyal

Query executed successfully.

6. What products are often sold together?

We will first conduct an XML path analysis to help us figure out which products are sold together.

SQLQuery1.sql - K...KYE\Nana Ama (60))*

```

--What products are most often sold together?
--select * from [dbo].[sales_data_sample] where ORDERNUMBER =
select distinct OrderNumber, stuff(
(
select ', ' + PRODUCTCODE
from [dbo].[sales_data_sample] j
where ORDERNUMBER in
(
select ORDERNUMBER
from (
select ORDERNUMBER, count(*) rn
FROM [dbo].[sales_data_sample]
where STATUS = 'Shipped'
group by ORDERNUMBER
)m
where rn = 2
and j.ORDERNUMBER = k.ORDERNUMBER
for xml path (''))
, 1, 1, '' ) ProductCodes
from [dbo].[sales_data_sample] k
order by 2 desc

```

100 %

Results Messages

	OrderNumber	ProductCodes
5	10290	S18_3320,S24_4258
6	10323	S18_3320,S18_4600
7	10265	S18_3278,S18_3482
8	10130	S18_3029,S18_3856
9	10269	S18_2957,S24_4258
10	10255	S18_2795,S24_2022
11	10243	S18_2325,S24_1937
12	10409	S18_2325,S24_1937
13	10303	S18_2248,S24_3969
14	10125	S18_1242,S18_2705

We can see that for line 11 and 12; we had different customers who bought same products. This is helpful to track which products are easily sold together and could help in making decisions when running promotions or campaigns by advertising those things together since you will have a higher chance of customers buying those things together. (Where $rn = 2$, can be switched to 3, and by doing so will bring orders with three products purchased together.)

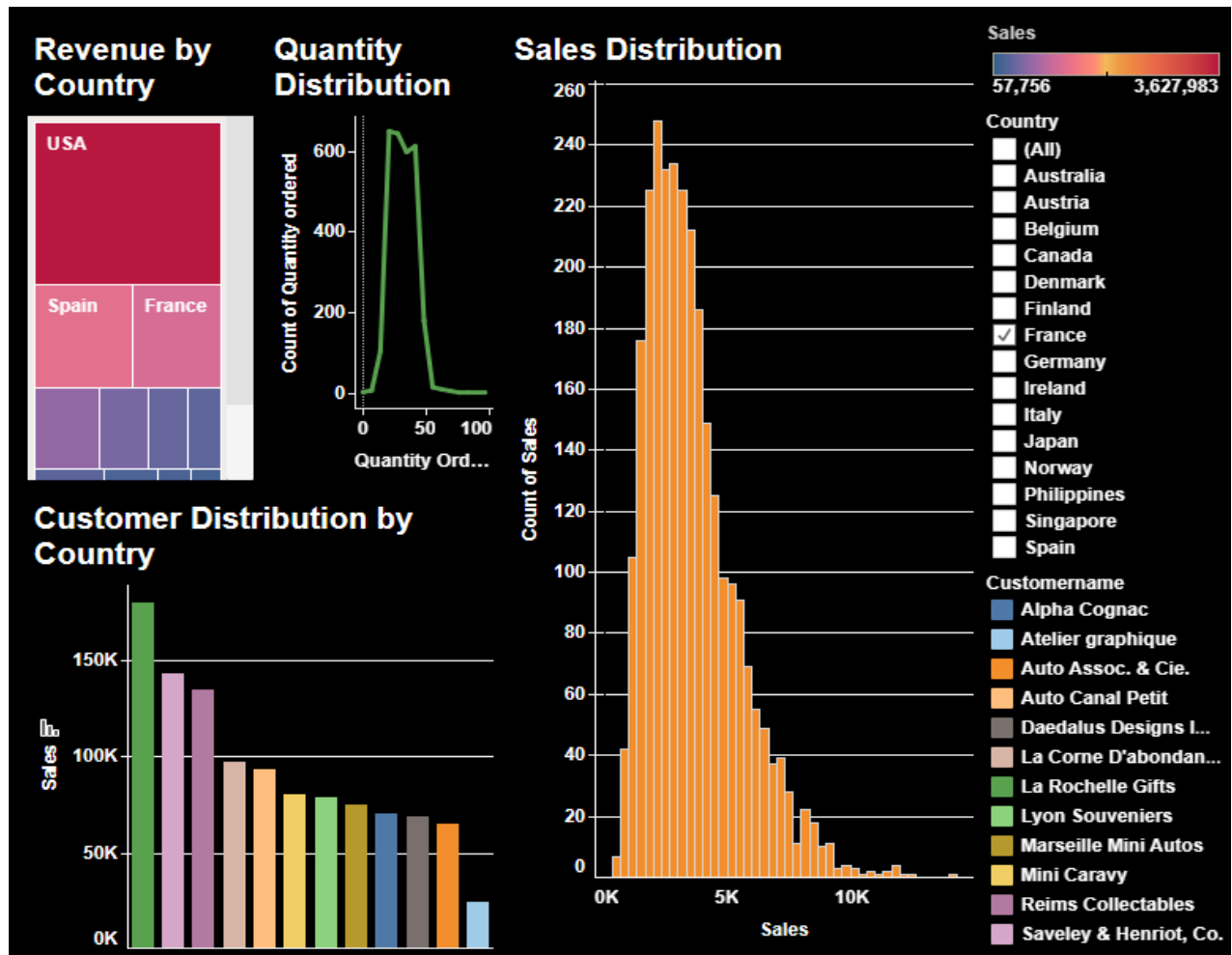
Now let's visualize this data in Tableau for presentation

DATA LOADED INTO TABLEAU

The screenshot shows the Tableau interface with the 'sales_data_sample' data source loaded. The sidebar on the left contains the 'Connections' and 'Files' sections. The main view displays a table with columns: Ordernumber, Quantityordered, Priceeach, Orderlinenumber, Sales, and Orderdate. The table shows 25 fields and 2823 rows of data.

#	#	#	#	#	#	#
Ordernumber	Quantityordered	Priceeach	Orderlinenumber	Sales	Orderdate	St
10.107	30	95.7000	2	2.871.00	2/24/2003 12:00:00 AM	\$
10.121	34	81.3500	5	2.765.90	5/7/2003 12:00:00 AM	\$
10.134	41	94.7400	2	3.884.34	7/1/2003 12:00:00 AM	\$
10.145	45	83.2600	6	3.746.70	8/25/2003 12:00:00 AM	\$
10.159	49	100.0000	14	5.205.27	10/10/2003 12:00:00 AM	\$
10.168	36	96.6600	1	3.479.76	10/28/2003 12:00:00 AM	\$
10.180	29	86.1300	9	2.497.77	11/11/2003 12:00:00 AM	\$

FINAL OUTPUT (To interact with the dashboard: click on this link: [Dashboard1](#))



Dashboard2

