Regional and Market Analysis

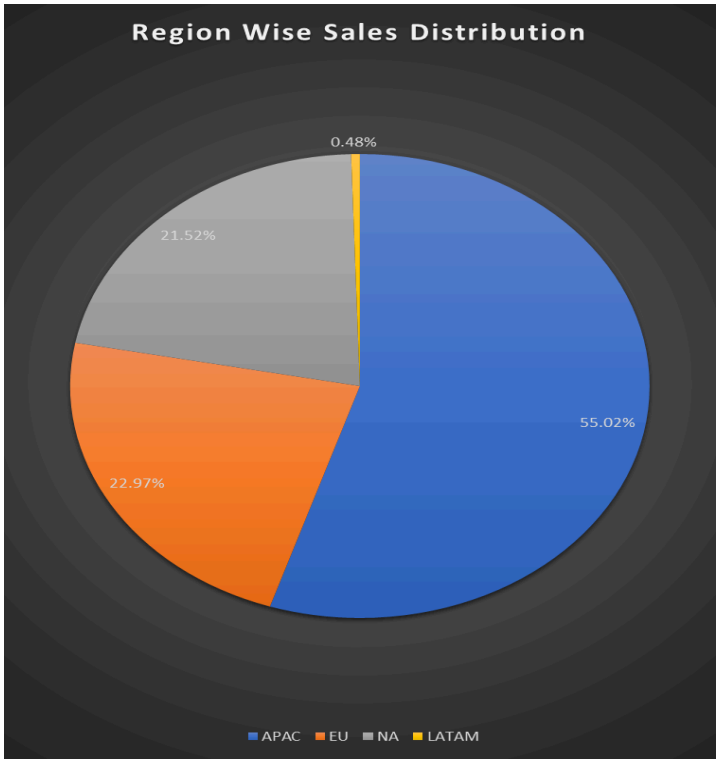
Revenue by Region (Business Q)

```
-- QUESTION.
-- Which region and markets contribute the most to the overall revenue and profitability.
-- sales amt vs region wise analysis.
create view sales_table as
(select t1.*,
t2.date,t2.product_code,t2.sold_quantity,t2.fiscal_year,
t3.gross_price,
t4.division,t4.segment,t4.category,t4.product,t4.variant,
t2.sold_quantity * t3.gross_price as sales_amount
from dim_customer t1 inner join fact_sales_monthly t2 on t1.customer_code = t2.customer_code
inner join fact_gross_price t3 on t3.product_code = t2.product_code and t2.fiscal_year = t3.fiscal_year
inner join dim_product t4 on t2.product_code = t4.product_code);
```

```
-- apac region highest sales.
select region,sum(sales_amount) as sales from sales_table
group by region
order by sum(sales_amount) desc;
```

Output:

region	sales
APAC	1210838221.4522
EU	505411517.9119
NA	473667906.4218
LATAM	10667399.6517

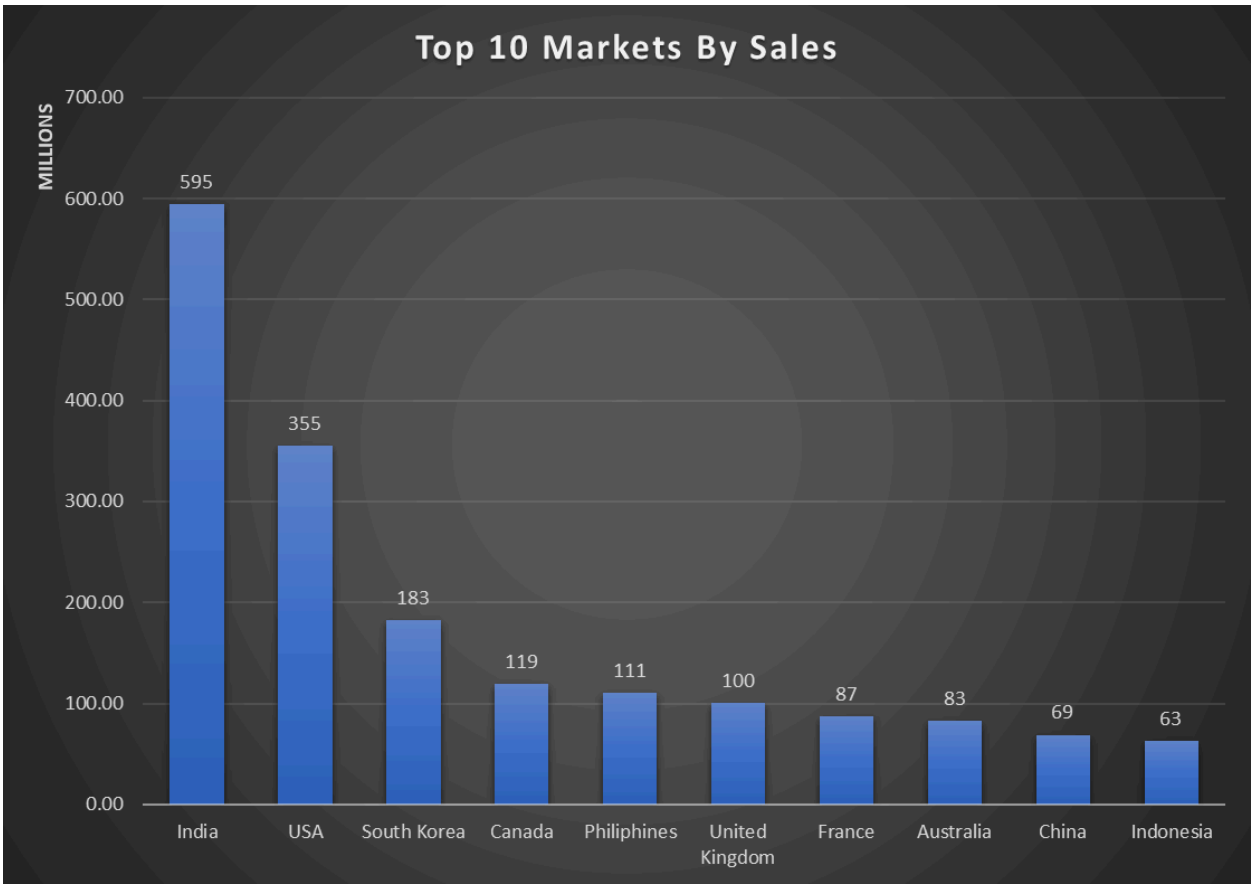


Top 10 Markets

```
select market, sum(sales_amount) as sales from sales_table
group by market
order by sum(sales_amount) desc limit 10;
```

Output:

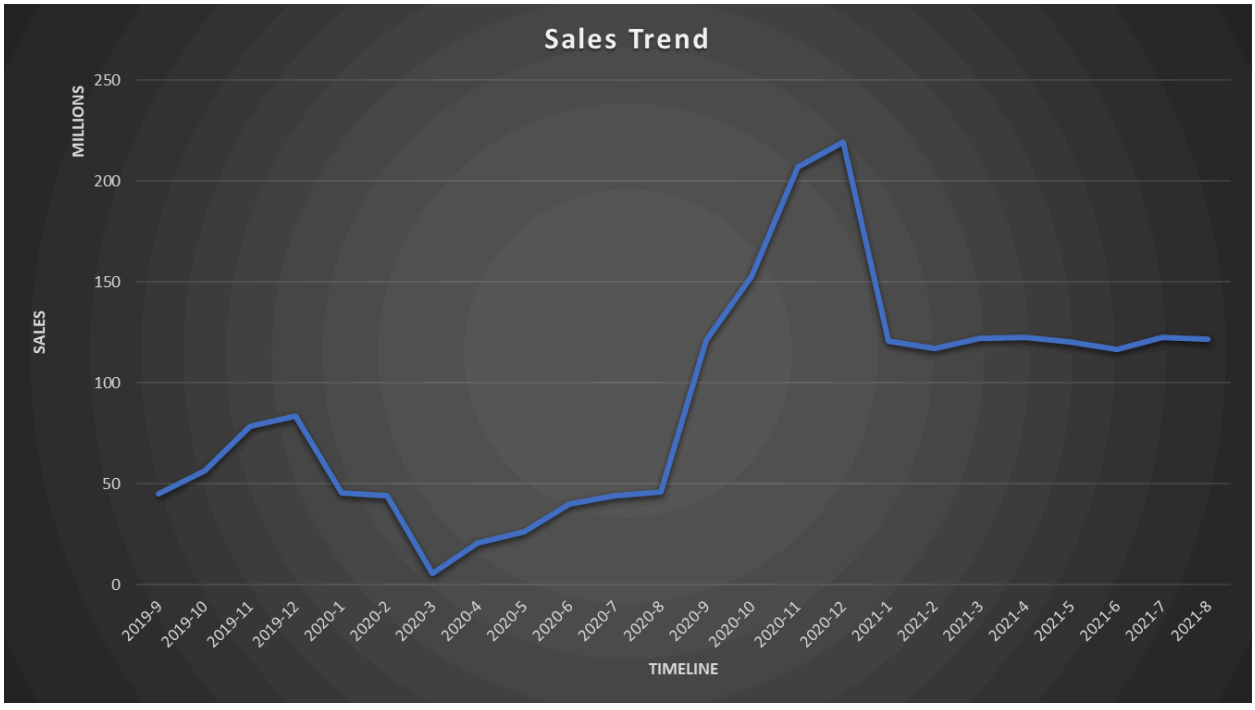
market	sales
India	594752489.9312
USA	354779274.2561
South Korea	182637580.3667
Canada	118888632.1657
Philiphines	110794209.5149
United Kingdom	100428100.2123
France	86979117.1974
Australia	82957143.4515
China	68876356.1581
Indonesia	62972061.2284



```sql
-- india, south korea, philiphines
select market,sum(sales_amount) as sales from sales_table
where region = 'APAC'
group by market
order by sum(sales_amount) desc;
```

market	sales
India	594752489.9312
South Korea	182637580.3667
Philiphines	110794209.5149
Australia	82957143.4515
China	68876356.1581
Indonesia	62972061.2284
Newzealand	34699277.2967
Pakistan	26186703.8586
Bangladesh	24710857.6066
Japan	22251542.0395

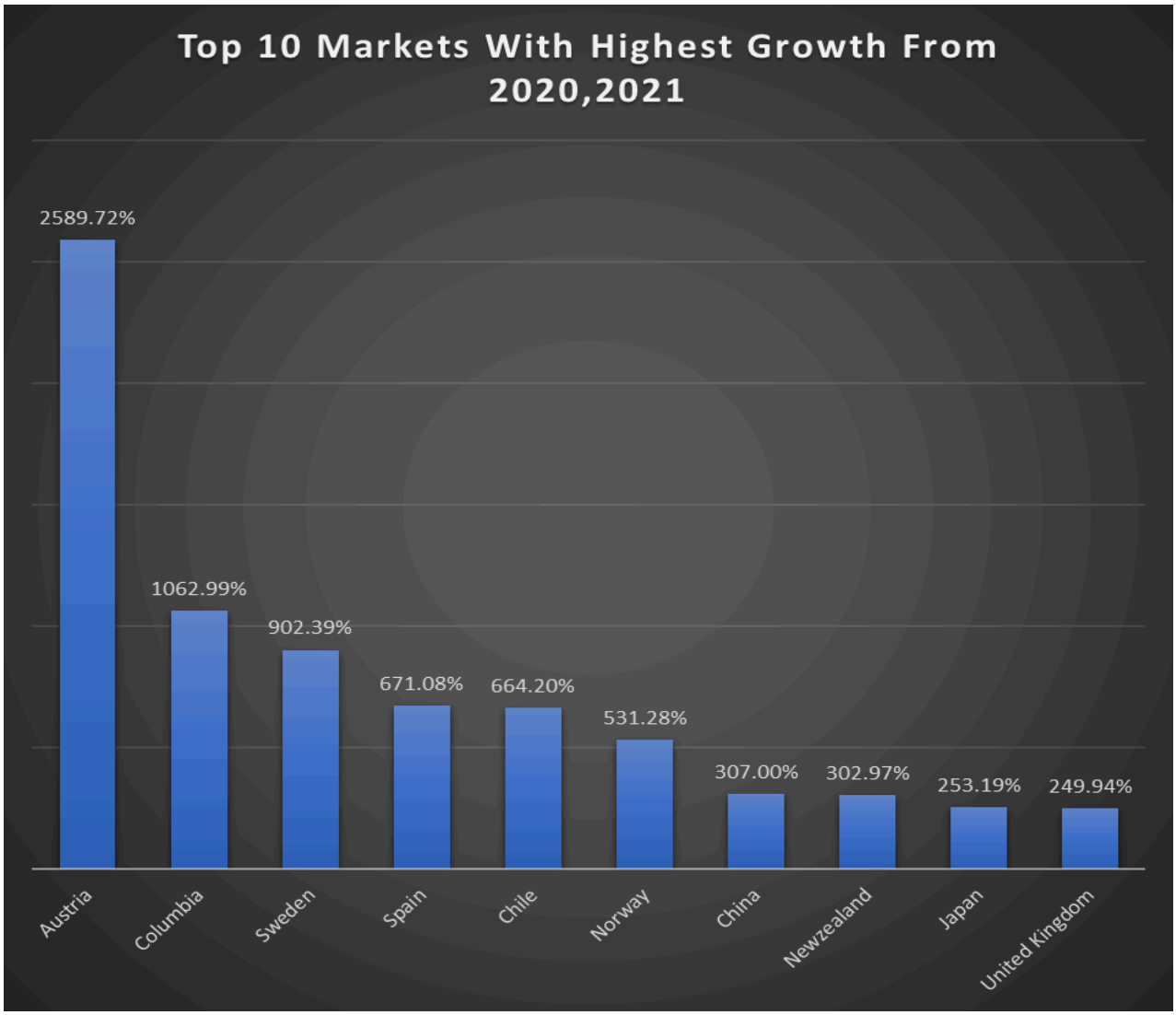
Sales Trend Across Markets



Market Wise Increase in sales

```sql
with sales_2020 as
(select market, sum(sales_amount) as sales_2020 from sales_table
where fiscal_year = 2020
group by market
),
sales_2021 as
(
select market, sum(sales_amount) as sales_2021 from sales_table
where fiscal_year = 2021
group by market
)
select t1.market,t1.sales_2020,t2.sales_2021,
t2.sales_2021 - t1.sales_2020 as sales_inc_or_dec,
round(((t2.sales_2021 - t1.sales_2020)/t1.sales_2020)*100,2) as sales_inc_or_dec_prec
from sales_2020 t1 inner join sales_2021 t2
on t1.market = t2.market
order by round(((t2.sales_2021 - t1.sales_2020)/t1.sales_2020)*100,2) desc limit 10;
```

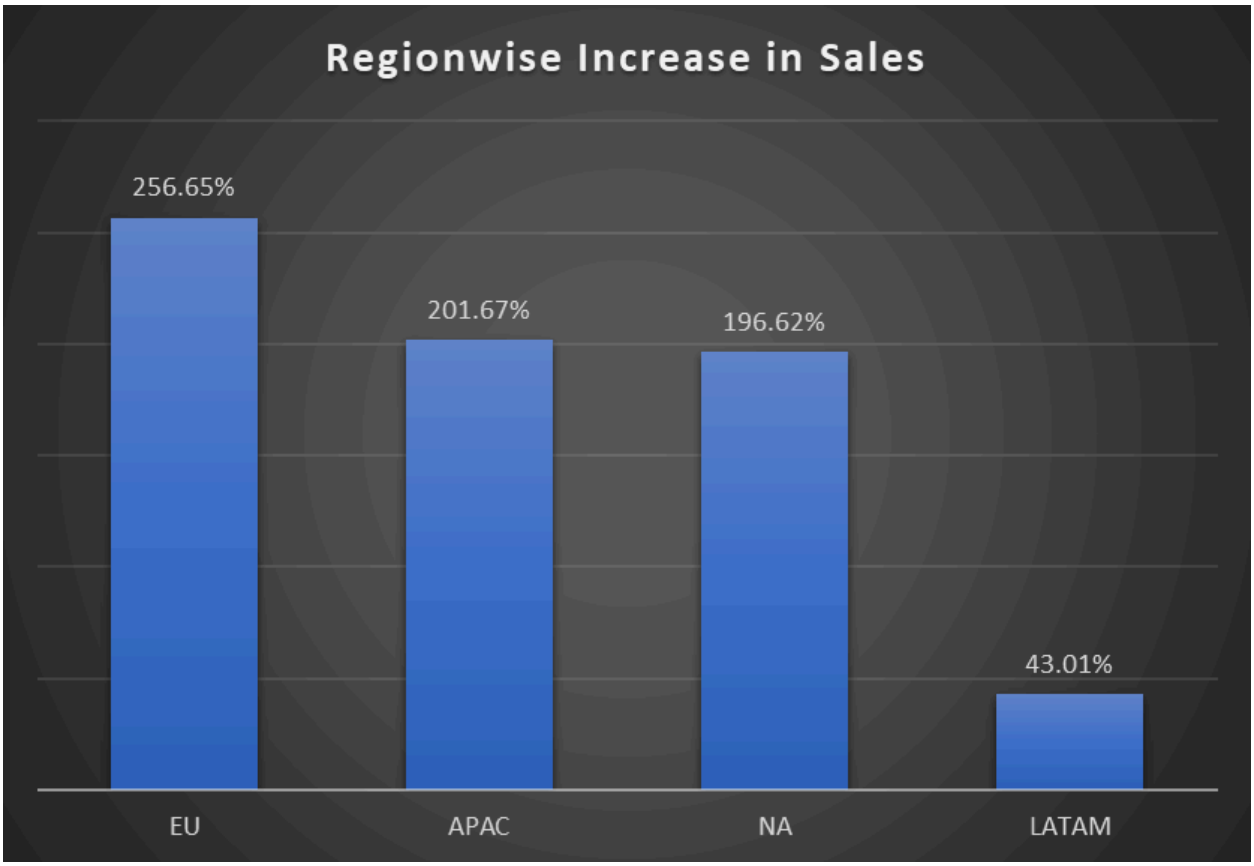
Output:



Region WIse Sales Increase

```
with regional_sales_2020 as
(select region,sum(sales_amount) as 2020_sales from sales_table where fiscal_year = 2020
group by region
),
regional_sales_2021 as
(
select region,sum(sales_amount) as 2021_sales from sales_table where fiscal_year = 2021
group by region
)
select t1.region,2020_sales,2021_sales,
2021_sales - 2020_sales as regional_sales_increase,
((2021_sales - 2020_sales)/(2020_sales)) * 100 as regional_sales_increase_perc
 from regional_sales_2020 t1 inner join regional_sales_2021 t2 on
t1.region = t2.region
order by ((2021_sales - 2020_sales)/(2020_sales)) * 100 desc;
```

Output:

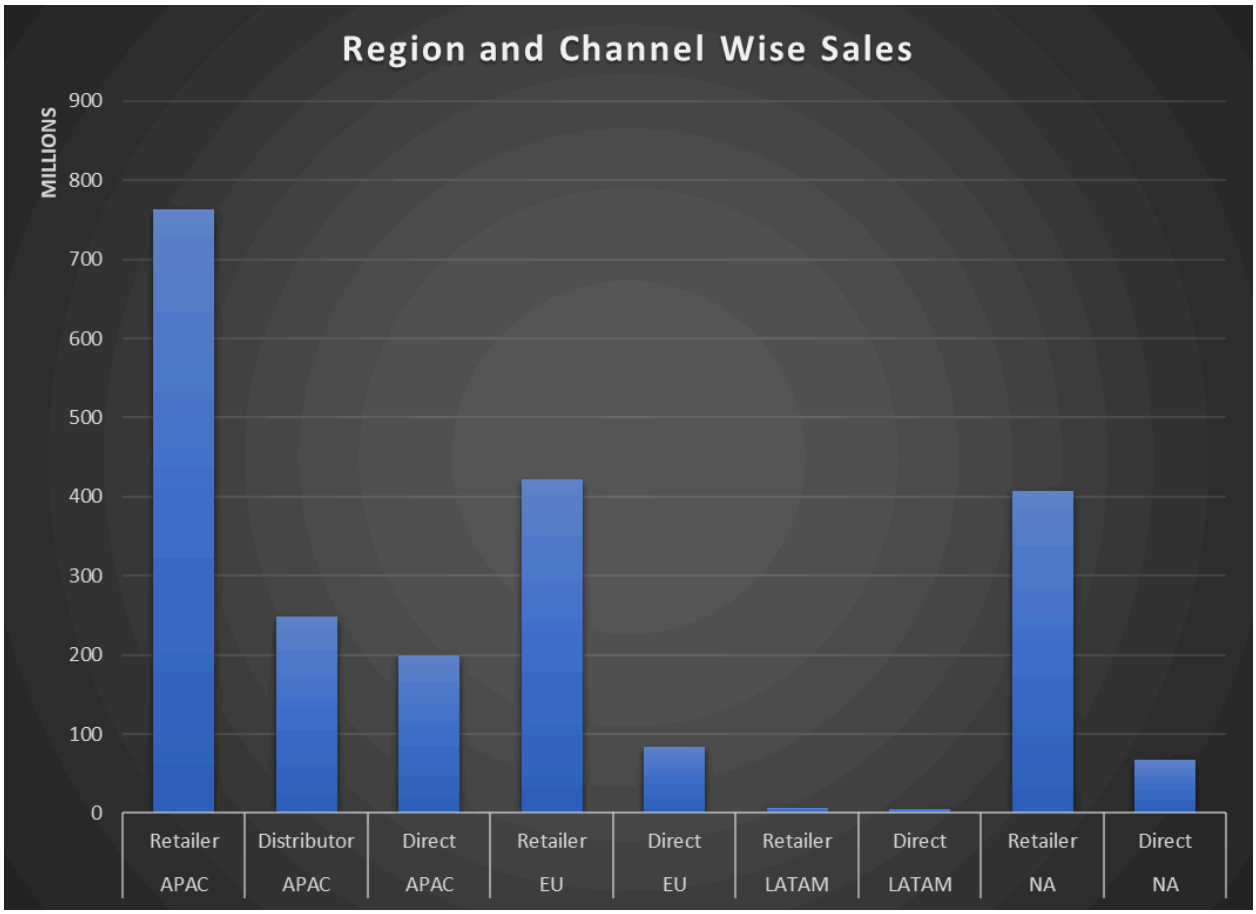


Channel and Region.

```
-- channel and region.
select region,channel,sum(sales_amount) as sales_amt from sales_table
group by region,channel
order by region asc, channel desc;
```

Output:

region	channel	sales_amt
APAC	Retailer	763572318.8344
APAC	Distributor	248465233.7650
APAC	Direct	198800668.8528
EU	Retailer	421586528.3839
EU	Direct	83824989.5280
LATAM	Retailer	6309457.9526
LATAM	Direct	4357941.6991
NA	Retailer	406687654.2340
NA	Direct	66980252.1878

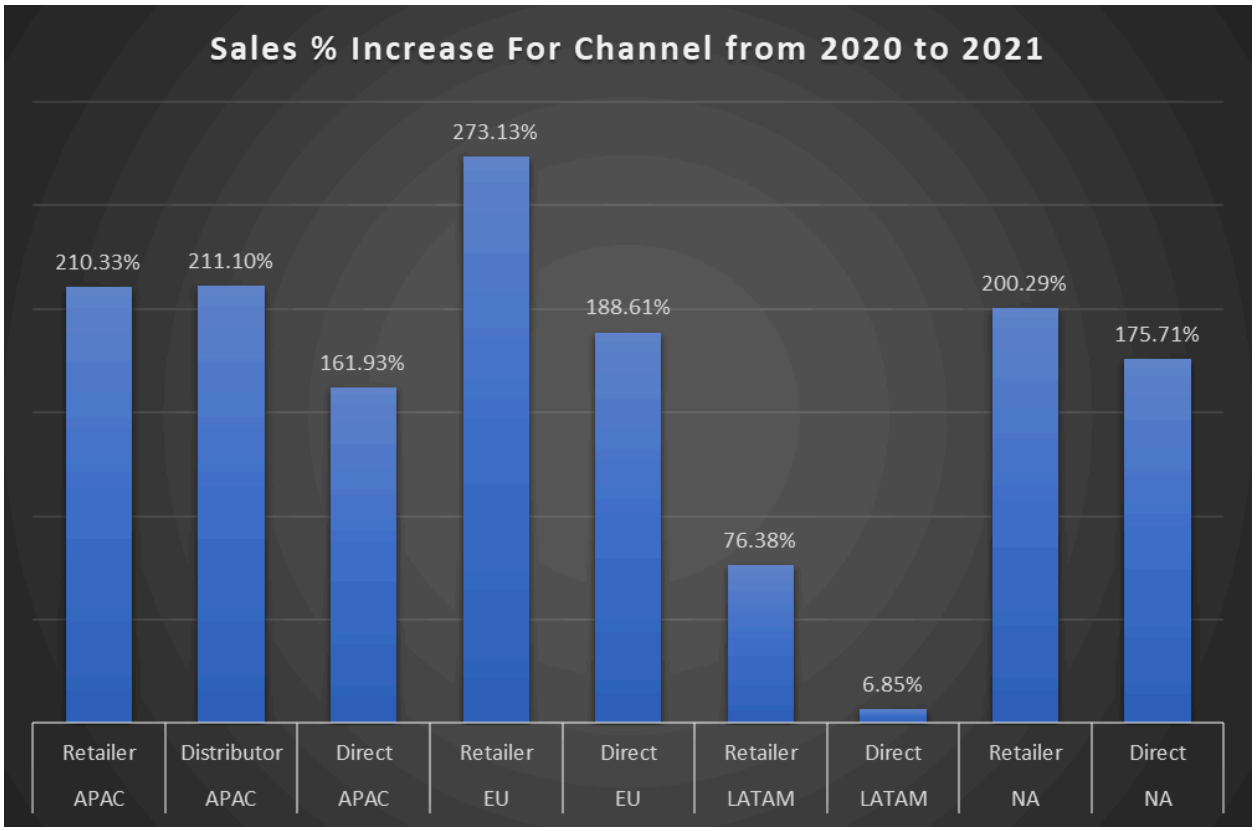


Channel and Region YoY.

```sql
-- channel and region 2020,2021
with channel_region_sales_2020 as
(select region,channel,sum(sales_amount) as sales_amt_2020 from sales_table
where fiscal_year = 2020
group by region,channel
),
channel_region_sales_2021 as
(select region,channel,sum(sales_amount) as sales_amt_2021 from sales_table
where fiscal_year = 2021
group by region,channel
)
select t1.region,t1.channel,t1.sales_amt_2020,
t2.sales_amt_2021, t2.sales_amt_2021 - t1.sales_amt_2020 as sales_inc_or_dec,
((t2.sales_amt_2021 - t1.sales_amt_2020)/(t1.sales_amt_2020)) * 100 as perc_inc_or_dec
from channel_region_sales_2020 t1 inner join channel_region_sales_2021 t2
on t1.region = t2.region and t1.channel = t2.channel
order by t1.region asc, t1.channel desc;
```

Output:

region	channel	sales_amt_2020	sales_amt_2021	sales_inc_or_dec	perc_inc_or_dec
APAC	Retailer	186085759.8748	577486558.9596	391400799.0848	210.33355768
APAC	Distributor	60439602.8302	188025630.9348	127586028.1046	211.09673481
APAC	Direct	54927215.7384	143873453.1144	88946237.3760	161.93472795
EU	Retailer	89106744.4988	332479783.8851	243373039.3863	273.12527324
EU	Direct	21570316.1370	62254673.3910	40684357.2540	188.61270737
LATAM	Retailer	2282876.4008	4026581.5518	1743705.1510	76.38193423
LATAM	Direct	2106794.9138	2251146.7853	144351.8715	6.85172869
NA	Retailer	101598938.6833	305088715.5507	203489776.8674	200.28730566
NA	Direct	17827522.8249	49152729.3629	31325206.5380	175.71261496

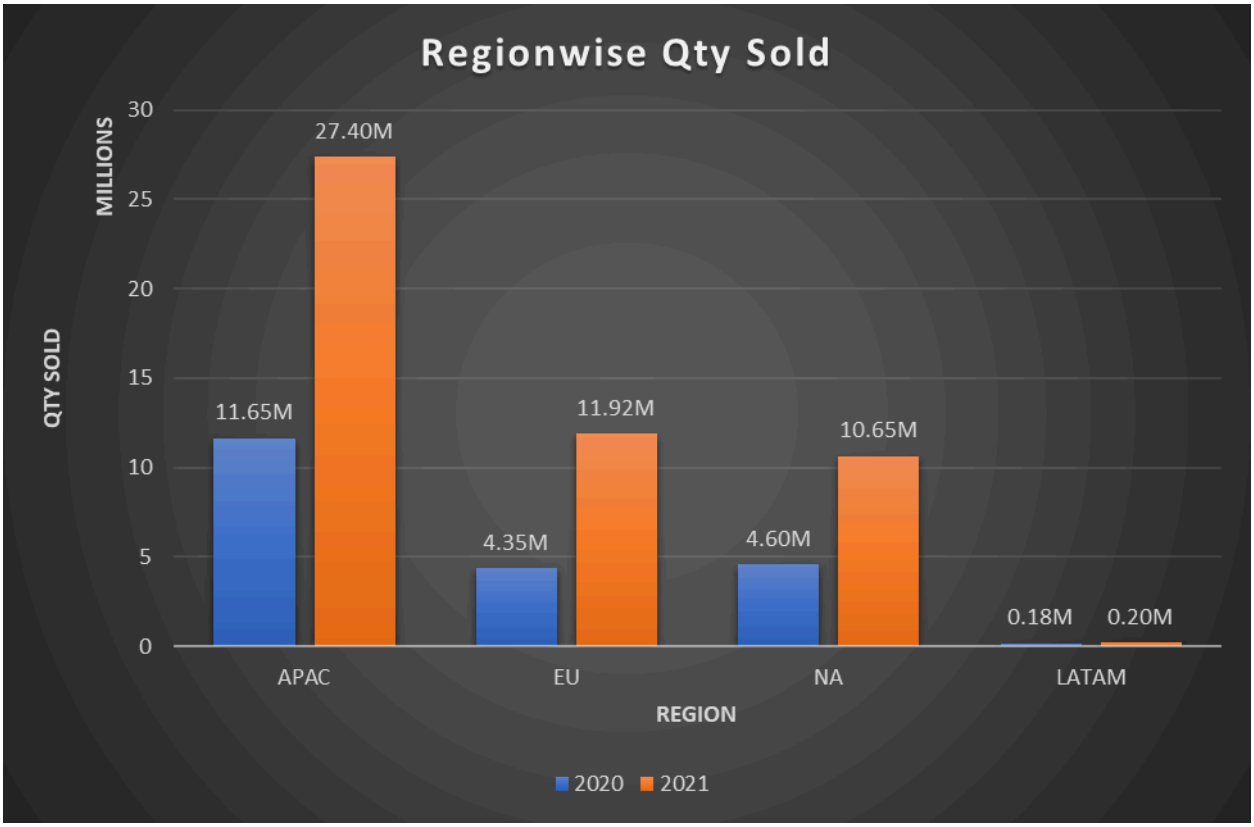


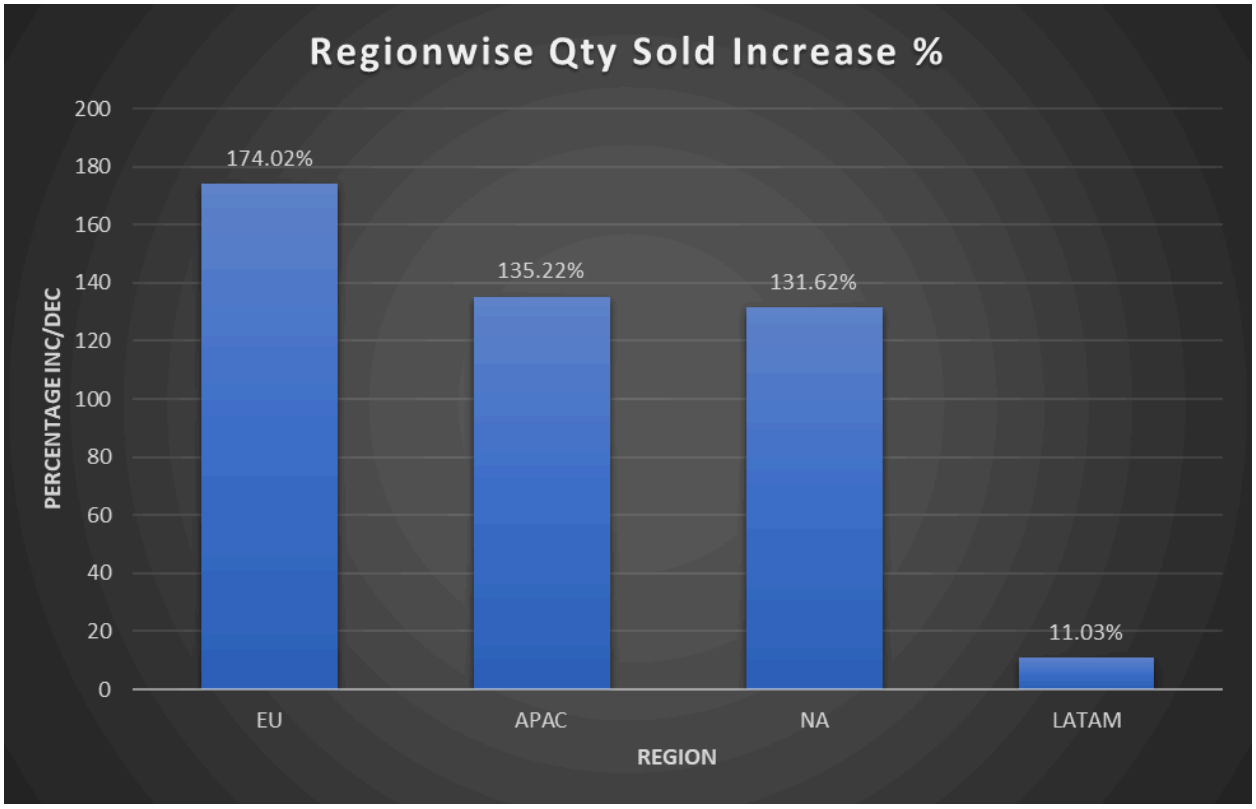
Qty Sold Analysis for region increase/dec from 2020 to 2021.

```sql
-- Qty Sold Analysis for region increase/dec from 2020 to 2021.
with region_wise_sold_qty_2020 as
(select region,sum(sold_quantity) as sold_qty_2020 from sales_table
where fiscal_year = '2020'
group by region),
region_wise_sold_qty_2021 as
(select region,sum(sold_quantity) as sold_qty_2021 from sales_table
where fiscal_year = '2021'
group by region)
select t1.region,t1.sold_qty_2020, t2.sold_qty_2021,
sold_qty_2021 - sold_qty_2020 as inc_or_dec_in_sold_qty,
((t2.sold_qty_2021 - t1.sold_qty_2020)/(t1.sold_qty_2020)) * 100 as inc_or_dec_percentage
from region_wise_sold_qty_2020 t1 inner join
region_wise_sold_qty_2021 t2 on
t1.region = t2.region;
```

Output:

region	sold_qty_2020	sold_qty_2021	inc_or_dec_in_sold_qty	inc_or_dec_percentage
APAC	11647166	27396809	15749643	135.2230
EU	4350382	11920886	7570504	174.0193
NA	4598079	10650081	6052002	131.6202
LATAM	177262	196806	19544	11.0255



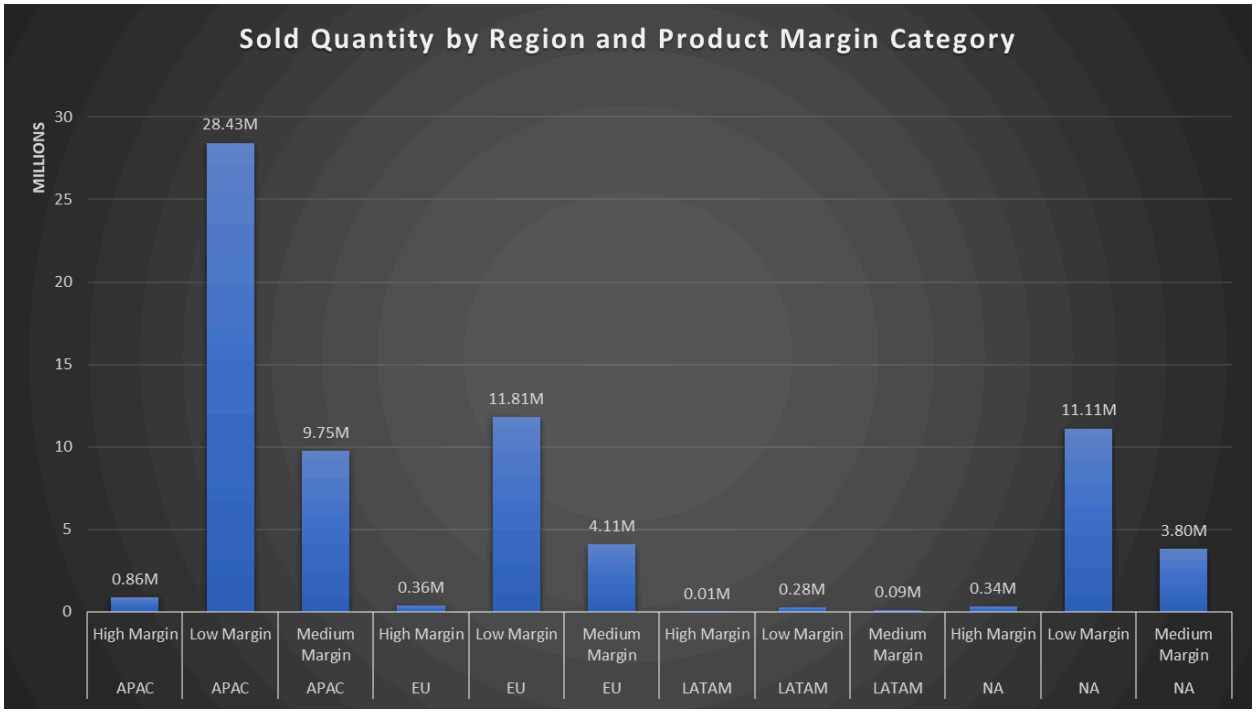


Sold Quantity by Region and Product Margin Category

```
-- region and margin category sold most.
select t1.region, t2.margin_category, sum(sold_quantity) as qty_sold from sales_table
t1 inner join pcode_margin_cat t2
on t1.product_code = t2.product_code
and t1.fiscal_year = t2.fiscal_year
group by t1.region,t2.margin_category
order by t1.region, t2.margin_category,qty_sold desc;
```

Output:

region	margin_category	qty_sold
APAC	high_margin_product	858603
APAC	low_maring_product	28434558
APAC	medium_margin_product	9750814
EU	high_margin_product	359812
EU	low_maring_product	11806213
EU	medium_margin_product	4105243
LATAM	high_margin_product	6820
LATAM	low_maring_product	279930
LATAM	medium_margin_product	87318
NA	high_margin_product	337793
NA	low_maring_product	11111564
NA	medium_margin_product	3798803

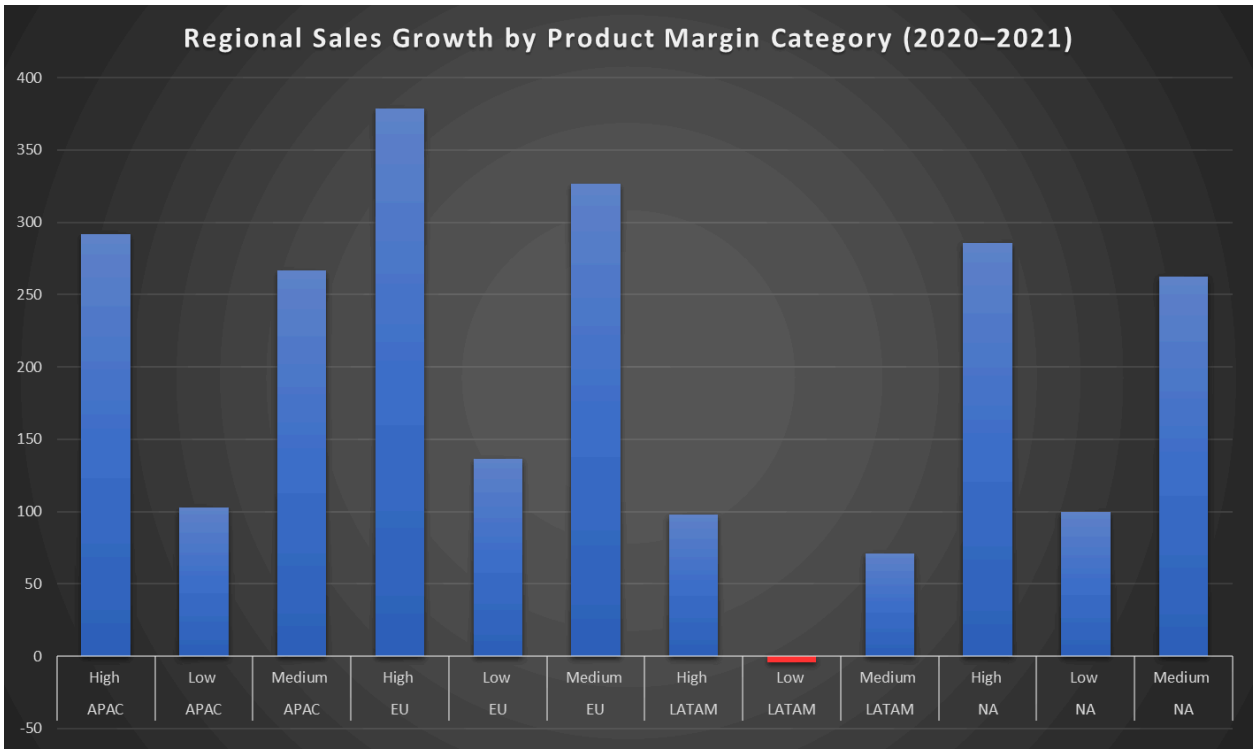


Regionwise Increase/Decrease in sales - By product margin category.

```sql
-- regionwise increase in sales of by product margin category from 2020 to 2021.

with sold_qty_2020 as
(select t1.region,t2.margin_category, sum(t1.sold_quantity) as sold_qty_2020 from
sales_table t1 inner join pcode_margin_cat t2 on
t1.product_code = t2.product_code and
t1.fiscal_year = t2.fiscal_year
where t1.fiscal_year = 2020
group by t1.region,t2.margin_category),
sold_qty_2021 as
(
select t1.region,t2.margin_category, sum(t1.sold_quantity) as sold_qty_2021 from
sales_table t1 inner join pcode_margin_cat t2 on
t1.product_code = t2.product_code and
t1.fiscal_year = t2.fiscal_year
where t1.fiscal_year = 2021
group by t1.region,t2.margin_category
)
select t1.region,t1.margin_category,
t1.sold_qty_2020,t2.sold_qty_2021,
((t2.sold_qty_2021 - t1.sold_qty_2020)/(t1.sold_qty_2020)) * 100 as perc_increase
from sold_qty_2020 t1 inner join sold_qty_2021 t2 on
t1.region = t2.region and
t1.margin_category = t2.margin_category
order by t1.region asc,t2.margin_category asc,perc_increase desc;
```

Output:

region	margin_category	sold_qty_2020	sold_qty_2021	perc_increase
APAC	high_margin_product	174482	684121	292.0869
APAC	low_maring_product	9384110	19050448	103.0075
APAC	medium_margin_product	2088574	7662240	266.8647
EU	high_margin_product	62150	297662	378.9413
EU	low_maring_product	3508880	8297333	136.4667
EU	medium_margin_product	779352	3325891	326.7508
LATAM	high_margin_product	2287	4533	98.2073
LATAM	low_maring_product	142791	137139	-3.9582
LATAM	medium_margin_product	32184	55134	71.3087
NA	high_margin_product	69546	268247	285.7116
NA	low_maring_product	3707013	7404551	99.7444
NA	medium_margin_product	821520	2977283	262.4115



Provide the list of markets in which customer "Atliq Exclusive" operates its business in the APAC region.

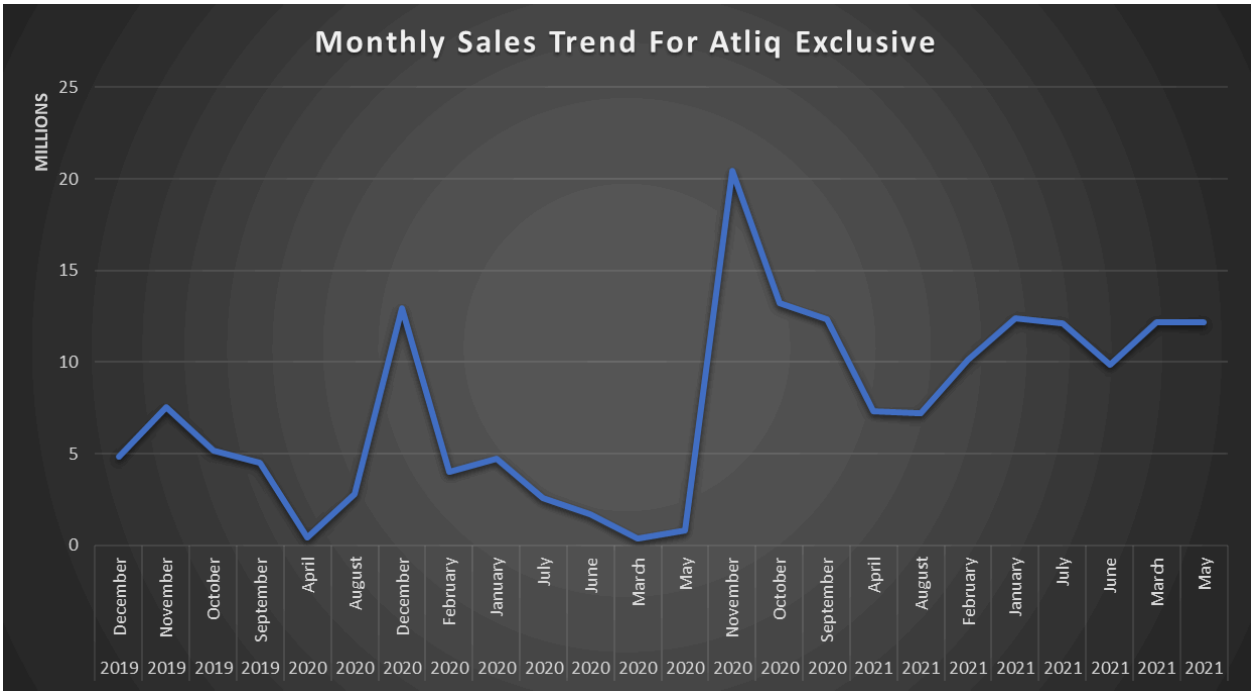
```
2
3 • select distinct market from dim_customer where customer = 'Atliq Exclusive' and region = 'APAC';
4
```

market
India
Indonesia
Japan
Philiphines
South Korea
Australia
Newzealand
Bangladesh

Monthly Sales of Atliq Exclusive

```sql
-- monthly sales of Atliq Exclusive.
select year(date),monthname(date), sum(sales_amount) as sum_sales from sales_table
where customer = 'Atliq Exclusive'
group by year(date), monthname(date)
order by year(date),monthname(date) asc;
```

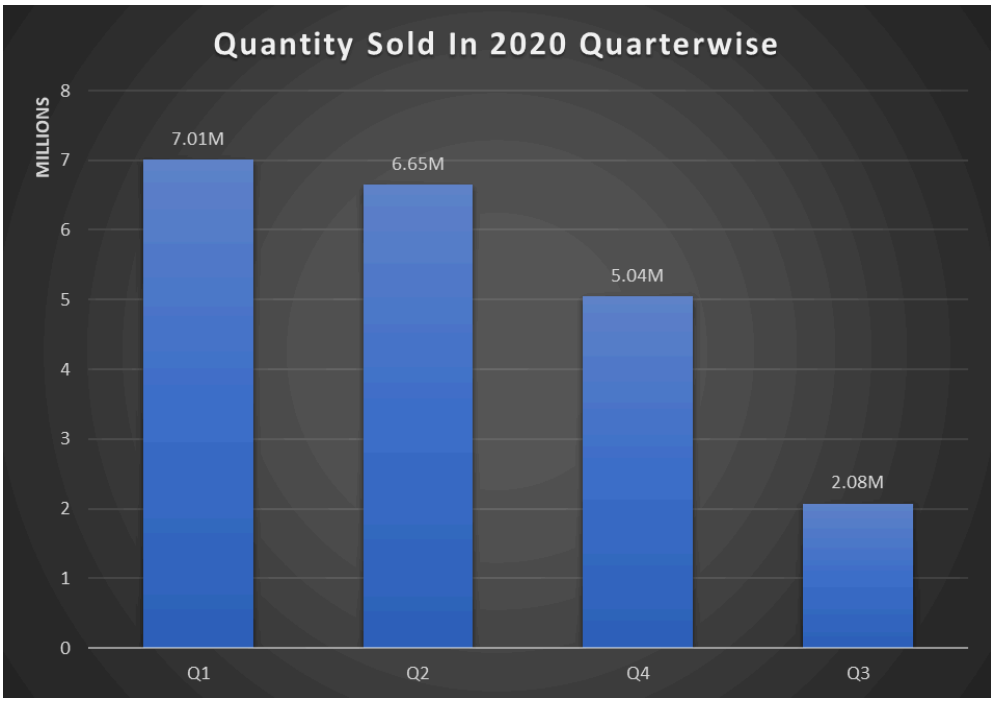
Output:



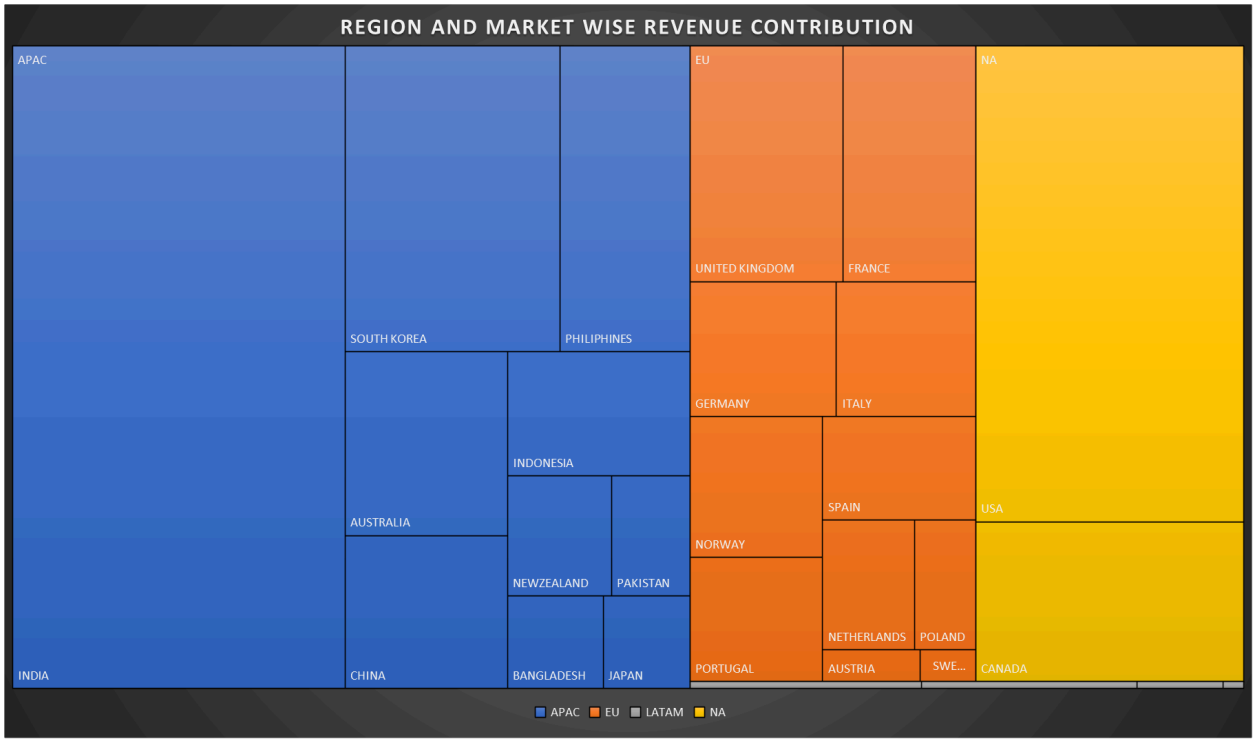
In which quarter of 2020, got the maximum total_sold_quantity?

```sql
-- In which quarter of 2020, got the maximum total_sold_quantity?
with qty_sold as
(select *,
case
    when month(date) in (9,10,11) then 'Q1'
    when month(date) in (12,1,2) then ' Q2'
    when month(date) in (3,4,5) then 'Q3'
    when month(date) in (6,7,8) then 'Q4'
end
as FY20_Quarter
from fact_sales_monthly where fiscal_year = 2020)
select FY20_Quarter,sum(sold_quantity) as qty_sold from qty_sold
group by FY20_Quarter
order by sum(sold_quantity) desc;
```

Output:



Region & markets revenue contribution (explicit view).



Insights:

APAC is the sales leader.
India and USA are the largest Markets by sales.
India, South Korea and Philippines are the biggest contributors to sales in the APAC Region.
Overall Sales are increasing YoY.
The EU region witnessed the highest sales growth from 2020 to 2021.
For the LATAM region sales grew the least among other regions.
Across all regions retailer channel leads for sales.
LATAM Region sales grew the least from 2020 to 2021. Despite giving the highest average discount regionwise. Sold Qty also grew the least for the LATAM region.
For the LATAM region the sales for lower product margin(avg selling price - avg manufacturing cost) products have reduced.

Customer Analysis

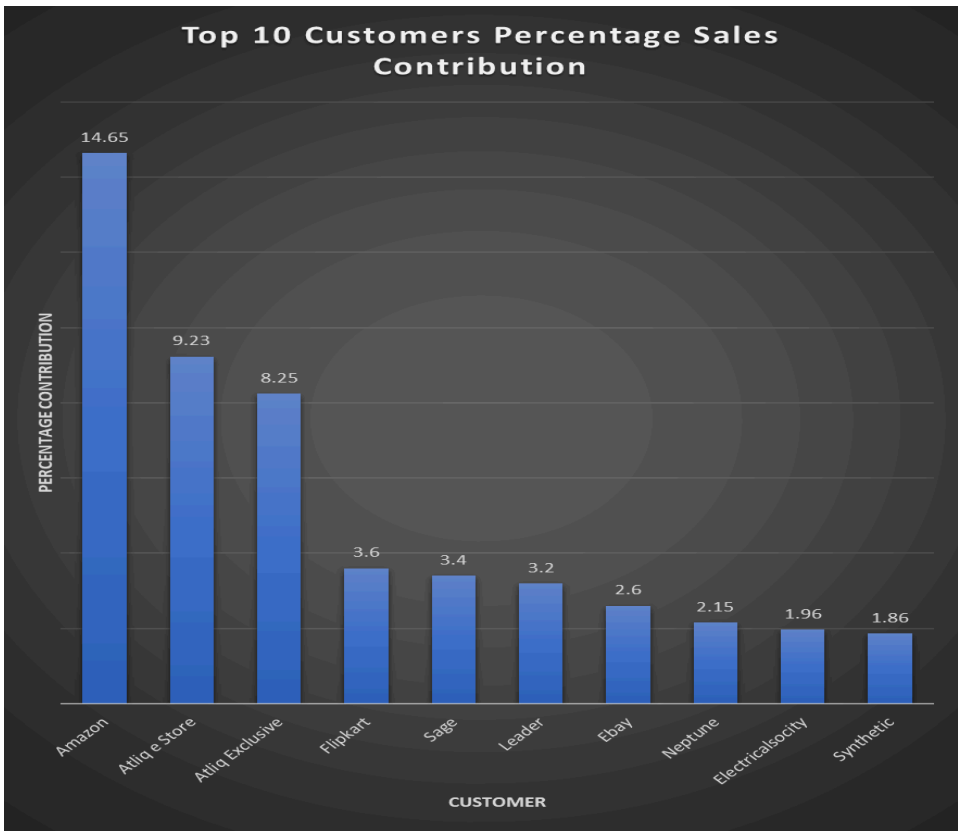
Who are our top 10 customers by gross sales, and what percentage of total sales do they contribute?

```
select customer, sum(sales_amount) as sales,
round((sum(sales_amount)/(select sum(sales_amount) from sales_table)) * 100,2) as perc_contribution
from sales_table
group by customer
order by sum(sales_amount) desc;
```

customer	sales	perc_contribution
Amazon	322328100.6506	14.65
Atliq e Store	203057127.0996	9.23
Atliq Exclusive	181525725.4333	8.25
Flipkart	79239858.0464	3.60
Sage	74773715.8104	3.40
Leader	70417435.9929	3.20
Ebay	57264631.7732	2.60
Neptune	47349601.5651	2.15
Electricalsocity	43228715.8260	1.96
Synthetic	41020748.8062	1.86

Output:

Top 10 customers constitute a total of 50 % of total sales



Which markets/customers have shown the highest sales growth vs decline over the last 2 years?

```
with customer_sales_2020 as
(select customer,sum(sales_amount) as 2020_sales from sales_table where fiscal_year = 2020
group by customer
),
customer_sales_2021 as
(
select customer,sum(sales_amount) as 2021_sales from sales_table where fiscal_year = 2021
group by customer
)
select t1.customer,2020_sales,2021_sales,
2021_sales - 2020_sales as customer_sales_increase,
((2021_sales - 2020_sales)/(2020_sales)) * 100 as customer_sales_increase_perc
 from customer_sales_2020 t1 inner join customer_sales_2021 t2 on
t1.customer = t2.customer
order by ((2021_sales - 2020_sales)/(2020_sales)) * 100 desc limit 10;
```

customer	2020_sales	2021_sales	customer_sales_increase	customer_sales_increase_perc
Nova	34864.8734	1075689.3179	1040824.4445	2985.30969139
Integration Stores	398854.1234	4383207.1628	3984353.0394	998.94994326
Chiptec	1064793.7359	8427158.3601	7362364.6242	691.43575661
Electricalsbea Stores	501825.8737	2724215.7330	2222389.8593	442.86075624
Boulanger	2267309.8090	12092009.6084	9824699.7994	433.31968840
Elite	2169999.9250	10210876.5308	8040876.6058	370.54732183
Neptune	8521078.6705	38828522.8946	30307444.2241	355.67614613
Atlas Stores	1747510.2707	7911861.4192	6164351.1485	352.75049605
Euronics	2370173.4993	10679087.9322	8308914.4329	350.56144351
UniEuro	4221618.7223	18586795.4471	14365176.7248	340.27650695

Which customers generate the highest net profit (gross sales – discounts – manufacturing cost) and which ones are unprofitable?

```
select customer_code,customer,discount_category,
avg(net_profit) as average_net_profit,
avg(profit_percentage) as avg_profit_percentage
from customer_analysis_table
group by customer_code, customer,discount_category
order by discount_category asc , avg(profit_percentage) desc,avg(net_profit) desc;
```

Not making a loss by giving discounts to customers.

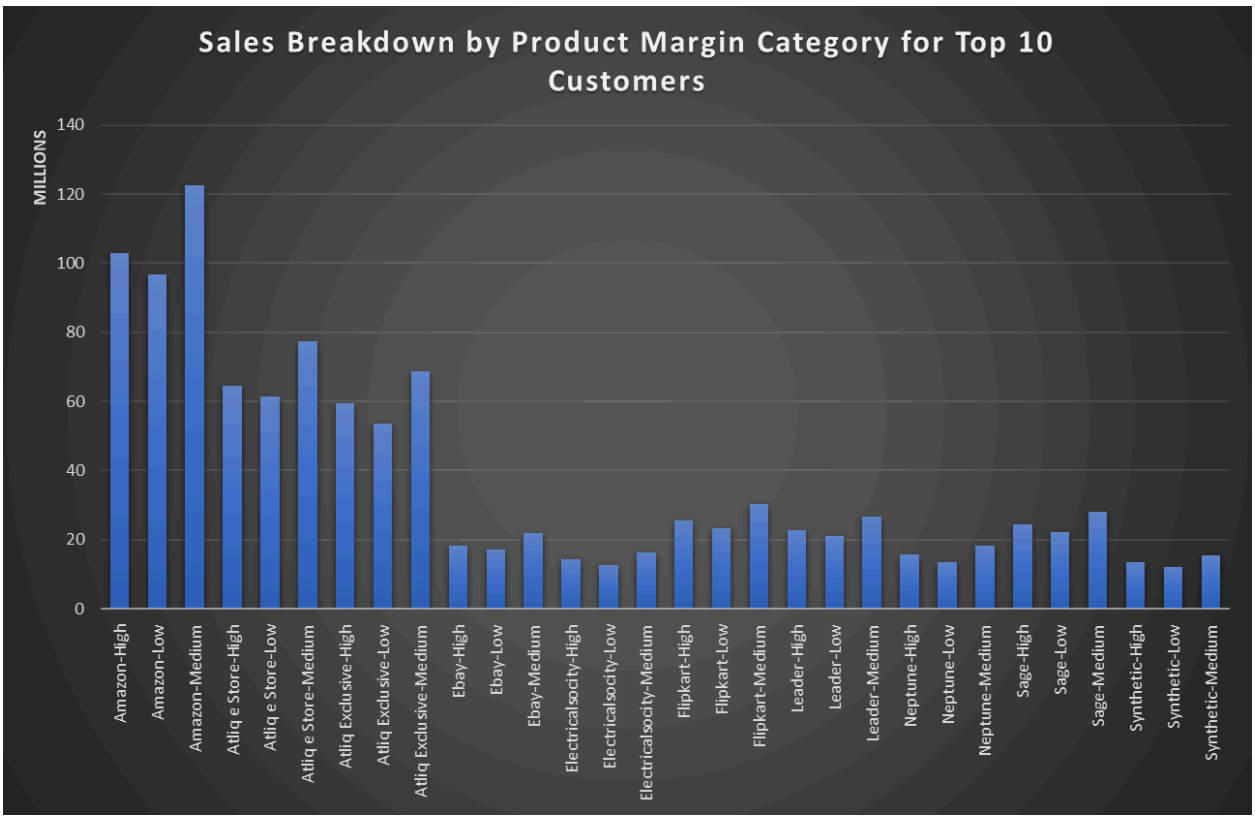
Top 10 Customer and product margin category - by sales amount analysis.

```
with temp_table as (
select distinct customer from top_10_customer_sales_table group by customer
order by sum(sales_amount) desc limit 10
)
select t1.customer,t2.margin_category,sum(sales_amount) as sum_sales from sales_table t1
inner join pcode_margin_cat t2 on
t1.product_code = t2.product_code
and t1.fiscal_year = t2.fiscal_year
where t1.customer in (select temp_table.customer from temp_table)
group by t1.customer,t2.margin_category;
```

Output:

customer	margin_category	sum_sales
Amazon	high_margin_product	102967681.257
Amazon	low_maring_product	96788958.4416
Amazon	medium_margin_product	122571460.951
Atliq e Store	high_margin_product	64355396.8607
Atliq e Store	low_maring_product	61319885.8423
Atliq e Store	medium_margin_product	77381844.3966
Atliq Exclusive	high_margin_product	59327021.0598
Atliq Exclusive	low_maring_product	53562132.7716
Atliq Exclusive	medium_margin_product	68636571.6019
Ebay	high_margin_product	18252460.3028
Ebay	low_maring_product	17175257.7720
Ebay	medium_margin_product	21836913.6984
Electricalsocity	high_margin_product	14263942.5777
Electricalsocity	low_maring_product	12637822.3883
Electricalsocity	medium_margin_product	16326950.8600
Flipkart	high_margin_product	25585824.1120
Flipkart	low_maring_product	23407216.2856
Flipkart	medium_margin_product	30246817.6488
Leader	high_margin_product	22697746.7535
Leader	low_maring_product	20949710.8801
Leader	medium_margin_product	26769978.3593

	Neptune	high_margin_product	15698283.9176
	Neptune	low_maring_product	13466396.2893
	Neptune	medium_margin_product	18184921.3582
	Sage	high_margin_product	24449154.6936
	Sage	low_maring_product	22162268.6106
	Sage	medium_margin_product	28162292.5062
	Synthetic	high_margin_product	13413623.7724
	Synthetic	low_maring_product	12108759.7035
	Synthetic	medium_margin_product	15498365.3303



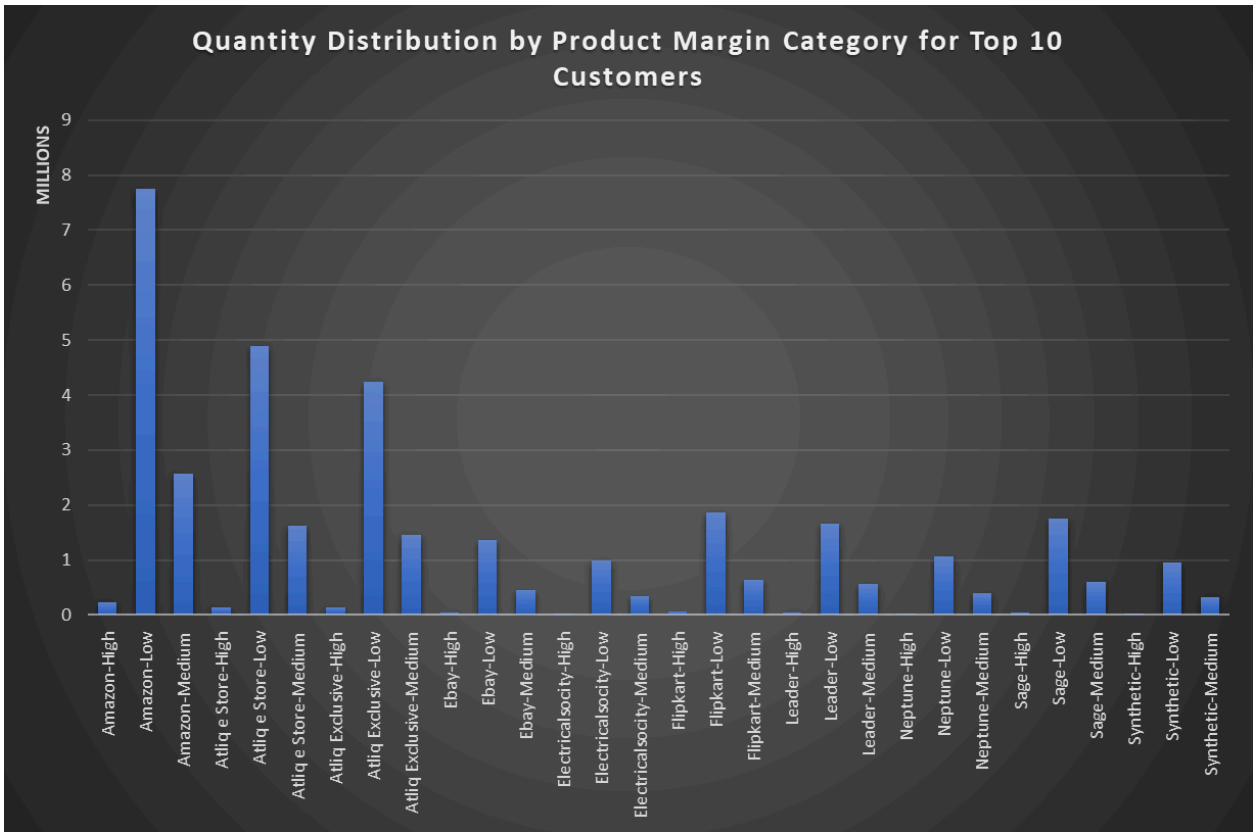
Top 10 Customer and product margin category - by product quantity sold.

```
with temp_table as (
select distinct customer from top_10_customer_sales_table group by customer
order by sum(sold_quantity) desc limit 10
)
select t1.customer,t2.margin_category,sum(sold_quantity) as sold_quantity from sales_table t1
inner join pcode_margin_cat t2 on
t1.product_code = t2.product_code
and t1.fiscal_year = t2.fiscal_year
where t1.customer in (select temp_table.customer from temp_table)
group by t1.customer,t2.margin_category;
```

Output:

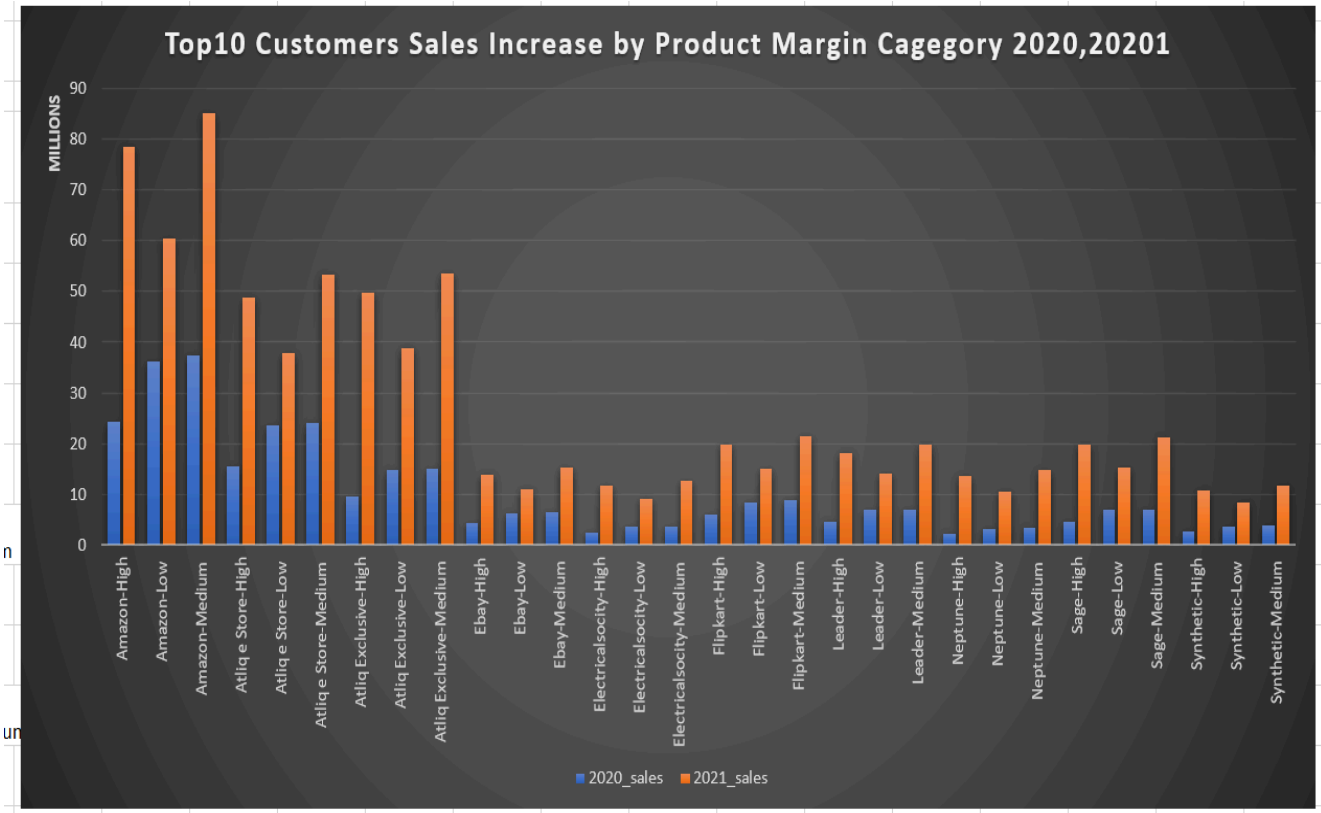
customer	margin_category	sold_quantity
Amazon	high_margin_product	225820
Amazon	low_maring_product	7745350
Amazon	medium_margin_product	2576292
Atliq e Store	high_margin_product	141286
Atliq e Store	low_maring_product	4899735
Atliq e Store	medium_margin_product	1619829
Atliq Exclusive	high_margin_product	129509
Atliq Exclusive	low_maring_product	4248257
Atliq Exclusive	medium_margin_product	1460213
Ebay	high_margin_product	39970
Ebay	low_maring_product	1366470
Ebay	medium_margin_product	455355
Electricalsocity	high_margin_product	31225
Electricalsocity	low_maring_product	999434
Electricalsocity	medium_margin_product	346891
Flipkart	high_margin_product	56123
Flipkart	low_maring_product	1864190
Flipkart	medium_margin_product	634881
Leader	high_margin_product	49510
Leader	low_maring_product	1653978
Leader	medium_margin_product	566888

Leader	medium_margin_product	566888
Neptune	high_margin_product	34331
Neptune	low_maring_product	1058654
Neptune	medium_margin_product	389312
Sage	high_margin_product	53191
Sage	low_maring_product	1753831
Sage	medium_margin_product	593626
Synthetic	high_margin_product	29274
Synthetic	low_maring_product	961773
Synthetic	medium_margin_product	323209



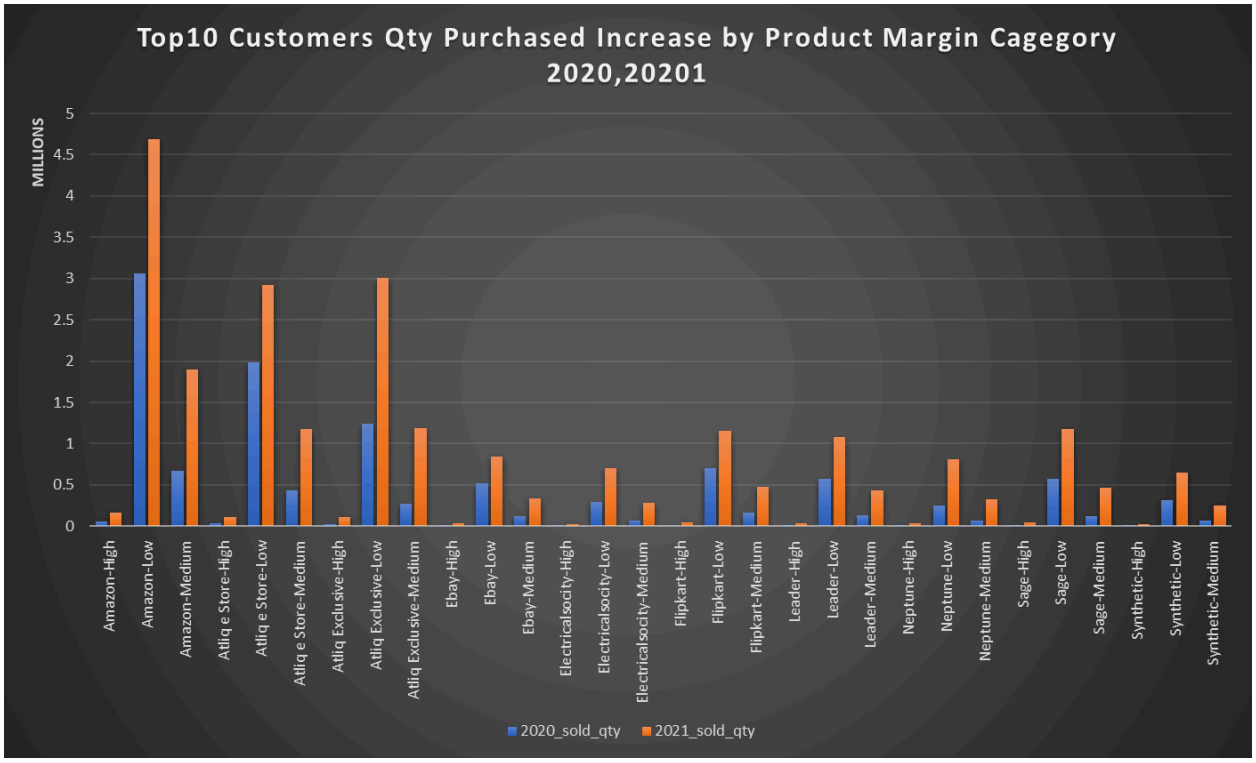
Top 10 Customer and product margin category - by sales amount YoY.

```sql
-- customer and product category margin analysis -- by sales amount yoy.
with sales_amt_2020 as
(with temp_table as (
select distinct customer from top_10_customer_sales_table group by customer
order by sum(sales_amount) desc limit 10
)
select t1.customer,t2.margin_category,sum(sales_amount) as sum_sales_2020 from sales_table t1
inner join pcode_margin_cat t2 on
t1.product_code = t2.product_code
and t1.fiscal_year = t2.fiscal_year
where t1.customer in (select temp_table.customer from temp_table) and t1.fiscal_year = 2020
group by t1.customer,t2.margin_category
),
sales_amt_2021 as
(with temp_table as (
select distinct customer from top_10_customer_sales_table group by customer
order by sum(sales_amount) desc limit 10
)
select t1.customer,t2.margin_category,sum(sales_amount) as sum_sales_2021 from sales_table t1
inner join pcode_margin_cat t2 on
t1.product_code = t2.product_code
and t1.fiscal_year = t2.fiscal_year
where t1.customer in (select temp_table.customer from temp_table) and t1.fiscal_year = 2021
group by t1.customer,t2.margin_category
)
select t1.customer,t1.margin_category,t1.sum_sales_2020,t2.sum_sales_2021 from sales_amt_2020 t1
inner join sales_amt_2021 t2
on t1.customer = t2.customer and
t1.margin_category = t2.margin_category;
```


Top10 Customers Sales Increase by Product Margin Cagegory 2020,20201
MILLIONS
90
80
70
60
50
40
30
20
10
0
Amazon-High
Amazon-Low
Amazon-Medium
Atliq e Store-High
Atliq e Store-Low
Atliq e Store-Medium
Atliq Exclusive-High
Atliq Exclusive-Low
Atliq Exclusive-Medium
Ebay-High
Ebay-Low
Ebay-Medium
Electricalsocity-High
Electricalsocity-Low
Electricalsocity-Medium
Flipkart-High
Flipkart-Low
Flipkart-Medium
Leader-High
Leader-Low
Leader-Medium
Neptune-High
Neptune-Low
Neptune-Medium
Sage-High
Sage-Low
Sage-Medium
Synthetic-High
Synthetic-Low
Synthetic-Medium
2020_sales
2021_sales

Top 10 Customer and product margin category - by qty sold YoY.

```sql
-- customer and product category margin analysis -- first by sold qty yoy.
with sold_quantity_2020 as
(with temp_table as (
select distinct customer from top_10_customer_sales_table group by customer
order by sum(sales_amount) desc limit 10
)
select t1.customer,t2.margin_category,sum(sold_quantity) as sold_quantity_2020 from sales_table t1
inner join pcode_margin_cat t2 on
t1.product_code = t2.product_code
and t1.fiscal_year = t2.fiscal_year
where t1.customer in (select temp_table.customer from temp_table) and t1.fiscal_year = 2020
group by t1.customer,t2.margin_category
),
sold_quantity_2021 as
(with temp_table as (
select distinct customer from top_10_customer_sales_table group by customer
order by sum(sales_amount) desc limit 10

)
select t1.customer,t2.margin_category,sum(sold_quantity) as sold_quantity_2021 from sales_table t1
inner join pcode_margin_cat t2 on
t1.product_code = t2.product_code
and t1.fiscal_year = t2.fiscal_year
where t1.customer in (select temp_table.customer from temp_table) and t1.fiscal_year = 2021
group by t1.customer,t2.margin_category
)
select t1.customer,t1.margin_category,t1.sold_quantity_2020,t2.sold_quantity_2021 from sold_quantity_2020 t1
inner join sold_quantity_2021 t2
on t1.customer = t2.customer and
t1.margin_category = t2.margin_category;
```


Top10 Customers Qty Purchased Increase by Product Margin Cagegory 2020,20201
MILLIONS
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
Amazon-High
Amazon-Low
Amazon-Medium
Atliq e Store-High
Atliq e Store-Low
Atliq e Store-Medium
Atliq Exclusive-High
Atliq Exclusive-Low
Atliq Exclusive-Medium
Ebay-High
Ebay-Low
Ebay-Medium
Electricalsocity-High
Electricalsocity-Low
Electricalsocity-Medium
Flipkart-High
Flipkart-Low
Flipkart-Medium
Leader-High
Leader-Low
Leader-Medium
Neptune-High
Neptune-Low
Neptune-Medium
Sage-High
Sage-Low
Sage-Medium
Synthetic-High
Synthetic-Low
Synthetic-Medium
2020_sold_qty
2021_sold_qty

Insights

Top 10 customers contribute to approximately 50% of total sales.

Top 10 customers are Amazon,Atliq e Store,Atliq Exclusive,Ebay, Electricalsocity, Flipkart,Leader,Neptune,Sage,Synthetic.

The top 10 customers may be buying the most in low margin categories, but they are spending the most in high and medium margin categories—this is where the profit lies.

High and Medium margin products dominate in revenue, despite lower purchase quantities.

Amazon, Atliq e Store, and Atliq Exclusive especially generate significant revenue from medium and high margin products.

Low margin products, although purchased in large quantities, contribute less to total sales compared to medium and high margin products

Low margin products are being bought in bulk by nearly all top 10 customers.

Medium and high margin products have significantly lower volumes, even from the same customers who generate large revenue from them.

Amazon is buying a very high quantity of low margin goods, while buying relatively lower quantities of medium and high margin products that still generate large revenue.

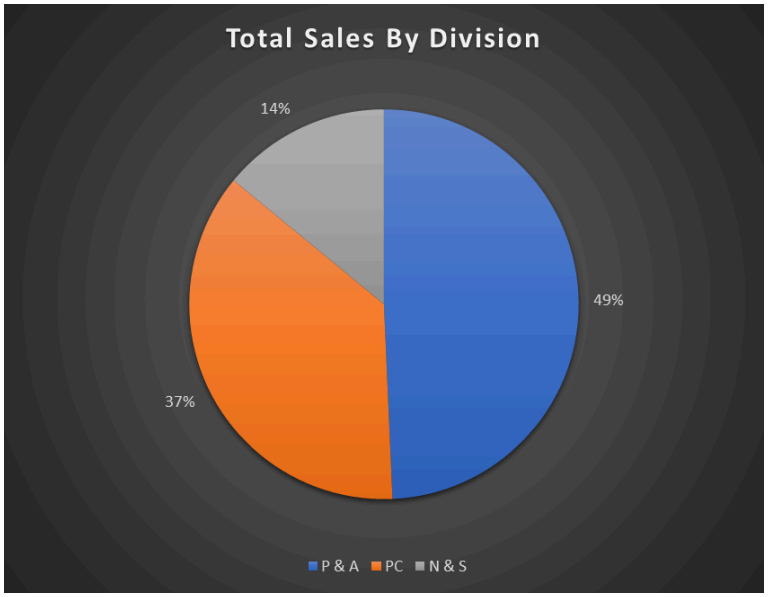
Product, Division and Segment Analysis

Which product divisions (PC, Peripherals & Accessories, Network & Storage) are driving the most revenue and profit margins?

```
-- P & A contribute highest to the revenue.
select division, sum(sales_amount) as sales_amt from sales_table
group by division
order by sum(sales_amount) desc;
```

Output:

division	sales_amt
P & A	1084927720.6001
PC	806774954.8502
N & S	308882369.9873



Get the Top 3 products in each division that have a high total_sold_quantity in the fiscal_year 2021?

```
with temp_table as
(select
t1.division,t1.product_code,t1.product,t1.variant,
sum(t2.sold_quantity) as qty_sold,
rank() over(partition by division order by sum(t2.sold_quantity) desc) as `rank`
from dim_product t1 inner join fact_sales_monthly t2
on t1.product_code = t2.product_code
where fiscal_year = 2021
group by t1.division,t1.product_code,t1.product,t1.variant)
select division,product,qty_sold from temp_table where `rank` < 4;
```

Output:

division	product	qty_sold
N & S	AQ Pen Drive 2 IN 1	701373
N & S	AQ Pen Drive DRC	688003
N & S	AQ Pen Drive DRC	676245
P & A	AQ Gamers Ms	428498
P & A	AQ Maxima Ms	419865
P & A	AQ Maxima Ms	419471
PC	AQ Digit	17434
PC	AQ Velocity	17280
PC	AQ Digit	17275

Which Segment had the most increase in unique products in 2021 vs 2022?

```
-- Which segment had the most increase in unique products in 2021 vs 2020?
with t1 as
(select segment,count(distinct product_code) as product_count_2020
from unique_products_data where fiscal_year = 2020
group by segment),
t2 as
(select segment,count(distinct product_code) as product_count_2021
from unique_products_data where fiscal_year = 2021
group by segment)
select t1.*,t2.product_count_2021,
product_count_2021 - product_count_2020 as difference
from t1 inner join t2 on t1.segment = t2.segment
order by product_count_2021 - product_count_2020 desc;
```

Output:

segment	product_count_2020	product_count_2021	difference
Accessories	69	103	34
Notebook	92	108	16
Peripherals	59	75	16
Desktop	7	22	15
Storage	12	17	5
Networking	6	9	3

Which divisions/segments are showing declining average selling prices (ASP) year-over-year?

Average prices haven't reduced for any segment particularly, need to focus on the product variant to check for decline in average selling prices.

```
-- Which divisions/segments are showing declining average selling prices (ASP) year-over-year?
with avg_segment_price_2020 as
(select segment,avg(gross_price) as avg_selling_price_2020 from sales_table
where fiscal_year = 2020
group by segment),
avg_segment_price_2021 as
(
select segment,avg(gross_price) as avg_selling_price_2021 from sales_table
where fiscal_year = 2021
group by segment
)
select t1.segment,t1.avg_selling_price_2020,t2.avg_selling_price_2021,t2.avg_selling_price_2021 - t1.avg_selling_price_2020 as inc_or_decrease
 from avg_segment_price_2020 t1 inner join avg_segment_price_2021 t2 on
t1.segment = t2.segment;
```

Output:

segment	avg_selling_price_2020	avg_selling_price_2021	inc_or_decrease
Peripherals	49.32645646	57.22090895	7.89445249
Accessories	12.41111998	16.79917607	4.38805609
Notebook	386.26763634	442.07105241	55.80341607
Desktop	724.84979077	752.77499440	27.92520363
Storage	14.59874374	15.18884706	0.59010332
Networking	30.98528818	35.73341111	4.74812293

What is the percentage of unique product increase in 2021 vs. 2020?

```
create view unique_products_data as
(select t1.*,t2.fiscal_year,t2.customer_code,t2.date from dim_product t1 inner join fact_sales_monthly t2 on
t1.product_code = t2.product_code);
```

```
with temp_table as
(select
(select count(distinct product_code) from unique_products_data where fiscal_year = 2020 group by fiscal_year) as unique_products_20
(select count(distinct product_code) from unique_products_data where fiscal_year = 2021 group by fiscal_year) as unique_products_20
)
select unique_products_2020, unique_products_2021,
round((((unique_products_2021 - unique_products_2020)/unique_products_2020) * 100),2) as percentage_chg
from temp_table;
```

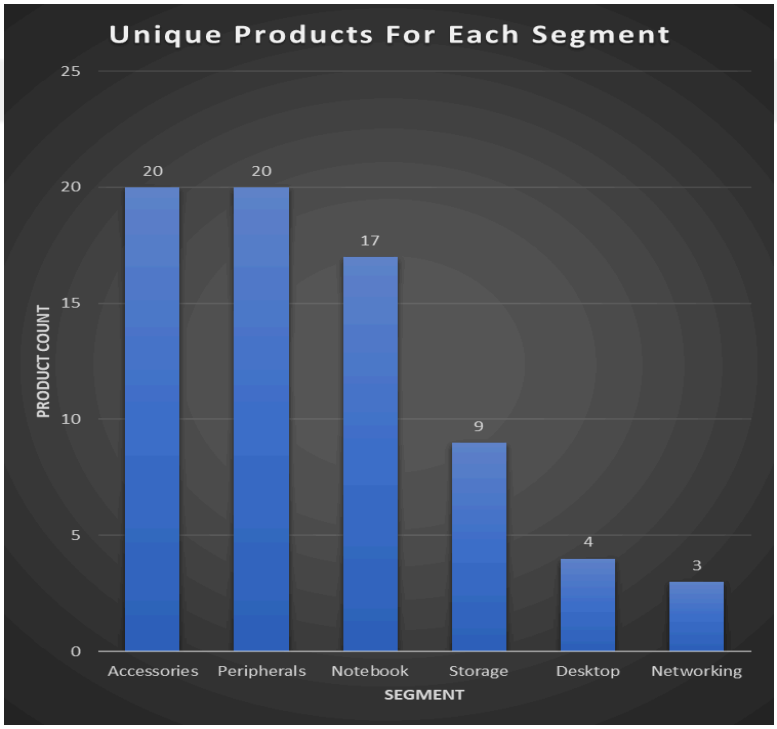
Output:

	unique_products_2020	unique_products_2021	percentage_chg
▶	245	334	36.33

Provide a report with all the unique product counts for each segment and sort them in descending order of product counts.

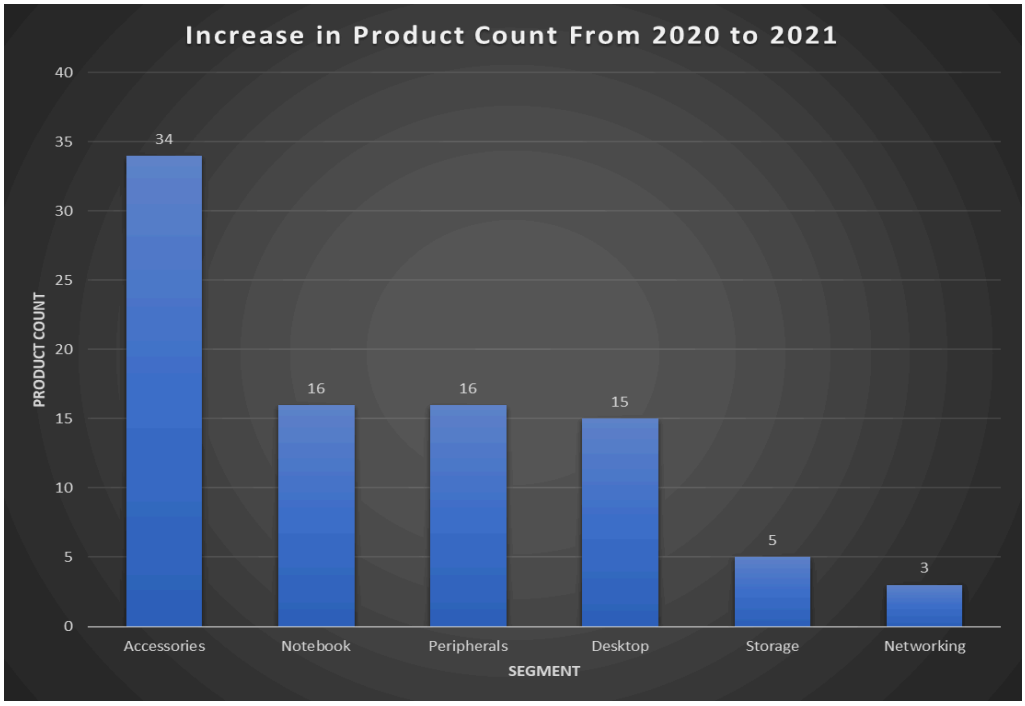
```
-- Provide a report with all the unique product counts for each  segment  and sort them in descending order of product counts.
select segment, count(distinct product) as product_count from dim_product
group by segment
order by count(distinct product) desc;
```

	segment	product_count
▶	Accessories	20
	Peripherals	20
	Notebook	17
	Storage	9
	Desktop	4
	Networking	3



Which segment had the most increase in unique products in 2021 vs 2020?

```
-- Which segment had the most increase in unique products in 2021 vs 2020?
with t1 as
(select segment,count(distinct product_code) as product_count_2020
from unique_products_data where fiscal_year = 2020
group by segment),
t2 as
(select segment,count(distinct product_code) as product_count_2021
from unique_products_data where fiscal_year = 2021
group by segment)
select t1.*,t2.product_count_2021,
product_count_2021 - product_count_2020 as difference
from t1 inner join t2 on t1.segment = t2.segment
order by product_count_2021 - product_count_2020 desc;
```



Get the products that have the highest and lowest manufacturing costs.

Output:

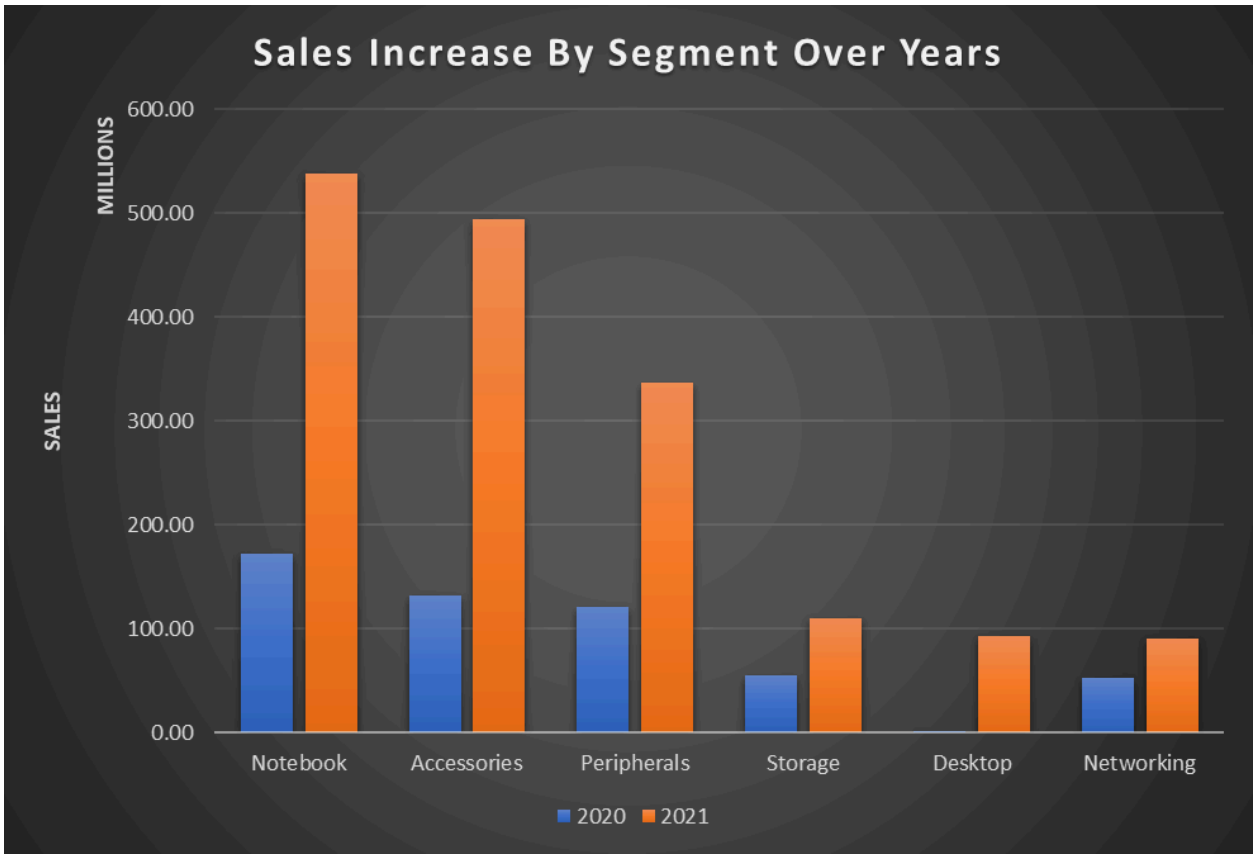
	min_costing_product	max_costing_product
▶	AQ Master wired x1 Ms	AQ HOME Allin1 Gen 2

What are the year-over-year sales growth rates by segment.

```
-- What are the year-over-year sales growth rates by segment.
with sales_2020 as
(select segment, round(sum(sales_amount),2) as sales_2020 from sales_table where fiscal_year = 2020
group by segment),
sales_2021 as
(select segment, round(sum(sales_amount),2) as sales_2021 from sales_table where fiscal_year = 2021
group by segment)

select t1.segment,t1.sales_2020,t2.sales_2021,
t2.sales_2021 - t1.sales_2020 as sales_inc_or_dec,
round(((t2.sales_2021 - t1.sales_2020)/(t1.sales_2020) * 100),2) as inc_or_dec_perc
from sales_2020 t1 inner join sales_2021 t2 on t1.segment = t2.segment;
```

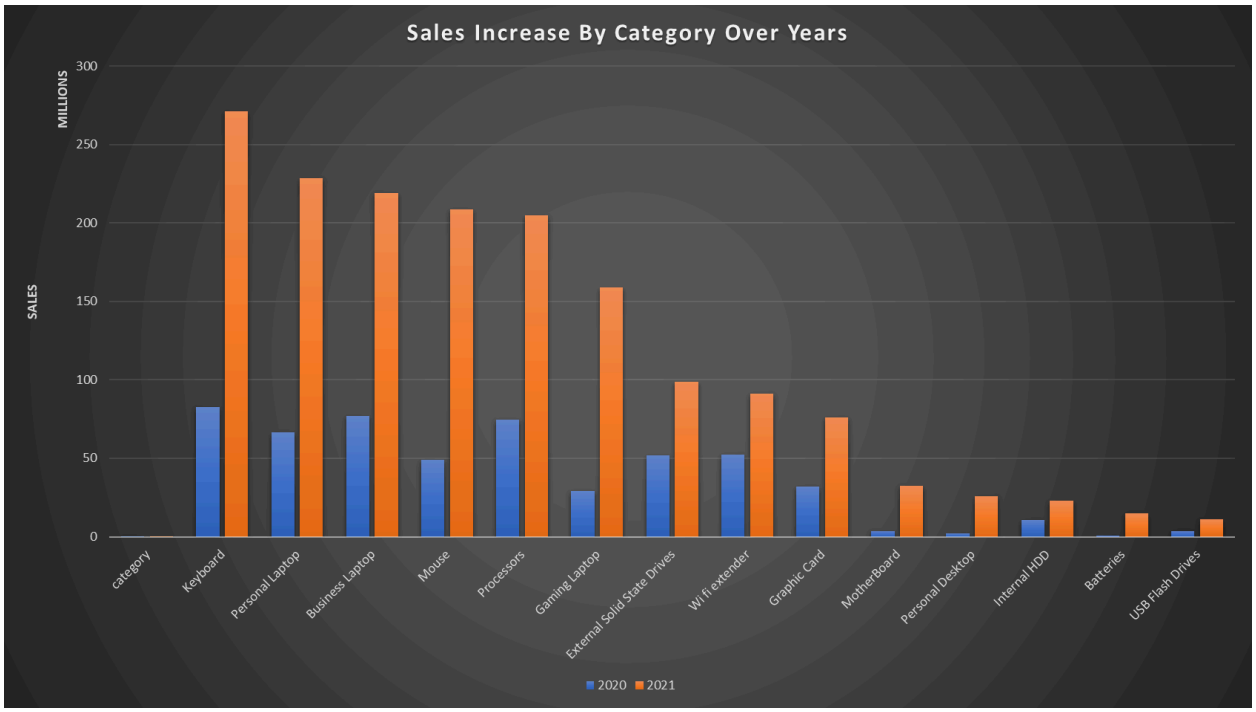
Output:

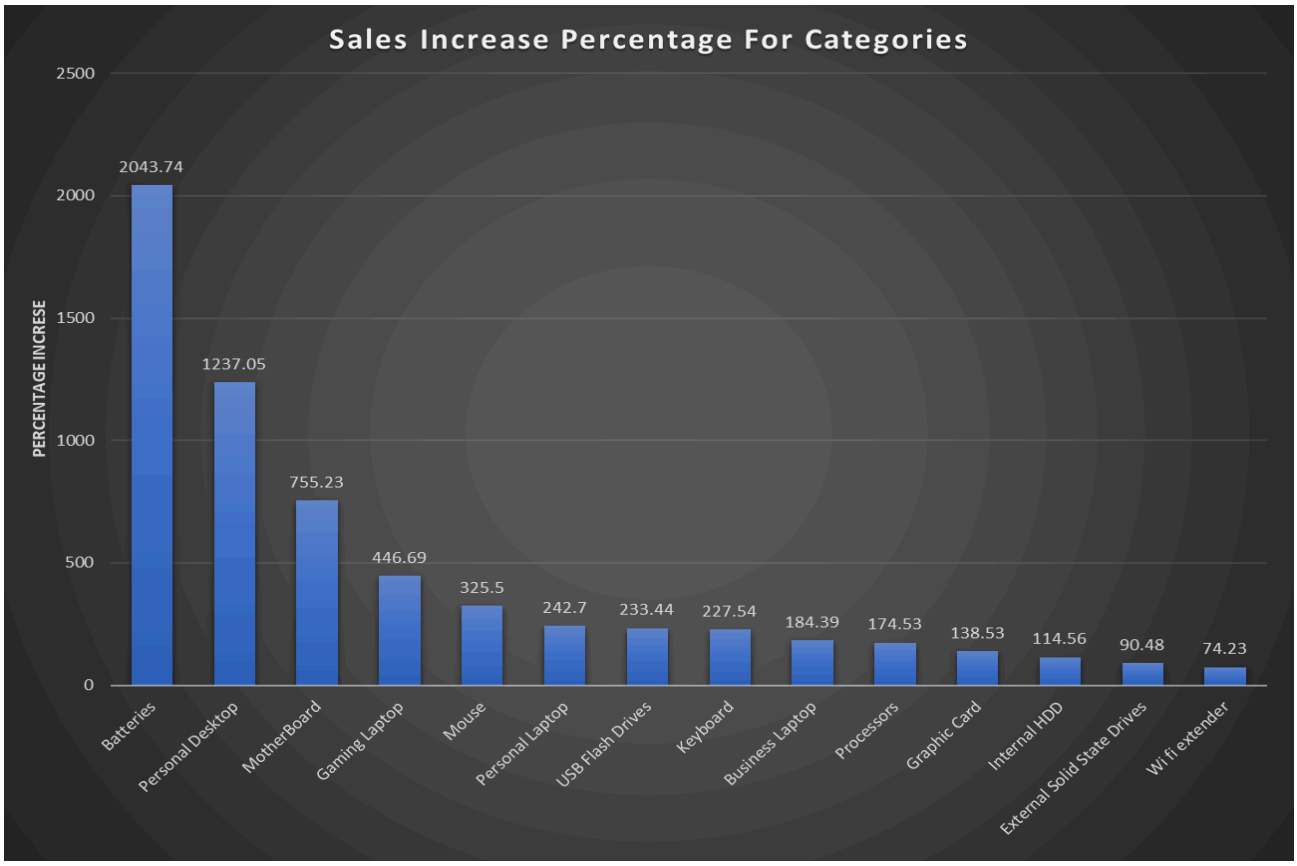
segment	sales_2020	sales_2021	sales_inc_or_dec	inc_or_dec_perc
Peripherals	121148195.62	336632418.14	215484222.52	177.87
Accessories	132488119.23	494658987.62	362170868.39	273.36
Notebook	172778744.89	538339860.80	365561115.91	211.58
Desktop	1920741.68	93735607.47	91814865.79	4780.18
Storage	55208714.60	109975673.99	54766959.39	99.20
Networking	52401255.88	91296725.52	38895469.64	74.23



What is the YoY Sales Growth By Product Category.

```sql
-- What are the year-over-year sales growth rates by category.
-- repeating same for the category.
with sales_2020 as
(select category, round(sum(sales_amount),2) as sales_2020 from sales_table where fiscal_year = 2020
group by category),
sales_2021 as
(select category, round(sum(sales_amount),2) as sales_2021 from sales_table where fiscal_year = 2021
group by category)

select t1.category,t1.sales_2020,t2.sales_2021,
t2.sales_2021 - t1.sales_2020 as sales_inc_or_dec,
round(((t2.sales_2021 - t1.sales_2020)/(t1.sales_2020) * 100),2) as inc_or_dec_perc
from sales_2020 t1 inner join sales_2021 t2 on t1.category = t2.category;
```

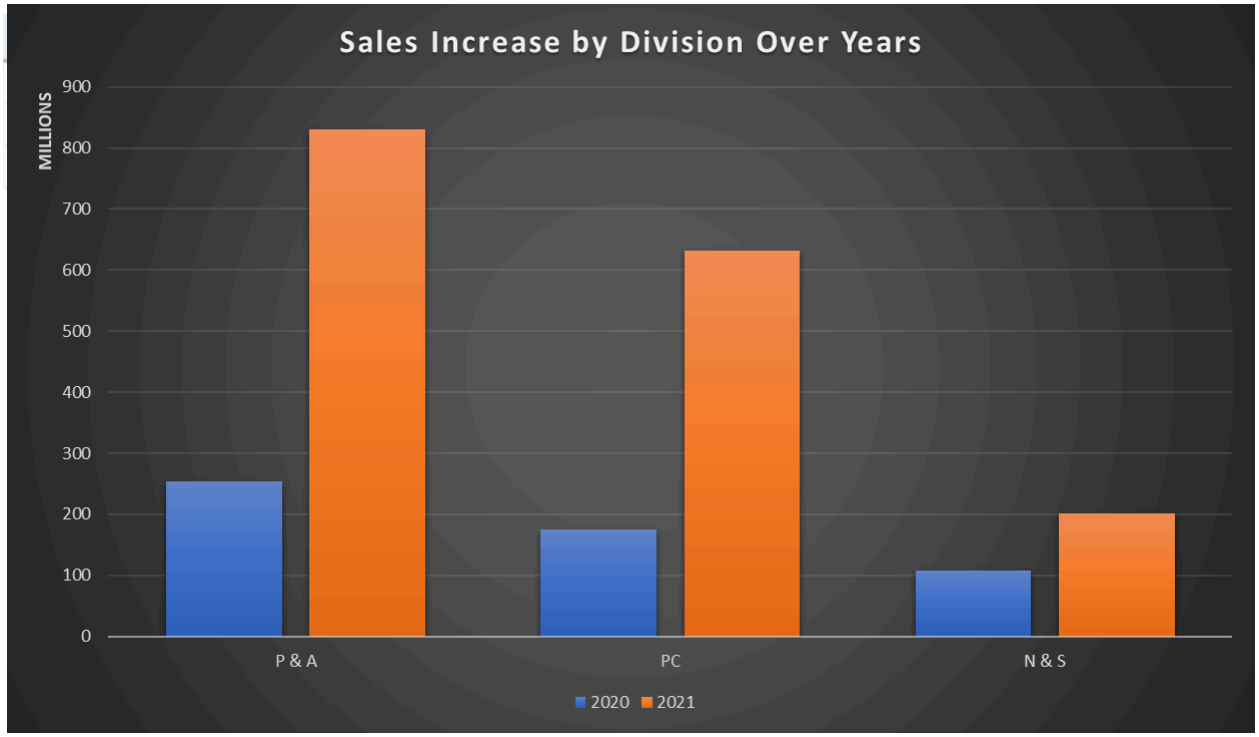


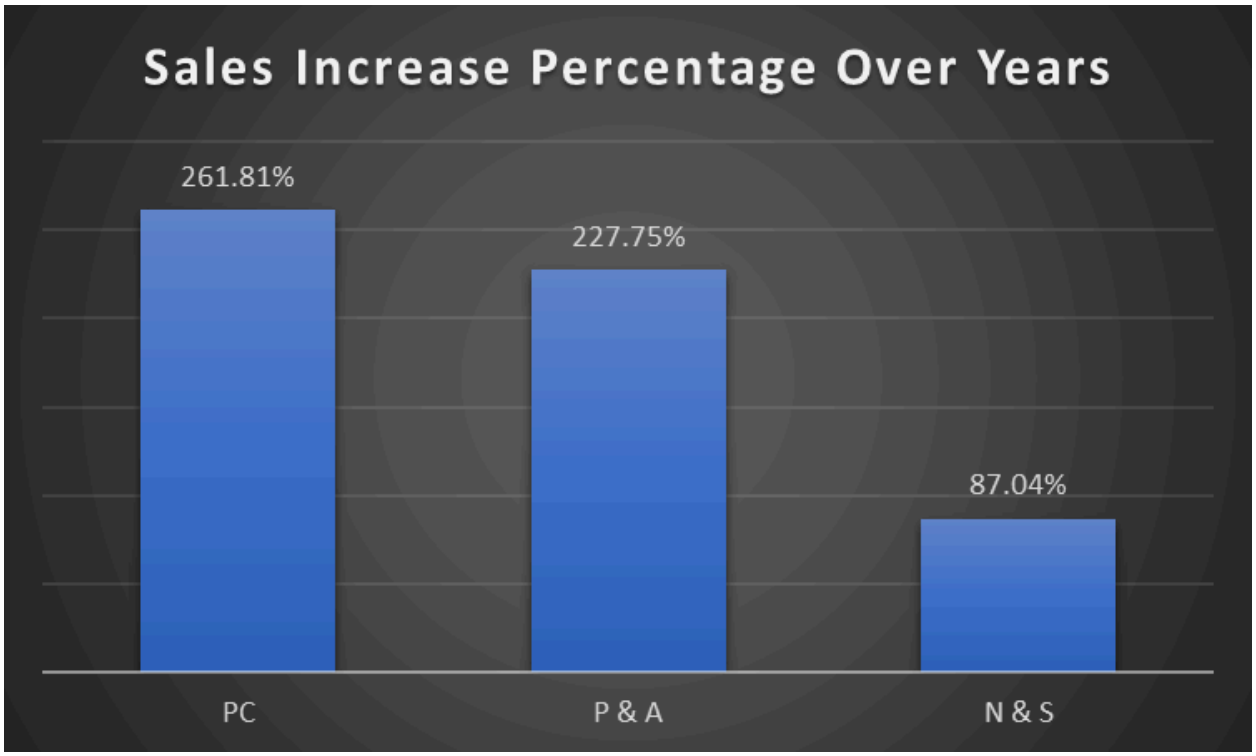


What are the year-over-year sales growth rates by division.

```
-- What are the year-over-year sales growth rates by division.
with sales_2020 as
(select division, round(sum(sales_amount),2) as sales_2020 from sales_table where fiscal_year = 2020
group by division),
sales_2021 as
(select division, round(sum(sales_amount),2) as sales_2021 from sales_table where fiscal_year = 2021
group by division)

select t1.division,t1.sales_2020,t2.sales_2021,
t2.sales_2021 - t1.sales_2020 as sales_inc_or_dec,
round(((t2.sales_2021 - t1.sales_2020)/(t1.sales_2020) * 100),2) as inc_or_dec_perc
```

Output:


Sales Increase by Division Over Years
MILLIONS
900
800
700
600
500
400
300
200
100
0
P & A
PC
N & S
2020
2021

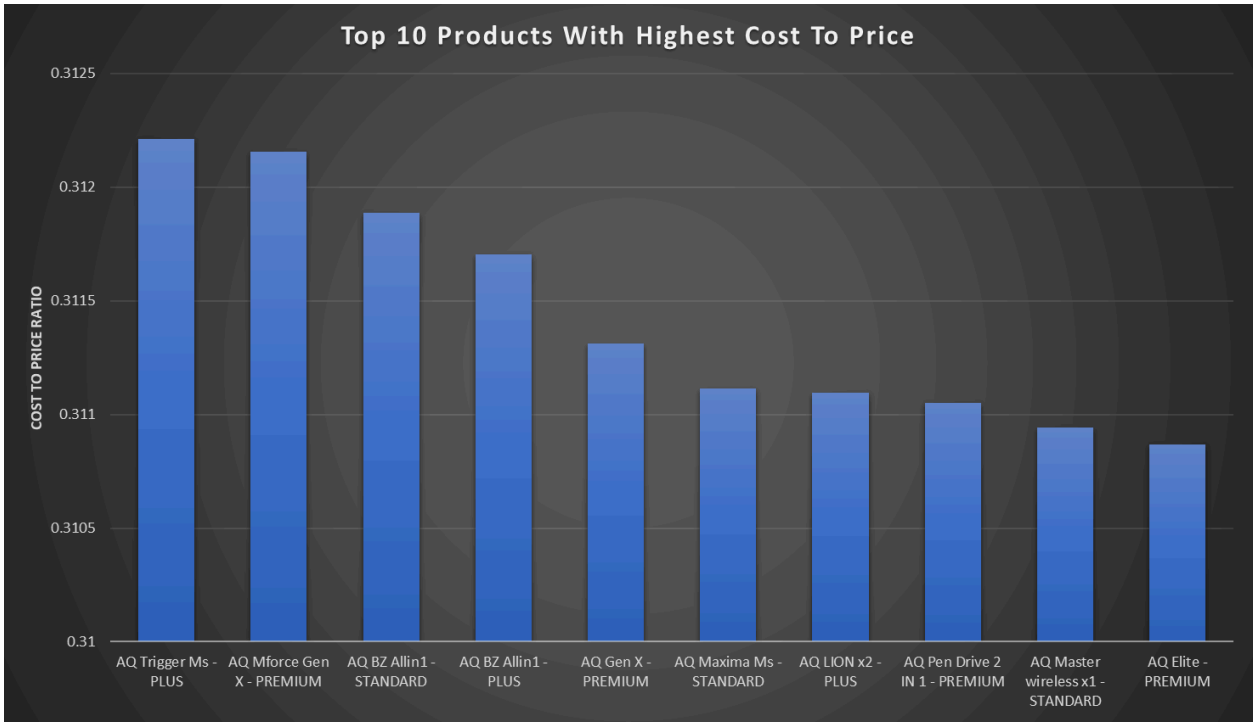

Sales Increase Percentage Over Years
261.81%
227.75%
87.04%
PC
P & A
N & S

Which Products have the highest manufacturing cost to price ratio. Low margin products.

```
-- Which products have the highest manufacturing cost-to-price ratio (low margin products).
with temp_table_2 as
(with temp_table as
(select t1.product_code,t1.cost_year,t1.manufacturing_cost,t2.gross_price
from
fact_manufacturing_cost t1 inner join
fact_gross_price t2 on t1.product_code = t2.product_code
and t1.cost_year = t2.fiscal_year)
select product_code, avg(manufacturing_cost)/avg(gross_price) as cost_to_price
from temp_table
group by product_code
order by avg(manufacturing_cost)/avg(gross_price) desc)
select t1.product_code,t1.cost_to_price,t2.product,t2.cleaned_variant
 from temp_table_2 t1 inner join dim_product t2 on t1.product_code = t2.product_code
 order by t1.cost_to_price desc limit 10;
```

Output:

product_code	cost_to_price	product	cleaned_variant
A2721150703	0.312211770680	AQ Trigger Ms	PLUS
A0418150108	0.312157487862	AQ Mforce Gen X	PREMIUM
A5820110101	0.311888615654	AQ BZ Allin1	STANDARD
A5820110106	0.311703331089	AQ BZ Allin1	PLUS
A4520110507	0.311313527543	AQ Gen X	PREMIUM
A2520150501	0.311113688903	AQ Maxima Ms	STANDARD
A3818150202	0.311097426875	AQ LION x2	PLUS
A6720160103	0.311050168285	AQ Pen Drive 2 IN 1	PREMIUM
A3018150201	0.310943122142	AQ Master wireless x1	STANDARD
A4419110407	0.310868597789	AQ Elite	PREMIUM



Most Sold Product for each year.

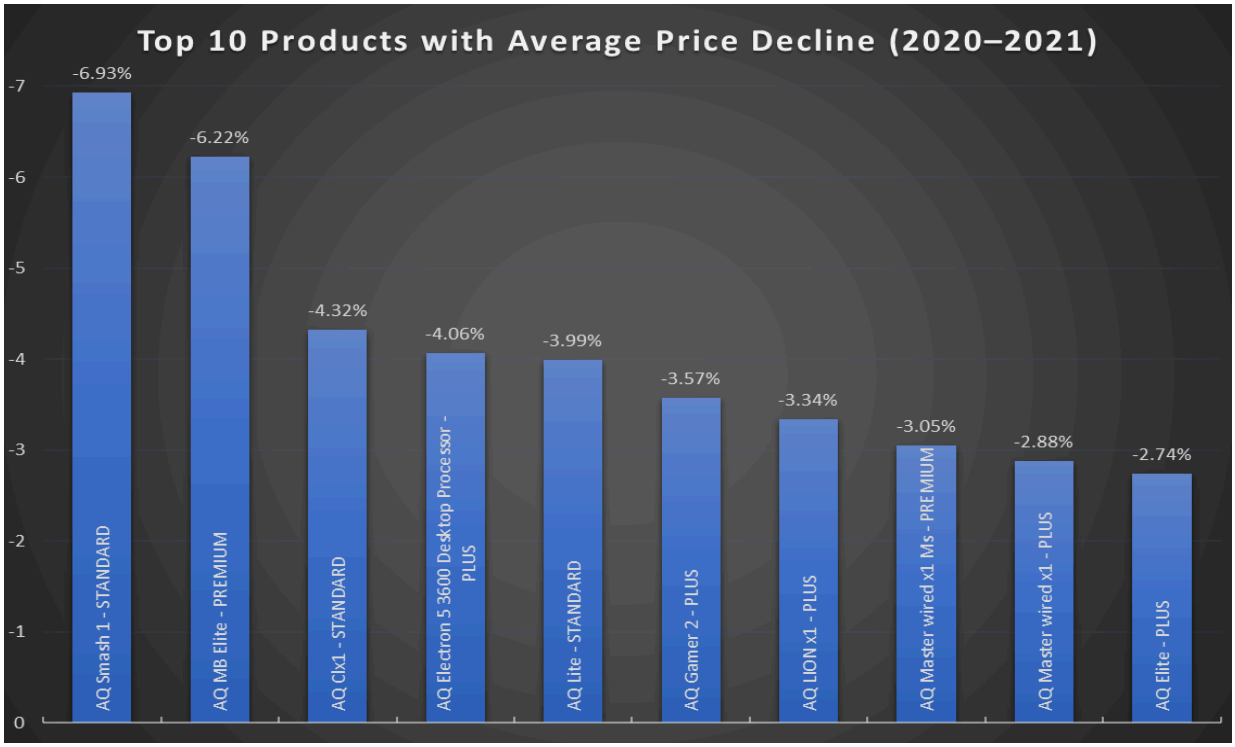
fiscal_year	product_code	product	sales_amt	rank
2020	A7220160203	AQ Wi Power Dx2	10723886.3952	1
2021	A1521150602	AQ Electron 3 3600 Desktop Processor	15150219.2310	1

Which products are showing declining average selling prices(ASP) year over year.

```
-- most revenue generating product across years.
with top_sales_generating_product_years as
(select fiscal_year,product_code,product,
sum(sales_amount) as sales_amt,
rank() over(partition by fiscal_year order by sum(sales_amount) desc rows between unbounded preceding and unbounded following) as `rank`
from sales_table
group by fiscal_year,product_code,product)
select * from top_sales_generating_product_years
where `rank` = 1;
```

```sql
create view price_decline_analysis as
(with avg_price_2020 as
(select product_code,product,standardised_variant, avg(gross_price) as avg_price_2020 from sales_table
where fiscal_year = 2020
group by product_code,product,standardised_variant),
avg_price_2021 as
(
select product_code,product,standardised_variant, avg(gross_price) as avg_price_2021 from sales_table
where fiscal_year = 2021
group by product_code,product,standardised_variant
)
select t1.*,t2.avg_price_2021,
t2.avg_price_2021 - t1.avg_price_2020 as price_inc_or_dec,
round(((t2.avg_price_2021 - t1.avg_price_2020)/(t1.avg_price_2020)) * 100,2) as price_inc_or_dec_perc
from avg_price_2020 t1 inner join avg_price_2021 t2
on t1.product_code = t2.product_code and
t1.product = t2.product);
select * from price_decline_analysis where price_inc_or_dec < 0 order by price_inc_or_dec_perc asc limit 10;
```

product_code	product	standardised_variant	avg_price_2020	avg_price_2021	price_inc_or_dec	price_inc_or_dec_perc
A5620110402	AQ Smash 1	STANDARD	638.04620000	593.82970000	-44.21650000	-6.93
A1618150104	AQ MB Elite	PREMIUM	22.87040000	21.44890000	-1.42150000	-6.22
A6419160301	AQ Clx1	STANDARD	18.60790000	17.80430000	-0.80360000	-4.32
A1320150402	AQ Electron 5 3600 Desktop Processor	PLUS	156.59570000	150.24500000	-6.35070000	-4.06
A3220150402	AQ Lite	STANDARD	19.57570000	18.79470000	-0.78100000	-3.99
A5419110205	AQ Gamer 2	PLUS	591.85990000	570.75780000	-21.10210000	-3.57
A3718150102	AQ LION x1	PLUS	16.65870000	16.10150000	-0.55720000	-3.34
A2118150105	AQ Master wired x1 Ms	PREMIUM	4.83810000	4.69040000	-0.14770000	-3.05
A2918150104	AQ Master wired x1	PLUS	9.21720000	8.95140000	-0.26580000	-2.88
A4419110405	AQ Elite	PLUS	299.41720000	291.20710000	-8.21010000	-2.74

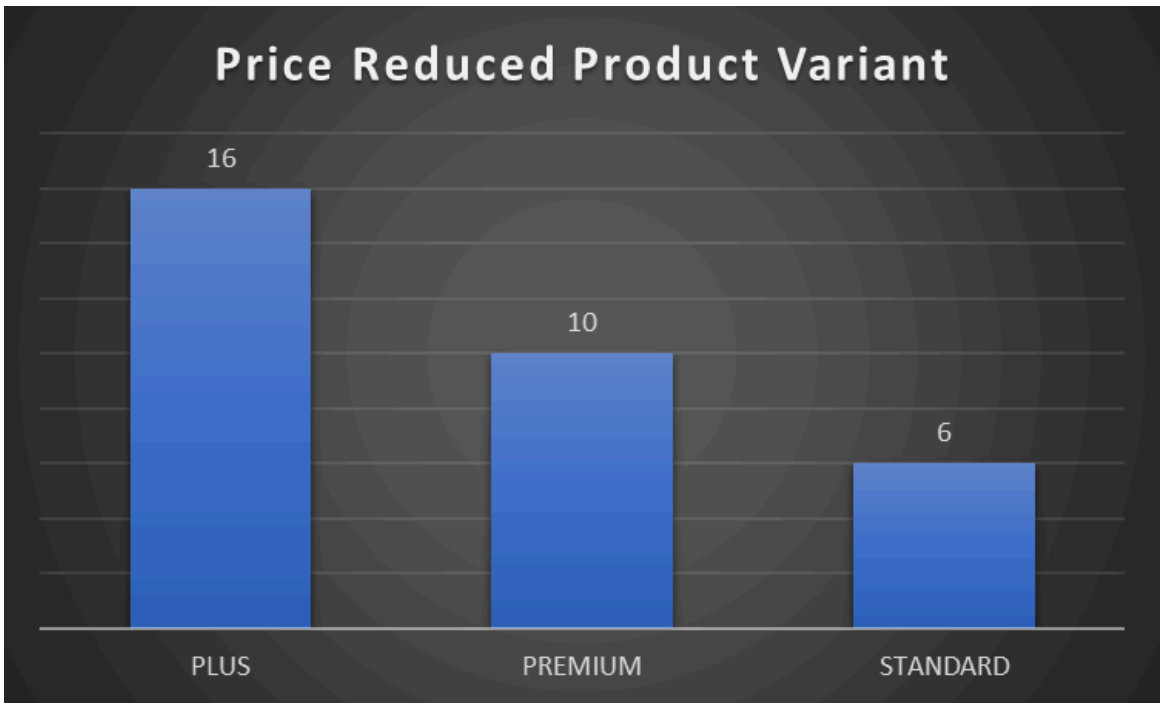

Top 10 Products with Average Price Decline (2020–2021)
-7
-6
-5
-4
-3
-2
-1
0
-6.93%
-6.22%
-4.32%
-4.06%
-3.99%
-3.57%
-3.34%
-3.05%
-2.88%
-2.74%
AQ Smash 1 - STANDARD
AQ MB Elite - PREMIUM
AQ Clx1 - STANDARD
AQ Electron 5 3600 Desktop Processor - PLUS
AQ Lite - STANDARD
AQ Gamer 2 - PLUS
AQ LION x1 - PLUS
AQ Master wired x1 Ms - PREMIUM
AQ Master wired x1 - PLUS
AQ Elite - PLUS

Which product variants are showing the declining average selling price (ASP) year over years.

```
-- Which product variant are showing the declining average selling price (ASP) year over years.
select standardised_variant, count(*) from price_decline_analysis where price_inc_or_dec < 0 group by standardised_variant
order by count(*) desc;
```

Output:

standardised_variant	count(*)
PLUS	16
PREMIUM	10
STANDARD	6



Get top 5 and bottom 5 margin product in each division

```sql
create view marginal_analysis as
(select t1.product_code,t1.fiscal_year,t2.division,t2.segment,t2.category,t2.product,t2.cleaned_variant,
t3.manufacturing_cost,t4.gross_price
from fact_sales_monthly t1
inner join dim_product t2 on t1.product_code = t2.product_code
inner join fact_manufacturing_cost t3 on t3.product_code = t1.product_code and t3.cost_year = t1.fiscal_year
inner join fact_gross_price t4 on t4.product_code = t1.product_code and t4.fiscal_year = t1.fiscal_year);
```

```sql
with temp_table as
(select division, product_code,product, avg(gross_price) - avg(manufacturing_cost) as margin,
rank() over(partition by division order by avg(gross_price) - avg(manufacturing_cost) desc rows between unbounded preceding and unbounded following) as top_5,
rank() over(partition by division order by avg(gross_price) - avg(manufacturing_cost) asc rows between unbounded preceding and unbounded following) as bottom_5
from marginal_analysis
group by division, product_code, product)
select division,product_code,product,
margin,
case
    when top_5 <= 5 then 'High Margin Product'
    when bottom_5 <= 5 then 'Low Margin Product'
end
as category
from temp_table
where top_5 <=5 or bottom_5 <= 5;
```

Output:

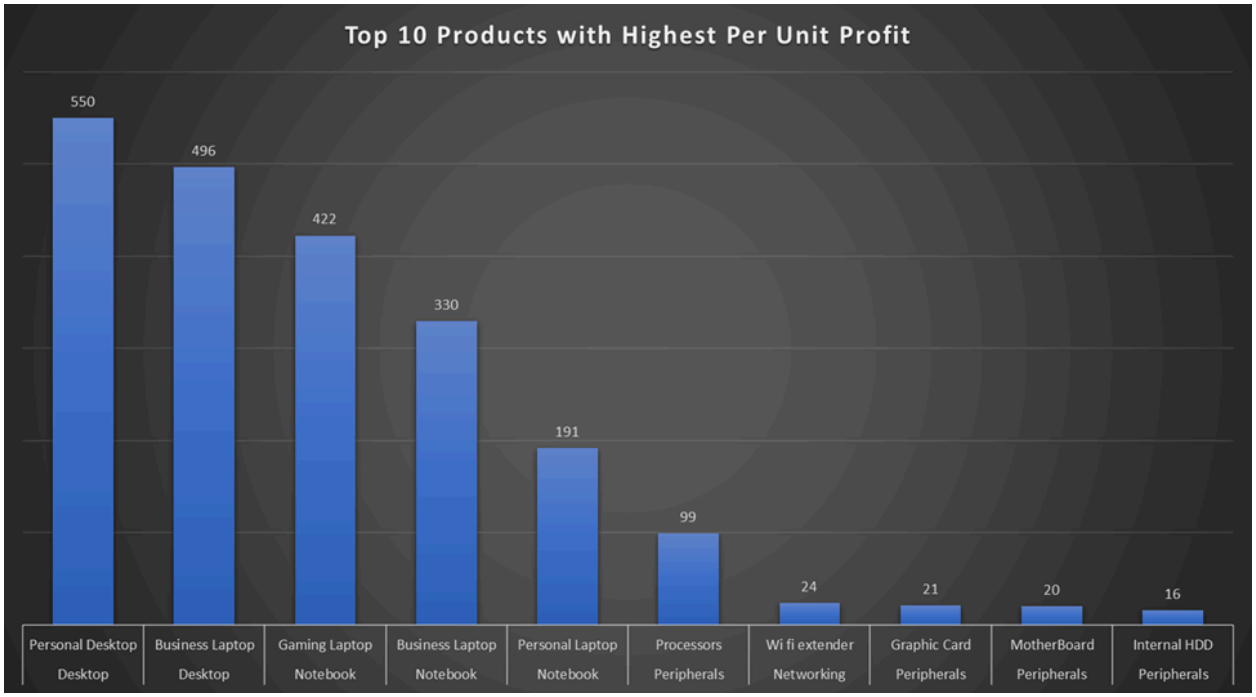
division	product_code	product	margin	category
N & S	A6818160201	AQ Pen Drive DRC	2.07375918	Low Margin Product
N & S	A6818160202	AQ Pen Drive DRC	2.70370000	Low Margin Product
N & S	A6720160103	AQ Pen Drive 2 IN 1	3.03979111	Low Margin Product
N & S	A6819160203	AQ Pen Drive DRC	3.56580000	Low Margin Product
N & S	A6218160101	AQ Digit SSD	9.53009287	Low Margin Product
N & S	A7220160202	AQ Wi Power Dx2	24.94668120	High Margin Product
N & S	A7220160203	AQ Wi Power Dx2	26.33586696	High Margin Product
N & S	A7321160301	AQ Wi Power Dx3	28.38160000	High Margin Product
N & S	A7321160303	AQ Wi Power Dx3	29.89810000	High Margin Product
N & S	A7321160302	AQ Wi Power Dx3	31.37760000	High Margin Product
P & A	A2118150101	AQ Master wired x1 Ms	2.02434868	Low Margin Product
P & A	A2118150102	AQ Master wired x1 Ms	2.65861408	Low Margin Product
P & A	A2118150103	AQ Master wired x1 Ms	3.14250264	Low Margin Product
P & A	A2118150105	AQ Master wired x1 Ms	3.32979272	Low Margin Product
P & A	A2118150104	AQ Master wired x1 Ms	3.44789448	Low Margin Product
P & A	A1320150403	AQ Electron 5 3600 Desktop Processor	108.74518601	High Margin Product
P & A	A1420150502	AQ Electron 4 3600 Desktop Processor	118.77190000	High Margin Product
P & A	A1421150503	AQ Electron 4 3600 Desktop Processor	122.41590000	High Margin Product
P & A	A1521150602	AQ Electron 3 3600 Desktop Processor	128.21850000	High Margin Product
P & A	A1521150601	AQ Electron 3 3600 Desktop Processor	132.03880000	High Margin Product
PC	A4118110101	AQ Aspiron	107.78260000	Low Margin Product
PC	A4118110103	AQ Aspiron	107.80540000	Low Margin Product
PC	A4118110104	AQ Aspiron	118.27060000	Low Margin Product
PC	A4118110104	AQ Aspiron	118.27060000	Low Margin Product
PC	A4118110102	AQ Aspiron	122.87360000	Low Margin Product
PC	A4118110106	AQ Aspiron	123.76806743	Low Margin Product
PC	A6019110108	AQ Home Allin1	562.48440000	High Margin Product
PC	A6119110204	AQ HOME Allin1 Gen 2	568.48990000	High Margin Product
PC	A6120110206	AQ HOME Allin1 Gen 2	570.06700000	High Margin Product

Which product segments (e.g., Notebooks, Storage, Peripherals) generate the highest profit per unit sold?

```
-- using session variable to avoid error code 2013.
```

segment	category	profit_per_unit_sold_without_disc	perc_contribution_to_total_qty_sold	per_contribution_to_overall_sales
Desktop	Personal Desktop	549.97	0.0496	1.25430432
Desktop	Business Laptop	496.33	0.1349	3.09255560
Notebook	Gaming Laptop	422.48	0.4359	8.53392123
Notebook	Business Laptop	329.64	0.6801	10.37402596
Notebook	Personal Laptop	191.48	1.4897	13.40703229
Peripherals	Processors	99.07	2.7511	12.70074280
Networking	Wi fi extender	23.78	5.9831	6.52998991
Peripherals	Graphic Card	21.24	4.9070	4.91476591
Peripherals	MotherBoard	19.98	1.7180	1.64546317
Peripherals	Internal HDD	15.92	2.0903	1.54170578
Accessories	Batteries	15.30	0.9746	0.71180921
Storage	External Solid St...	12.48	11.8435	6.84338328
Accessories	Keyboard	12.01	28.2066	16.07849543
Accessories	Mouse	7.32	33.5882	11.70880327
Storage	USB Flash Drives	2.75	5.1476	0.66300184

Output:

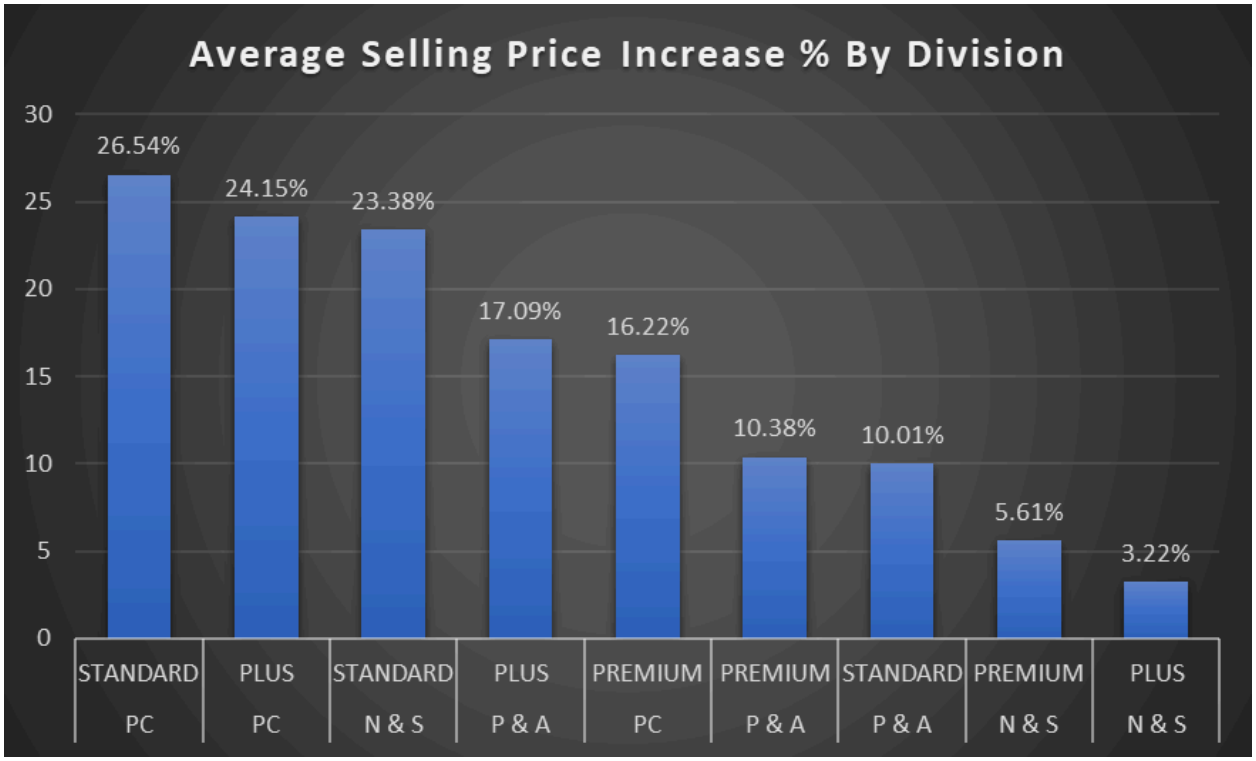


Which Product division shows the most decrease/increase in the average selling price over the year.

```sql
-- Which Product Division shows the most decrease in the selling price over year.
with product_division_price_decline_2020 as
(select division, standardised_variant ,avg(gross_price) as avg_gross_price_2020 from sales_table
where fiscal_year = 2020
group by division, standardised_variant),
product_division_price_decline_2021 as
(select division, standardised_variant ,avg(gross_price) as avg_gross_price_2021 from sales_table
where fiscal_year = 2021
group by division, standardised_variant)
select *,
avg_gross_price_2021 - avg_gross_price_2020 as price_inc_or_dec,
round(((avg_gross_price_2021 - avg_gross_price_2020)/(avg_gross_price_2020)) * 100,2) as price_inc_or_dec_perc
from product_division_price_decline_2020 t1
inner join product_division_price_decline_2021 t2 on t1.division = t2.division and t1.standardised_variant = t2.standardised_variant
order by round(((avg_gross_price_2021 - avg_gross_price_2020)/(avg_gross_price_2020)) * 100,2) desc;
```

Output:

division	standardised_variant	avg_gross_price_2020	division	standardised_variant	avg_gross_price_2021	price_inc_or_dec	price_inc_or_dec_perc
PC	STANDARD	379.72249568	PC	STANDARD	480.51687916	100.79438348	26.54
PC	PLUS	391.31863534	PC	PLUS	485.80619150	94.48755616	24.15
N & S	STANDARD	17.86048003	N & S	STANDARD	22.03582222	4.17534219	23.38
P & A	PLUS	28.86614007	P & A	PLUS	33.80007923	4.93393916	17.09
PC	PREMIUM	421.28501083	PC	PREMIUM	489.60120476	68.31619393	16.22
P & A	PREMIUM	31.13601885	P & A	PREMIUM	34.36679540	3.23077655	10.38
P & A	STANDARD	30.37439441	P & A	STANDARD	33.41537522	3.04098081	10.01
N & S	PREMIUM	20.50520046	N & S	PREMIUM	21.65562222	1.15042176	5.61
N & S	PLUS	22.59651193	N & S	PLUS	23.32351250	0.72700057	3.22



Which Product Category shows the most decrease/increase in the average selling price over the year.

Across all the product categories the average selling price has increased from 2020 to 2021.

```sql
-- Which Product Category Product are showing the most decrease in the selling price over year.
with product_category_price_decline_2020 as
(select category, standardised_variant ,avg(gross_price) as avg_gross_price_2020 from sales_table
where fiscal_year = 2020
group by category, standardised_variant),
product_category_price_decline_2021 as
(select category, standardised_variant ,avg(gross_price) as avg_gross_price_2021 from sales_table
where fiscal_year = 2021
group by category, standardised_variant)
select *,
avg_gross_price_2021 - avg_gross_price_2020 as price_inc_or_dec,
round(((avg_gross_price_2021 - avg_gross_price_2020)/(avg_gross_price_2020)) * 100,2) as price_inc_or_dec_perc
from product_category_price_decline_2020 t1
inner join product_category_price_decline_2021 t2 on t1.category = t2.category and t1.standardised_variant = t2.standardised_variant
order by round(((avg_gross_price_2021 - avg_gross_price_2020)/(avg_gross_price_2020)) * 100,2) desc;
```

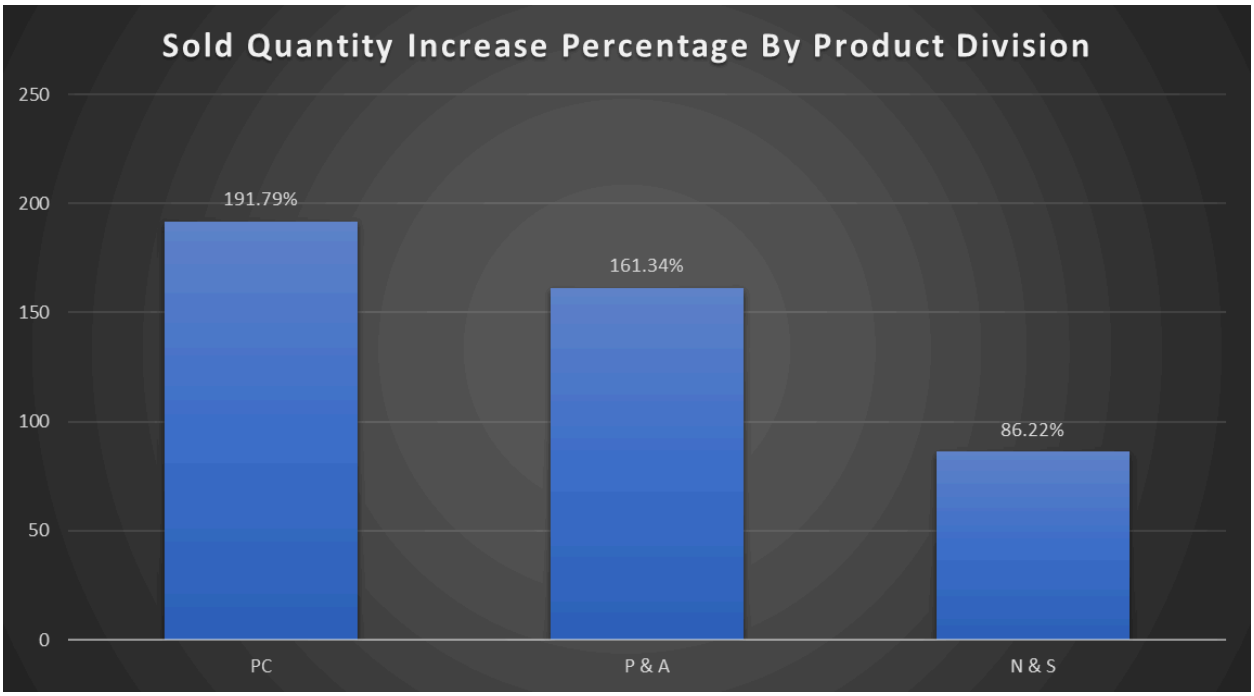
category	standardised_variant	avg_gross_price_2020	category	standardised_variant	avg_gross_price_2021	price_inc_or_dec	price_inc_or_dec_perc
Mouse	STANDARD	6.82305709	Mouse	STANDARD	11.51290667	4.68984958	68.74
Mouse	PREMIUM	8.59235731	Mouse	PREMIUM	12.51166429	3.91930698	45.61
Mouse	PLUS	8.04717903	Mouse	PLUS	11.68118571	3.63400668	45.16
Keyboard	PREMIUM	14.18881088	Keyboard	PREMIUM	20.03156923	5.84275835	41.18
Keyboard	PLUS	14.45080560	Keyboard	PLUS	19.50616429	5.05535869	34.98
Keyboard	STANDARD	13.72943675	Keyboard	STANDARD	18.05763571	4.32819896	31.52
Graphic Card	PREMIUM	28.58548888	Graphic Card	PREMIUM	37.08249535	8.49700647	29.72
MotherBoard	PLUS	24.29244787	MotherBoard	PLUS	31.17404850	6.88160063	28.33
Personal Laptop	STANDARD	232.56229519	Personal Laptop	STANDARD	298.18033879	65.61804360	28.22
MotherBoard	STANDARD	22.93542028	MotherBoard	STANDARD	29.10158495	6.16616467	26.88
Wi fi extender	STANDARD	27.28200000	Wi fi extender	STANDARD	34.47470000	7.19270000	26.36
Graphic Card	STANDARD	25.32475116	Graphic Card	STANDARD	31.71822512	6.39347396	25.25
Personal Laptop	PREMIUM	238.07139549	Personal Laptop	PREMIUM	292.59843193	54.52703644	22.90
Personal Laptop	PLUS	240.59911039	Personal Laptop	PLUS	293.00890676	52.40979637	21.78
Business Laptop	PREMIUM	452.16056456	Business Laptop	PREMIUM	542.15658066	89.99601610	19.90
Business Laptop	PLUS	464.05306020	Business Laptop	PLUS	554.27341714	90.22035694	19.44
Batteries	PREMIUM	20.75011869	Batteries	PREMIUM	24.69574565	3.94562696	19.01
Graphic Card	PLUS	27.44162984	Graphic Card	PLUS	32.52195364	5.08032380	18.51
Business Laptop	STANDARD	446.19886716	Business Laptop	STANDARD	521.58421497	75.38534781	16.90
Batteries	PLUS	19.73691178	Batteries	PLUS	22.85849983	3.12158805	15.82
Processors	PREMIUM	131.10928830	Processors	PREMIUM	148.77960000	17.67031170	13.48
External Solid S...	PLUS	16.24107581	External Solid S...	PLUS	18.41467500	2.17359919	13.38
Wi fi extender	PLUS	32.12796679	Wi fi extender	PLUS	36.35876667	4.23079988	13.17
Processors	STANDARD	129.58719802	Processors	STANDARD	144.49218333	14.90498531	11.50
Personal Desktop	PLUS	711.39673799	Personal Desktop	PLUS	792.72032024	81.32358225	11.43
Processors	PLUS	135.36147003	Processors	PLUS	150.57110961	15.20963958	11.24
MotherBoard	PREMIUM	27.23157433	MotherBoard	PREMIUM	30.27180000	3.04022567	11.16
Internal HDD	PLUS	22.00635808	Internal HDD	PLUS	24.43013333	2.42377525	11.01
External Solid S...	PREMIUM	17.20907042	External Solid S...	PREMIUM	19.06520000	1.85612958	10.79
Internal HDD	STANDARD	20.24698320	Internal HDD	STANDARD	22.15943333	1.91245013	9.45
External Solid S...	STANDARD	16.88212679	External Solid S...	STANDARD	18.37986000	1.49773321	8.87
USB Flash Drives	PREMIUM	4.38320000	USB Flash Drives	PREMIUM	4.76975000	0.38655000	8.82
Gaming Laptop	PLUS	569.11905641	Gaming Laptop	PLUS	618.96593279	49.84687638	8.76
Wi fi extender	PREMIUM	33.54863929	Wi fi extender	PREMIUM	36.36676667	2.81812738	8.40
Personal Desktop	PREMIUM	730.05900000	Personal Desktop	PREMIUM	791.33080656	61.27180656	8.39
Internal HDD	PREMIUM	22.27353820	Internal HDD	PREMIUM	24.09342500	1.81988680	8.17
Gaming Laptop	STANDARD	568.96223267	Gaming Laptop	STANDARD	612.04954373	43.08731106	7.57
Batteries	STANDARD	19.33297928	Batteries	STANDARD	20.71253635	1.37955707	7.14
Personal Desktop	STANDARD	736.54184509	Personal Desktop	STANDARD	786.67346710	50.13162201	6.81
Gaming Laptop	PREMIUM	578.38759833	Gaming Laptop	PREMIUM	608.89992961	30.51233128	5.28
USB Flash Drives	STANDARD	2.96910000	USB Flash Drives	STANDARD	2.99900000	0.02990000	1.01

Sold Quantity Increase from 2020 to 2021 By Product Division.

```sql
-- Sales by Quantity Increase by division from 2020 to 2021.
with sales_amt_by_division_2020 as
(select division,sum(sold_quantity) as amt_sold_2020 from sales_table where fiscal_year = 2020
group by division
),
sales_amt_by_division_2021 as
(select division,sum(sold_quantity) as amt_sold_2021 from sales_table where fiscal_year = 2021
group by division
)
select *,
(amt_sold_2021 - amt_sold_2020) as increase_in_amount_sold,
round(((amt_sold_2021 - amt_sold_2020)/(amt_sold_2020)) * 100,2) as perc_increase
from sales_amt_by_division_2020 t1 inner join sales_amt_by_division_2021 t2
on t1.division = t2.division;
```

Output:

division	amt_sold_2020	division	amt_sold_2021	increase_in_amount_sold	perc_increase
P & A	14573725	P & A	38087235	23513510	161.34
PC	505169	PC	1474023	968854	191.79
N & S	5693995	N & S	10603324	4909329	86.22



What share of revenue is driven by products with above-median manufacturing cost-to-sales ratios?

```
with temp_table as
(select product_code,product,variant,
avg(cost_to_sales) as avg_cost_to_sales_ratio,
(sum(sales_amount)/@total_sales_amount) * 100 as per_contribution_to_total_sales,
(sum(sold_quantity)/@total_qty_sold) * 100 as per_contribution_to_qty_sold
from cost_to_sales_table group by product_code,product,variant
having avg(cost_to_sales) > @median_cost_to_sales_ratio
order by avg(cost_to_sales) desc)
select sum(per_contribution_to_total_sales),
sum(per_contribution_to_qty_sold) from
temp_table;
```

Most of the products with the higher cost to price ratio are mostly plus or premium products.

These higher cost to sales ratio products contribute 48.58 % of total_sales and 52.74 % of total quantity sold.

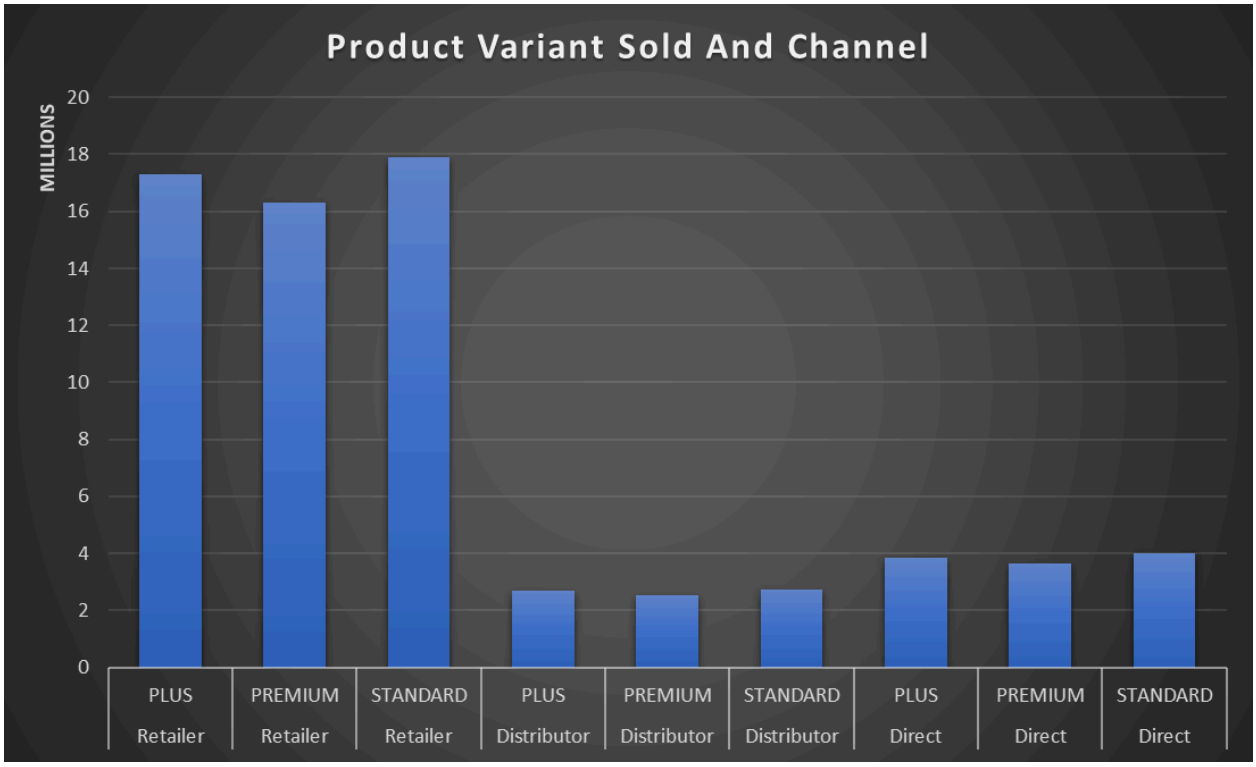
might need to reduce the dependence on the higher cost to price products.

Channel and Product Variant.

```
-- channel and product variant analysis.
select channel,standardised_variant, sum(sold_quantity) as quantity_sold from sales_table
group by channel,standardised_variant
order by channel desc, standardised_variant asc, quantity_sold desc;
```

Output:

channel	standardised_variant	quantity_sold
Retailer	PLUS	17292467
Retailer	PREMIUM	16307097
Retailer	STANDARD	17886057
Distributor	PLUS	2702941
Distributor	PREMIUM	2529903
Distributor	STANDARD	2728183
Direct	PLUS	3839533
Direct	PREMIUM	3645222
Direct	STANDARD	4006068



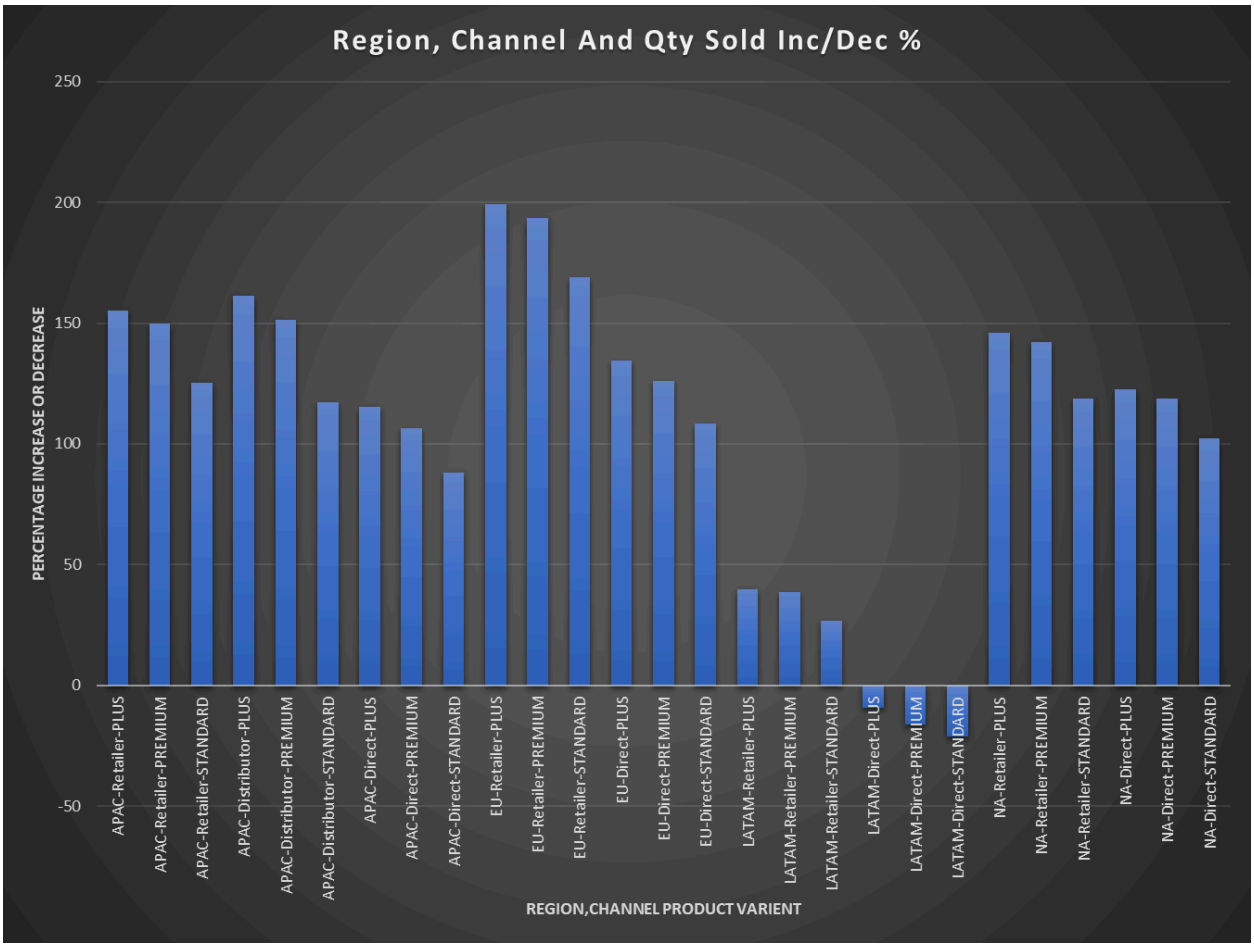
Region Channel Product variant - Nothing much to conclude.

Region Channel Product Variant YoY.

```
with t1 as
(select region,channel,standardised_variant,sum(sold_quantity) as quantity_sold_2020 from sales_table
where fiscal_year = '2020'
group by region, channel, standardised_variant
order by region asc,channel desc, standardised_variant asc,sum(sold_quantity) desc),
t2 as
(select region,channel,standardised_variant,sum(sold_quantity) as quantity_sold_2021 from sales_table
where fiscal_year = '2021'
group by region, channel, standardised_variant
order by region asc,channel desc, standardised_variant asc,sum(sold_quantity) desc)
select t1.region,t1.channel,t1.standardised_variant,t1.quantity_sold_2020, t2.quantity_sold_2021 ,
((t2.quantity_sold_2021 - t1.quantity_sold_2020)/(t1.quantity_sold_2020)) * 100 as perc_inc_dec_in_sold_qty
from t1 inner join t2 on t1.region = t2.region
and t1.channel = t2.channel and t1.standardised_variant = t2.standardised_variant;
```

Output:

region	channel	standardised_variant	quantity_sold_2020	quantity_sold_2021	perc_inc_dec_in_sold_qty
APAC	Retailer	PLUS	2330939	5951291	155.3173
APAC	Retailer	PREMIUM	2232844	5578640	149.8446
APAC	Retailer	STANDARD	2633776	5935783	125.3716
APAC	Distributor	PLUS	748334	1954607	161.1945
APAC	Distributor	PREMIUM	719855	1810048	151.4462
APAC	Distributor	STANDARD	860452	1867731	117.0639
APAC	Direct	PLUS	682629	1470488	115.4154
APAC	Direct	PREMIUM	663419	1370862	106.6359
APAC	Direct	STANDARD	774918	1457359	88.0662
EU	Retailer	PLUS	1136862	3404177	199.4363
EU	Retailer	PREMIUM	1093617	3209049	193.4344
EU	Retailer	STANDARD	1271192	3419736	169.0181
EU	Direct	PLUS	273441	641555	134.6228
EU	Direct	PREMIUM	265828	601337	126.2128
EU	Direct	STANDARD	309442	645032	108.4500
LATAM	Retailer	PLUS	30802	43004	39.6143
LATAM	Retailer	PREMIUM	29169	40457	38.6986
LATAM	Retailer	STANDARD	34326	43451	26.5833
LATAM	Direct	PLUS	26661	24192	-9.2607
LATAM	Direct	PREMIUM	26366	22065	-16.3127
LATAM	Direct	STANDARD	29938	23637	-21.0468
LATAM	Direct	STANDARD	29938	23637	-21.0468
NA	Retailer	PLUS	1270947	3124445	145.8360
NA	Retailer	PREMIUM	1205362	2917959	142.0815
NA	Retailer	STANDARD	1427040	3120753	118.6871
NA	Direct	PLUS	223328	497239	122.6496
NA	Direct	PREMIUM	218143	477202	118.7565
NA	Direct	STANDARD	253259	512483	102.3553

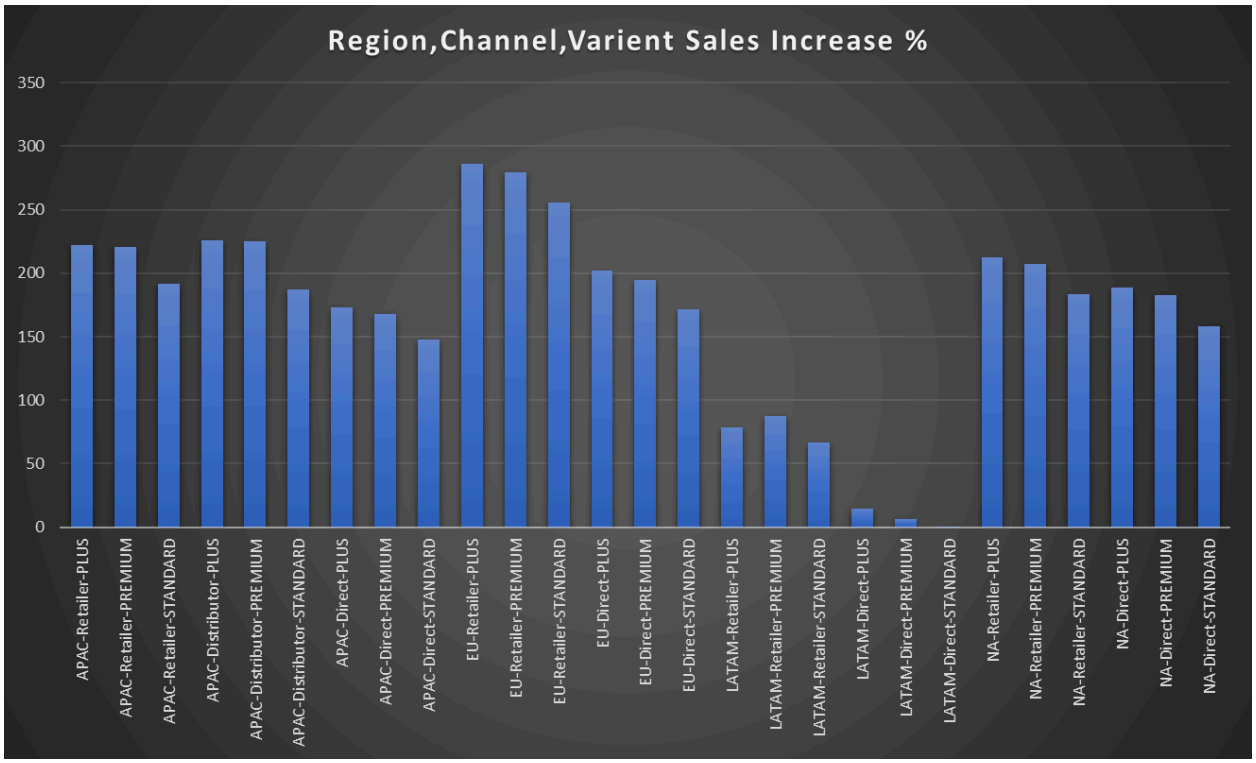


Region,Channel ,product variant year over year analysis for sales amount.

```sql
-- region,channel ,product variant -- yoy analysis for sales amount.

with t1 as
(select region,channel,standardised_variant,sum(sales_amount) as sales_amount_2020 from sales_table
where fiscal_year = '2020'
group by region, channel, standardised_variant
order by region asc,channel desc, standardised_variant asc,sum(sales_amount) desc),
t2 as
(select region,channel,standardised_variant,sum(sales_amount) as sales_amount_2021 from sales_table
where fiscal_year = '2021'
group by region, channel, standardised_variant
order by region asc,channel desc, standardised_variant asc,sum(sales_amount) desc)
select t1.region,t1.channel,t1.standardised_variant,t1.sales_amount_2020, t2.sales_amount_2021,
((t2.sales_amount_2021 - t1.sales_amount_2020)/(t1.sales_amount_2020)) * 100 as perc_inc_dec_in_sales
from t1 inner join t2 on t1.region = t2.region
and t1.channel = t2.channel and t1.standardised_variant = t2.standardised_variant;
```

Output:

region	channel	standardised_variant	sales_amount_2020	sales_amount_2021	perc_inc_dec_in_sales
APAC	Retailer	PLUS	65011361.9464	209444030.5059	222.16527117
APAC	Retailer	PREMIUM	51670538.5782	165668503.2160	220.62468822
APAC	Retailer	STANDARD	69403859.3502	202374025.2377	191.58900835
APAC	Distributor	PLUS	21011953.7263	68530600.6487	226.15054050
APAC	Distributor	PREMIUM	16638956.2104	54127257.9200	225.30440753
APAC	Distributor	STANDARD	22788692.8935	65367772.3661	186.84300882
APAC	Direct	PLUS	19095735.5566	52119306.2285	172.93688726
APAC	Direct	PREMIUM	15380562.0041	41144625.7139	167.51054807
APAC	Direct	STANDARD	20450918.1777	50609521.1720	147.46821014
EU	Retailer	PLUS	31168201.5818	120473792.8422	286.52789294
EU	Retailer	PREMIUM	25063439.3982	95160055.6232	279.67676388
EU	Retailer	STANDARD	32875103.5188	116845935.4197	255.42377944
EU	Direct	PLUS	7462525.3724	22567372.7799	202.40932732
EU	Direct	PREMIUM	6062863.7710	17855397.2767	194.50434565
EU	Direct	STANDARD	8044926.9936	21831903.3344	171.37478503
LATAM	Retailer	PLUS	813549.0641	1450104.2596	78.24422934
LATAM	Retailer	PREMIUM	623173.4846	1168576.0193	87.52017667
LATAM	Retailer	STANDARD	846153.8521	1407901.2729	66.38833108
LATAM	Direct	PLUS	716933.4041	823359.6766	14.84465250
LATAM	Direct	PREMIUM	604610.5232	641912.7860	6.16963506
LATAM	Direct	STANDARD	785250.9865	785874.3227	0.07938051
NA	Retailer	PLUS	35526330.6983	111069759.2488	212.64067261
NA	Retailer	PREMIUM	28191535.9075	86521945.1082	206.90752498
NA	Retailer	STANDARD	37881072.0775	107497011.1937	183.77499711
NA	Retailer	STANDARD	37881072.0775	107497011.1937	183.77499711
NA	Direct	PLUS	6123691.0722	17688243.4025	188.84937522
NA	Direct	PREMIUM	5005926.1049	14160946.8007	182.88365637
NA	Direct	STANDARD	6697905.6478	17303539.1597	158.34253376



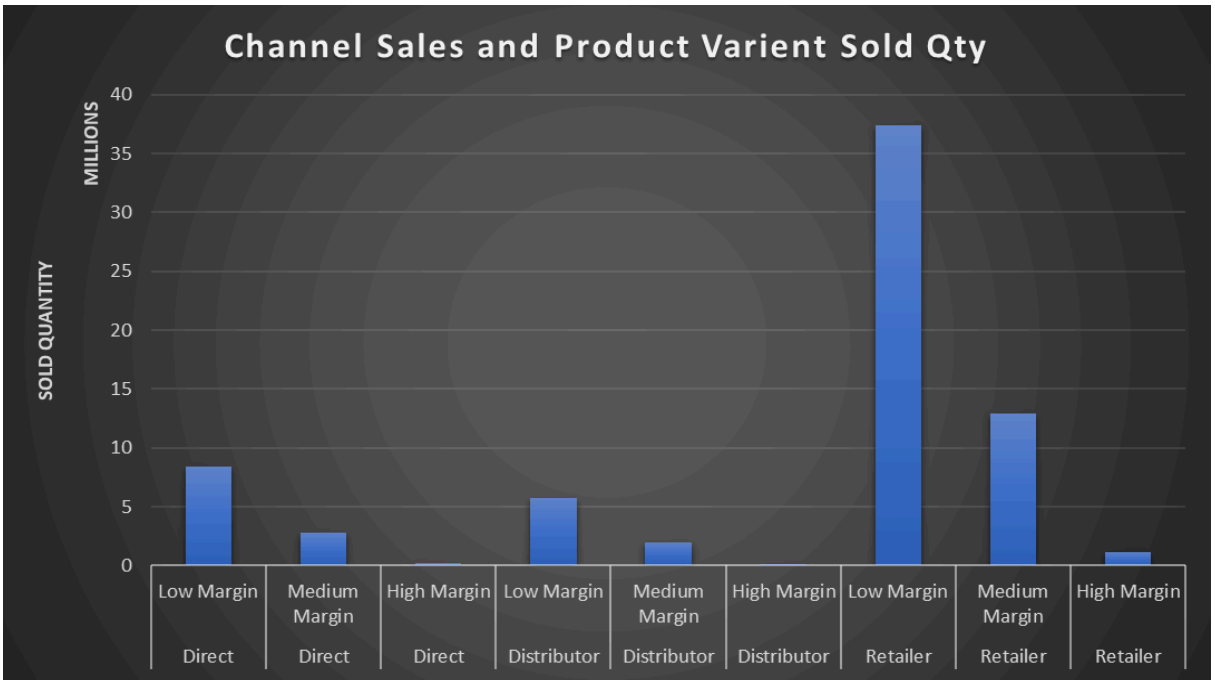
Channel Sales and Margin Category of Product.

```sql
-- channel and high margin product.
-- using product_code to avoid involving the cost of the plus and premium products.

create table pcode_margin_cat as
(with temp_table as
(select t1.product_code,t1.fiscal_year,
t1.gross_price - t2.manufacturing_cost as margin_for_product
from sales_table t1
inner join fact_manufacturing_cost t2
on t1.product_code = t2.product_code and t1.fiscal_year = t2.cost_year)
select product_code,fiscal_year,avg(margin_for_product) as avg_margin_for_product,
ntile(3) over(order by avg(margin_for_product) desc rows between unbounded preceding and unbounded following) as margin_cat_num
from temp_table
group by product_code,fiscal_year
);

alter table pcode_margin_cat
add margin_category varchar(255) not null;
```

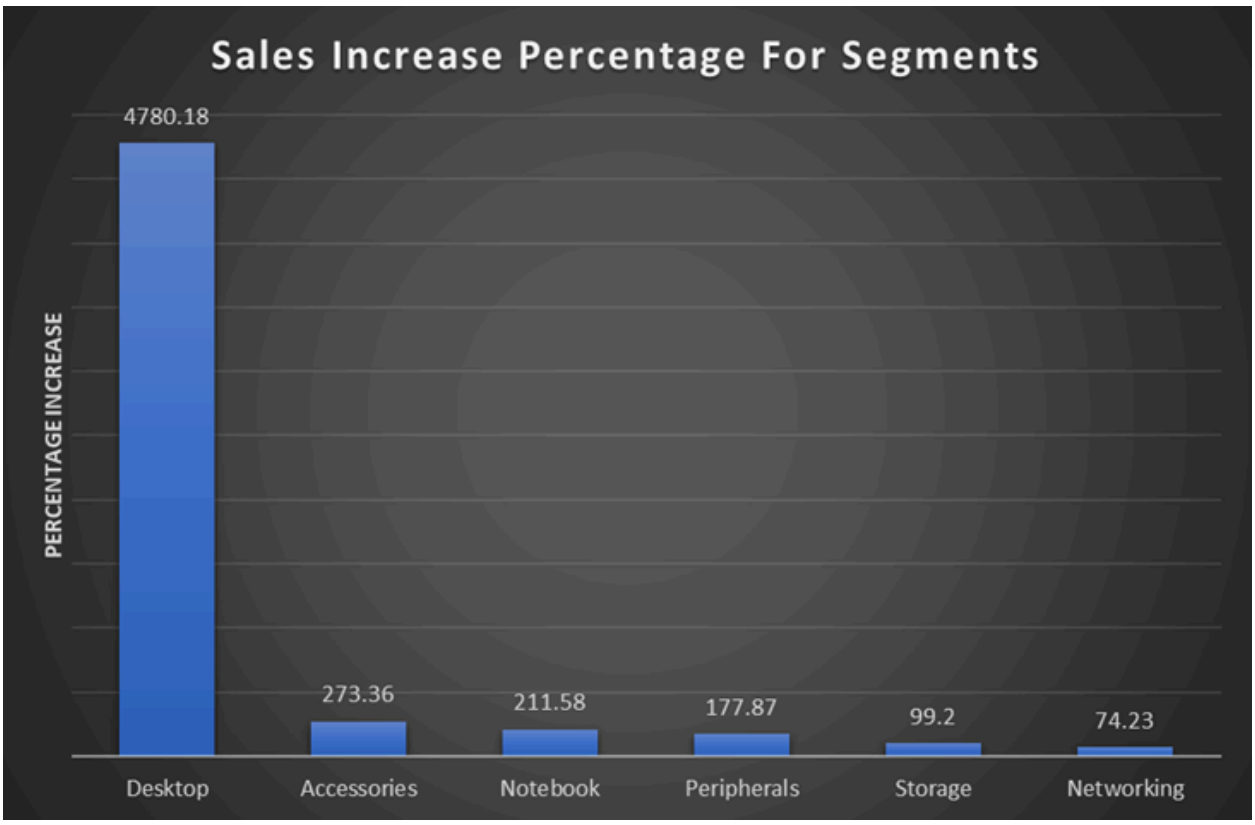
```sql
update pcode_margin_cat
set margin_category =
(case
    when margin_cat_num = 1 then 'high_margin_product'
    when margin_cat_num = 2 then 'medium_margin_product'
    when margin_cat_num = 3 then 'low_maring_product'
end);

-- channel and what kind of product are sold.
set @total_sales = (select sum(sales_amount) from sales_table);
set @total_qty_sold = (select sum(sold_quantity) from sales_table);
select t1.channel,t2.margin_category,sum(t1.sold_quantity) as sold_qty,
round((sum(t1.sales_amount)/(@total_sales)) * 100,2) as perc_contribution_to_sales,
round((sum(t1.sold_quantity)/(@total_qty_sold)) * 100,2) as perc_contribution_to_qty_sold
from sales_table t1 inner join
pcode_margin_cat t2 on t1.product_code = t2.product_code and t1.fiscal_year = t2.fiscal_year
group by t1.channel,t2.margin_category
order by channel asc,sold_qty desc;
```

Output:

channel	margin_category	sold_qty	perc_contribution_to_sales	perc_contribution_to_qty_sold
Direct	low_maring_product	8410786	4.80	11.86
Direct	medium_margin_product	2830645	6.11	3.99
Direct	high_margin_product	249392	5.17	0.35
Distributor	low_maring_product	5789744	3.33	8.16
Distributor	medium_margin_product	1994449	4.28	2.81
Distributor	high_margin_product	176834	3.68	0.25
Retailer	low_maring_product	37431735	21.40	52.77
Retailer	medium_margin_product	12917084	27.59	18.21
Retailer	high_margin_product	1136802	23.64	1.60

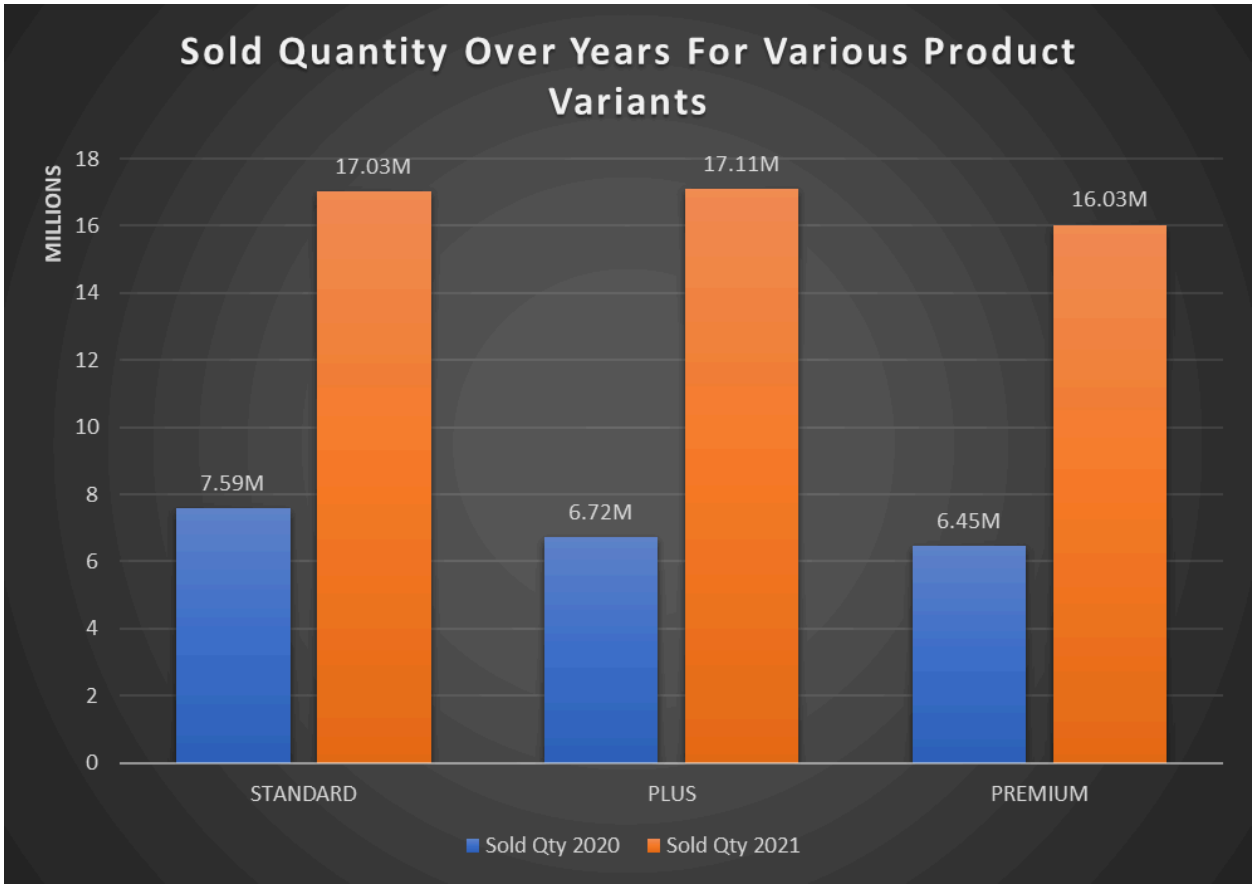


How the sales have increased/Decreased for various Product Segments Over years 2020 and 2021.



For each product (Standard / Plus / Premium), what is the share of sales volume and how has it shifted over the last 2 years?

```sql
-- For each product (Standard / Plus / Premium), what is the share of sales
-- volume and how has it shifted over the last 2 years?
with standardised_variant_qty_sold_2020 as
(select standardised_variant,sum(sold_quantity) as sold_quantity_2020
from sales_table where fiscal_year = 2020
group by standardised_variant),
standardised_variant_qty_sold_2021 as
(select standardised_variant,sum(sold_quantity) as sold_quantity_2021
from sales_table where fiscal_year = 2021
group by standardised_variant)
select t1.standardised_variant,t1.sold_quantity_2020,t2.sold_quantity_2021
from standardised_variant_qty_sold_2020 t1
inner join standardised_variant_qty_sold_2021 t2 on
t1.standardised_variant = t2.standardised_variant;
```


Sold Quantity Over Years For Various Product Variants
MILLIONS
18
16
14
12
10
8
6
4
2
0
17.03M
17.11M
16.03M
7.59M
6.72M
6.45M
STANDARD
PLUS
PREMIUM
Sold Qty 2020
Sold Qty 2021

Insights:

P & A division contributes highest to the sales.

In the accessories segment the highest number of products were added from 2020-2021 34 New products were added.

Sales amount increased highest by value for notebook, accessories and peripherals segment from 2020 to 2021.

Sales amount increased the highest by percentage for the Desktop segment from 2020 to 2021.

Sales amount will increase the highest for categories - keyboard,personal laptops, business laptops,mouse,processors etc. from 2020 to 2021.

Sales amount increased the highest by percentage for the categories batteries, personal laptops, mother board, from 2020 to 2021.

Sales amount will increase the highest for division - P & A.

Sales amount increased the highest by percentage for the division PC from 2020 to 2021.

Prices are mostly reduced for the product which are from the plus or premium product. Might be due to covid 19 because of reduced demand.

Average selling price increased the highest for the PC Division(Personal Computer Division) mostly because of the need of covid 19 effect.

For the sale of any kind of laptop or notebook we earn the highest profit per unit.

We had higher sales in 2021 because of the higher number of PC division product were sold whose prices have also increased as compared to 2020 and profit per product for this division product is higher.

Most of the products with the higher cost to price ratio are mostly plus or premium products.

These higher cost to sales ratio products contribute 48.58 % of total_sales and 52.74 % of total quantity sold.

Might need to reduce the dependence on the higher cost to price products.

For Latin America Region, For Direct sales Channel the quantity sold has reduced from 2020 to 2021.

For the Latin America Region , For Direct sales channel the sales amount has grown almost nothing for all product variants as compared to other regions and respective sales channels.

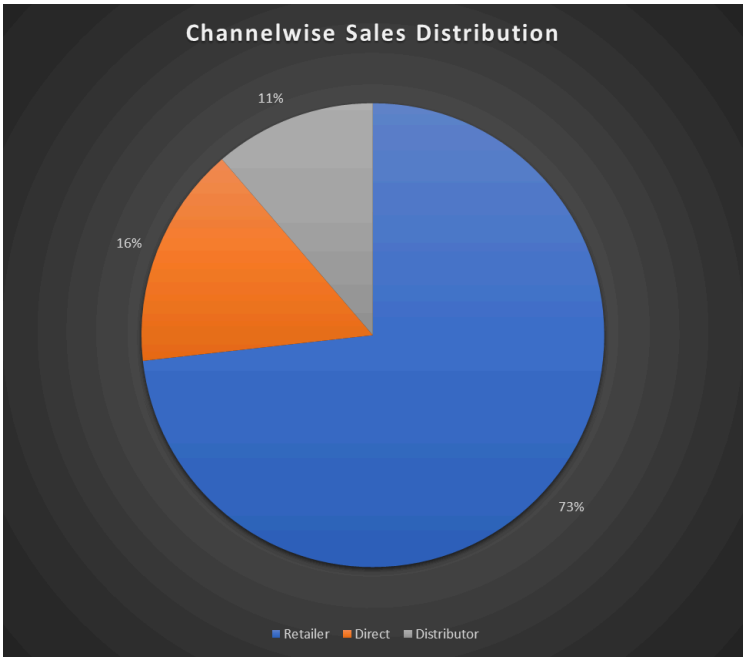
Channel Analysis

Which sales channels (Retailers, Direct, Distributors) generate the most revenue and which bring in higher margins?

```sql
-- Which sales channels (Retailers, Direct, Distributors) generate the most revenue and which bring in higher margins?
select channel,
sum(sales_amount) as sales,
round((sum(sales_amount)/(select sum(sales_amount) from sales_table)) * 100,2) as perc_contribution
from sales_table
group by channel
order by sum(sales_amount) desc;
```

Output:

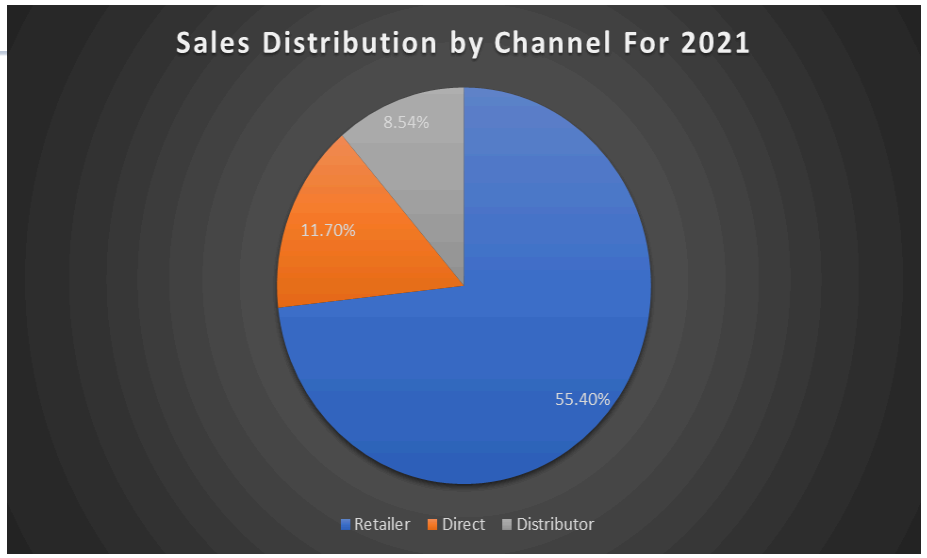
channel	gross_sales	perc_contribution
Retailer	1219081639.95	73.23
Direct	257532002.65	15.47
Distributor	188025630.93	11.30



Which channel helped to bring more gross sales in the fiscal year 2021 and the percentage of contribution?

```sql
-- Which channel helped to bring more gross sales in the fiscal year 2021 and the percentage of contribution?
set @total_sales_amount = (select sum(sales_amount) from sales_amount);
select channel,(sum(sales_amount)/(@total_sales_amount)) * 100 as sales_dist_by_channel_2021 from sales_table where fiscal_year = 2021
group by channel order by sales_dist_by_channel_2021 desc;
```

channel	sales_dist_by_channel_2021
Retailer	55.39806982
Direct	11.70288797
Distributor	8.54434739

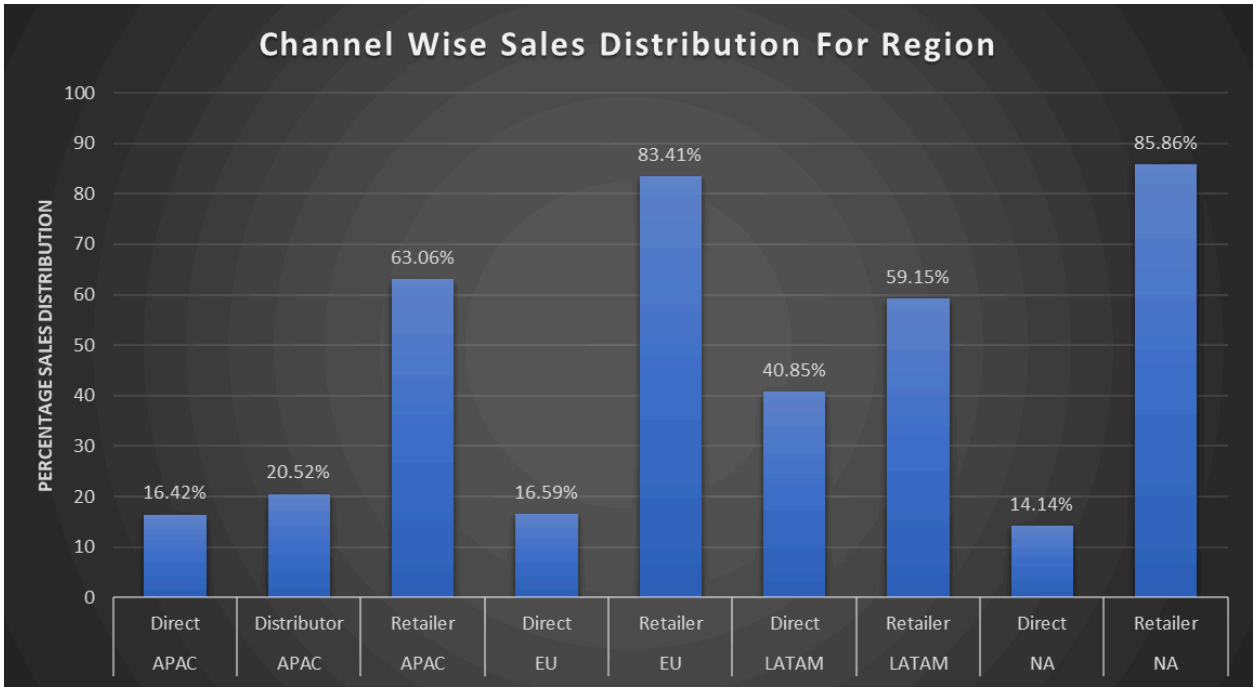


What % of sales in each region is tied to a single channel (e.g., Retailers or Distributors)?

```sql
-- Question: What % of sales in each region is tied to a single channel (e.g., Retailers or Distributors)?
with temp_table as
(select region, channel,
sum(sales_amount) as region_channel_sales
from sales_table
group by region,channel)
select *,
sum(region_channel_sales) over(partition by region) as region_total_sales,
round((region_channel_sales/sum(region_channel_sales) over(partition by region)) * 100,2) as per_c_to_overall_regional_sales
from temp_table
order by region,channel asc ;
-- across all the regions retailer channel brings the most revenue.
```

Output:

region	channel	region_channel_sales	region_total_sales	per_c_to_overall_regional_sales
APAC	Direct	198800668.8528	1210838221.4522	16.42
APAC	Distributor	248465233.7650	1210838221.4522	20.52
APAC	Retailer	763572318.8344	1210838221.4522	63.06
EU	Direct	83824989.5280	505411517.9119	16.59
EU	Retailer	421586528.3839	505411517.9119	83.41
LATAM	Direct	4357941.6991	10667399.6517	40.85
LATAM	Retailer	6309457.9526	10667399.6517	59.15
NA	Direct	66980252.1878	473667906.4218	14.14
NA	Retailer	406687654.2340	473667906.4218	85.86

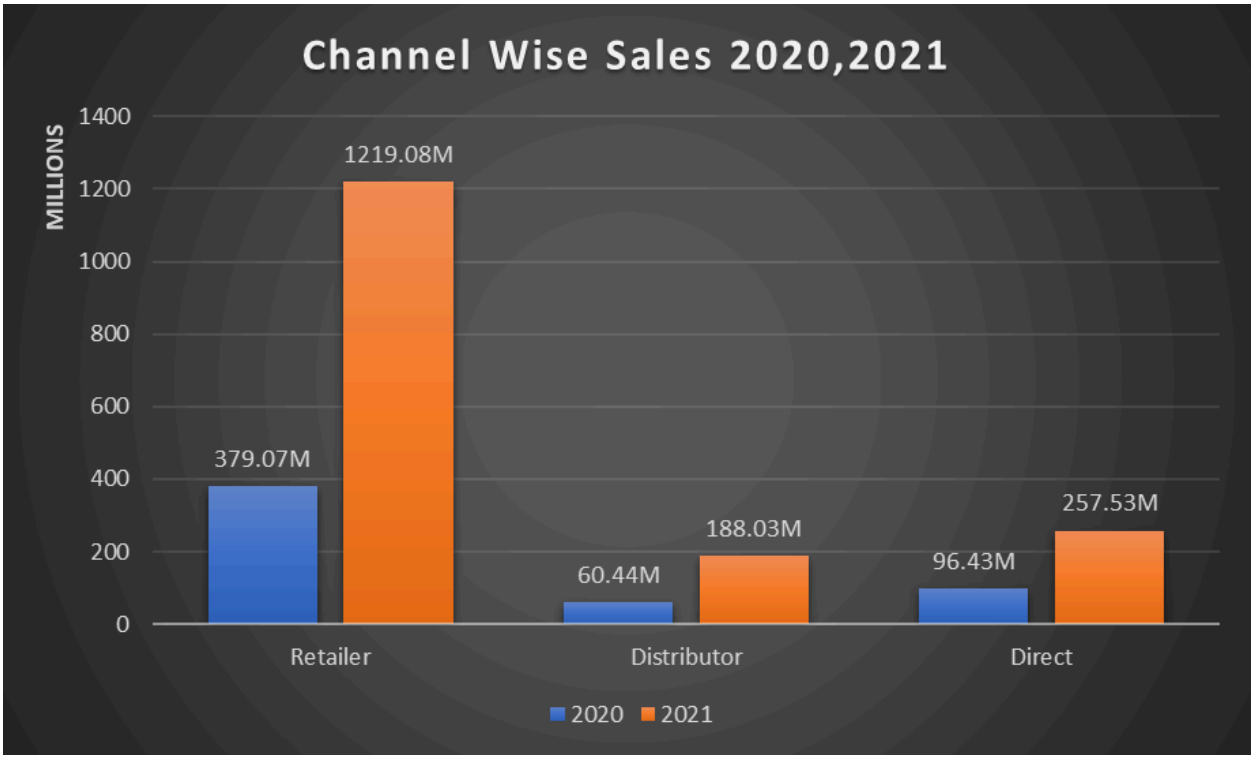


Channel Wise Sales in 2020 and 2021.

```
with channel_sales_2020 as
(select channel,sum(sales_amount) as total_sales_2020 from sales_table
where fiscal_year = 2020
group by channel),
channel_sales_2021 as
(select channel,sum(sales_amount) as total_sales_2021 from sales_table
where fiscal_year = 2021
group by channel)
select t1.channel,t1.total_sales_2020,t2.total_sales_2021,
((t2.total_sales_2021 - t1.total_sales_2020)/(t1.total_sales_2020)) * 100 as sales_inc_by_channel_perc
from channel_sales_2020 t1 inner join
channel_sales_2021 t2 on t1.channel = t2.channel
order by sales_inc_by_channel_perc desc;
```

Output:

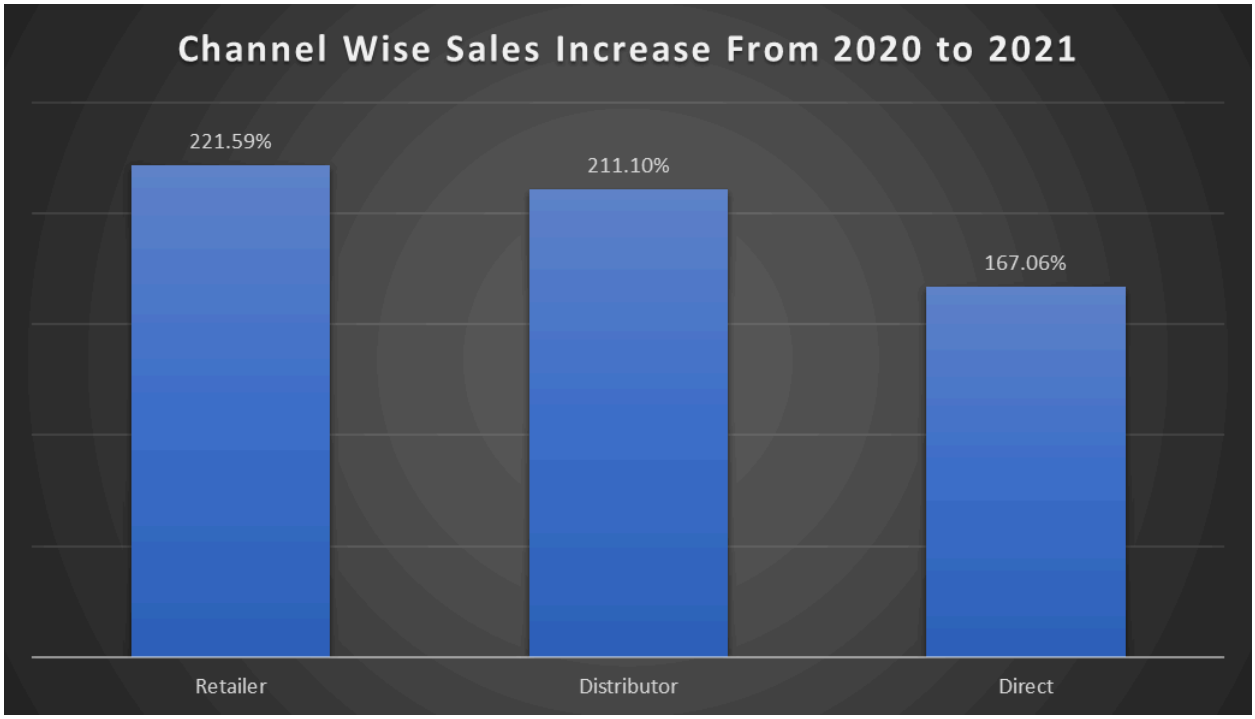
channel	total_sales_2020	total_sales_2021	sales_inc_by_channel_perc
Retailer	379074319.4577	1219081639.9472	221.59436221
Distributor	60439602.8302	188025630.9348	211.09673481
Direct	96431849.6141	257532002.6536	167.06114596



Channel Wise Sales Growth from 2020 to 2021.

```
with channel_sales_2020 as
(select channel,sum(sales_amount) as total_sales_2020 from sales_table
where fiscal_year = 2020
group by channel),
channel_sales_2021 as
(select channel,sum(sales_amount) as total_sales_2021 from sales_table
where fiscal_year = 2021
group by channel)
select t1.channel,t1.total_sales_2020,t2.total_sales_2021,
((t2.total_sales_2021 - t1.total_sales_2020)/(t1.total_sales_2020)) * 100 as sales_inc_by_channel_perc
from channel_sales_2020 t1 inner join
channel_sales_2021 t2 on t1.channel = t2.channel
order by sales_inc_by_channel_perc desc;
```

Output:



Channel and Region. covered.

Channel and Product covered earlier.

Channel and Region YoY. covered earlier in the regional analysis section.

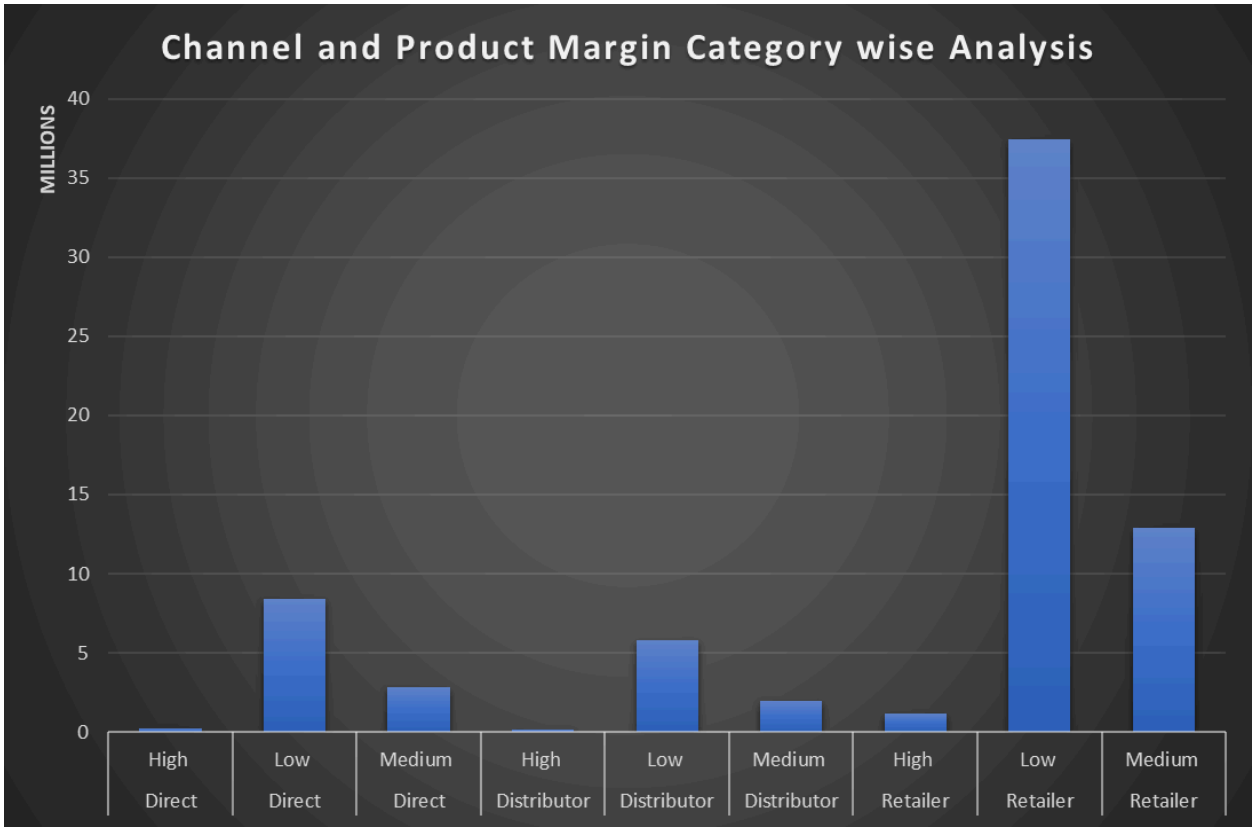
Channel and Product Variant.

Channel, Product margin category

```
-- channel, product_ margin category
select t1.channel,t2.margin_category,sum(sold_quantity) as sold_qty from sales_table t1 inner join
pcode_margin_cat t2 on t1.product_code = t2.product_code
and t1.fiscal_year = t2.fiscal_year
group by t1.channel,t2.margin_category;
```

channel	margin_category	sold_qty
Direct	high_margin_product	249392
Direct	low_maring_product	8410786
Direct	medium_margin_product	2830645
Distributor	high_margin_product	176834
Distributor	low_maring_product	5789744
Distributor	medium_margin_product	1994449
Retailer	high_margin_product	1136802
Retailer	low_maring_product	37431735
Retailer	medium_margin_product	12917084

Output:

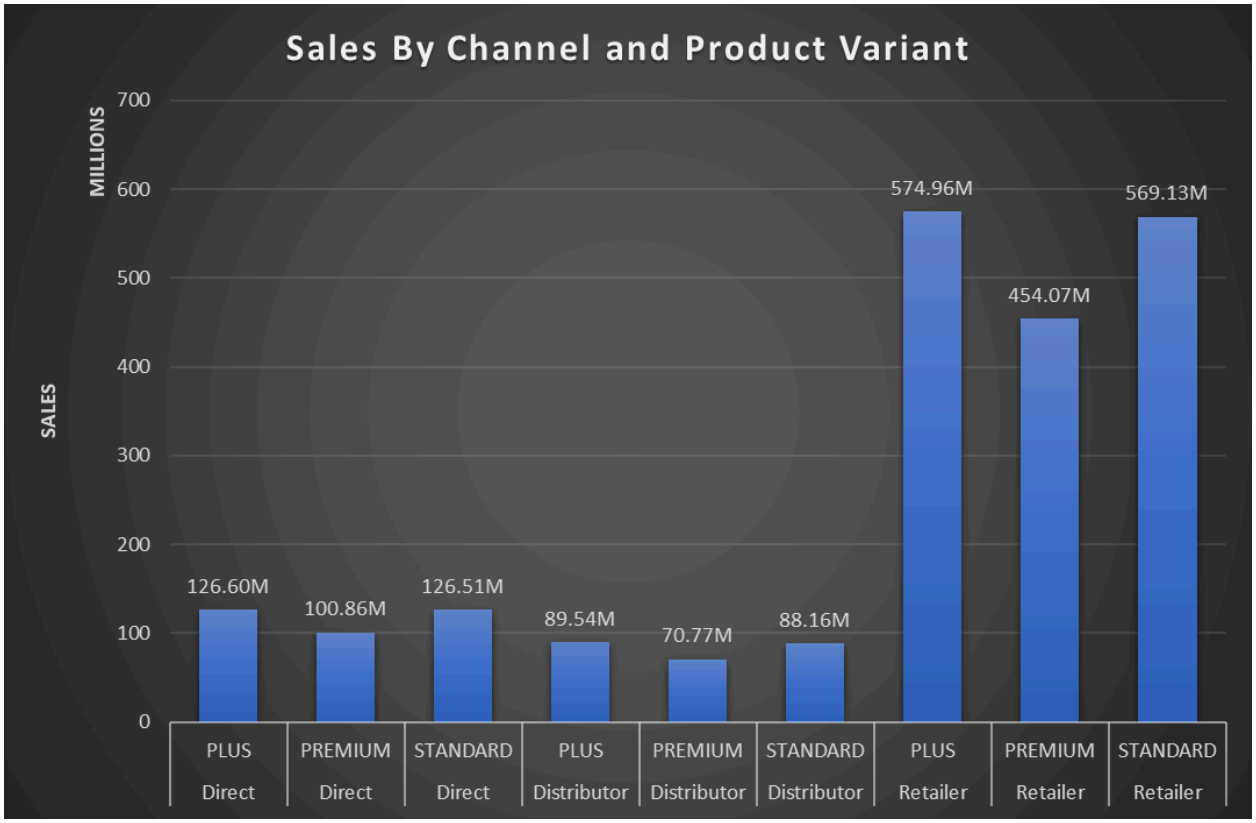


Channel Variant Analysis.

```
select channel,standardised_variant,sum(sales_amount) as sales_amt from sales_table
group by channel,standardised_variant
order by channel;
```

Output:

channel	standardised_variant	sales_amt
Direct	PLUS	126597167.4928
Direct	PREMIUM	100856844.9805
Direct	STANDARD	126509839.7944
Distributor	PLUS	89542554.3750
Distributor	PREMIUM	70766214.1304
Distributor	STANDARD	88156465.2596
Retailer	PLUS	574957130.1471
Retailer	PREMIUM	454067767.3352
Retailer	STANDARD	569131061.9226

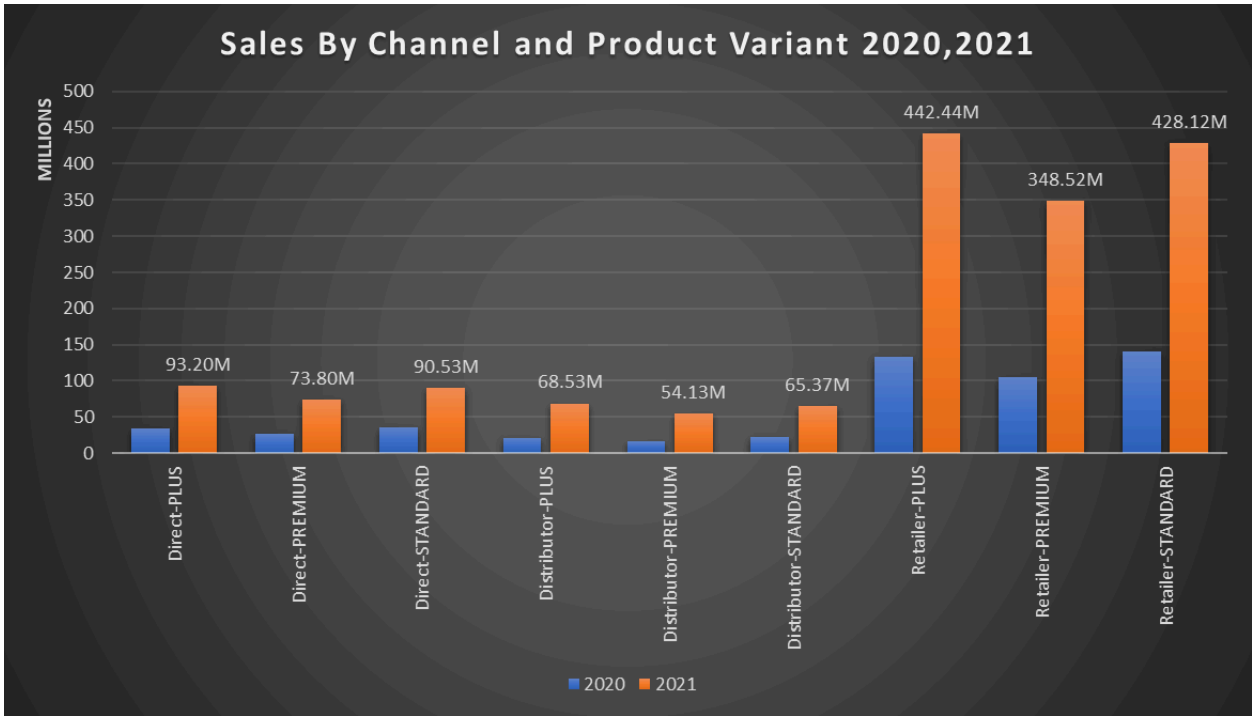


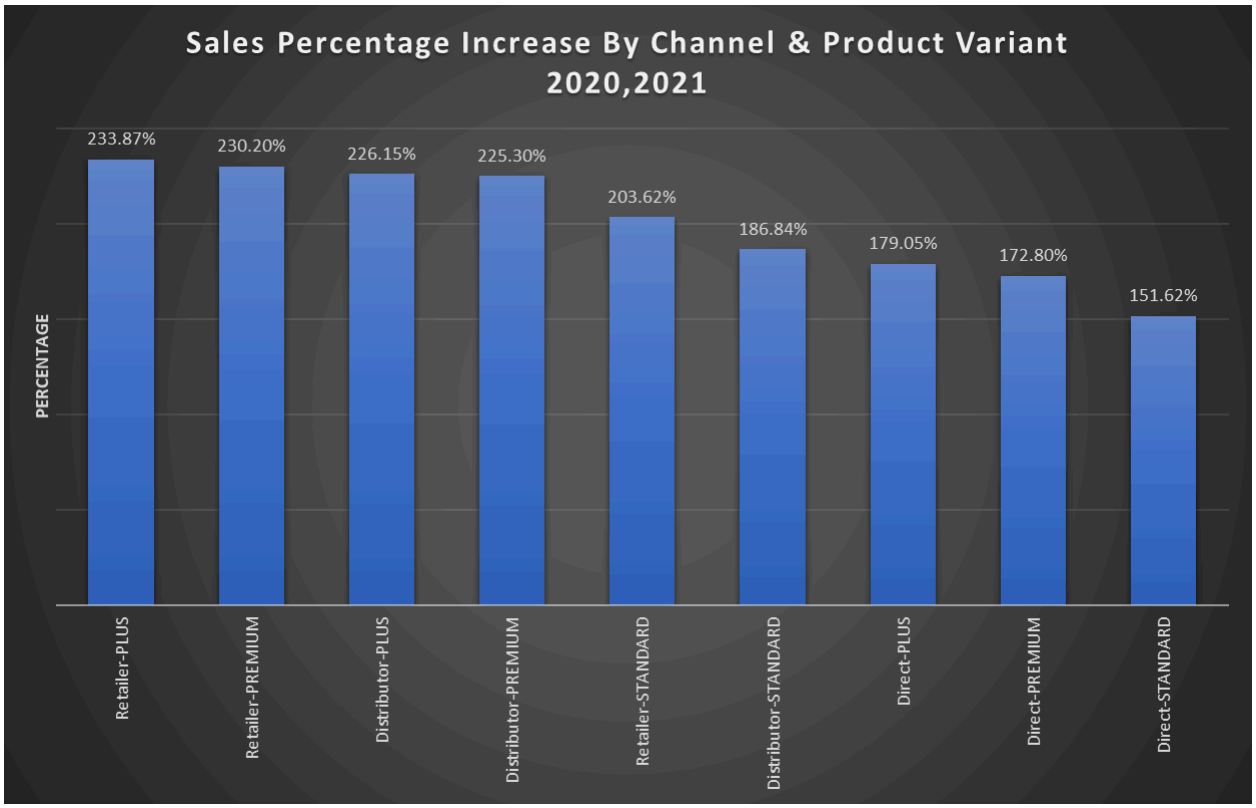
Channel and Variant Analysis YoY

```
with sales_2020 as
(select channel,standardised_variant,sum(sales_amount) as sales_amt_2020 from sales_table
where fiscal_year = 2020
group by channel,standardised_variant
order by channel
),

sales_2021 as
(
select channel,standardised_variant,sum(sales_amount) as sales_amt_2021 from sales_table
where fiscal_year = 2021
group by channel,standardised_variant
order by channel
)

select t1.channel,t1.standardised_variant,t1.sales_amt_2020,t2.sales_amt_2021,
t2.sales_amt_2021 - t1.sales_amt_2020 as inc_or_dec_in_sales_amt,
((t2.sales_amt_2021 - t1.sales_amt_2020)/(t1.sales_amt_2020)) * 100 as perc_inc_or_dec_in_sales
from sales_2020 t1 inner join sales_2021 t2 on
t1.channel = t2.channel and t1.standardised_variant = t2.standardised_variant;
```

Output:

channel	standardised_variant	sales_amt_2020	sales_amt_2021	inc_or_dec_in_sales_amt	perc_inc_or_dec_in_sales
Direct	PLUS	33398885.4053	93198282.0875	59799396.6822	179.04608479
Direct	PREMIUM	27053962.4032	73802882.5773	46748920.1741	172.79879183
Direct	STANDARD	35979001.8056	90530837.9888	54551836.1832	151.62131645
Distributor	PLUS	21011953.7263	68530600.6487	47518646.9224	226.15054050
Distributor	PREMIUM	16638956.2104	54127257.9200	37488301.7096	225.30440753
Distributor	STANDARD	22788692.8935	65367772.3661	42579079.4726	186.84300882
Retailer	PLUS	132519443.2906	442437686.8565	309918243.5659	233.86624322
Retailer	PREMIUM	105548687.3685	348519079.9667	242970392.5982	230.19745546
Retailer	STANDARD	141006188.7986	428124873.1240	287118684.3254	203.62133519


Sales By Channel and Product Variant 2020,2021
MILLIONS
500
450
400
350
300
250
200
150
100
50
0
93.20M
73.80M
90.53M
68.53M
54.13M
65.37M
442.44M
348.52M
428.12M
Direct-PLUS
Direct-PREMIUM
Direct-STANDARD
Distributor-PLUS
Distributor-PREMIUM
Distributor-STANDARD
Retailer-PLUS
Retailer-PREMIUM
Retailer-STANDARD
2020
2021


Sales Percentage Increase By Channel & Product Variant 2020,2021
PERCENTAGE
233.87%
230.20%
226.15%
225.30%
203.62%
186.84%
179.05%
172.80%
151.62%
Retailer-PLUS
Retailer-PREMIUM
Distributor-PLUS
Distributor-PREMIUM
Retailer-STANDARD
Distributor-STANDARD
Direct-PLUS
Direct-PREMIUM
Direct-STANDARD

Market ,Segment and Discount Analysis

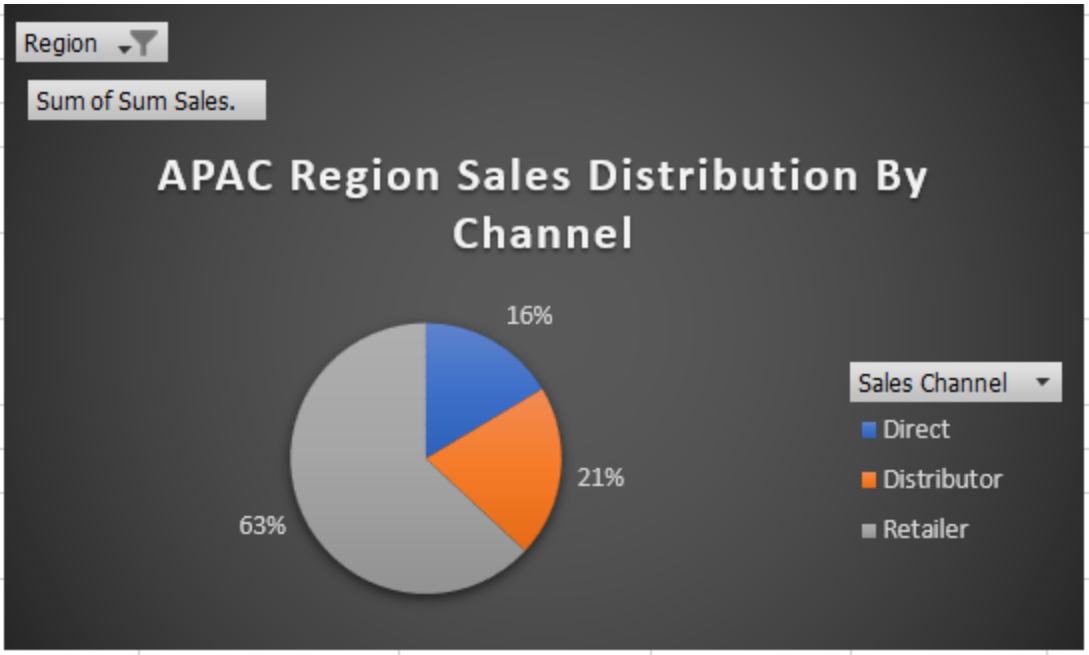
```
-- market, segment, discount analysis.
select t1.market, t1.segment, avg(t2.pre_invoice_discount_pct) as avg_discount_perc
from sales_table t1 inner join fact_pre_invoice_deductions t2
on t1.customer_code = t2.customer_code and t1.fiscal_year = t2.fiscal_year
group by t1.market,t1.segment
order by avg_discount_perc desc;
```

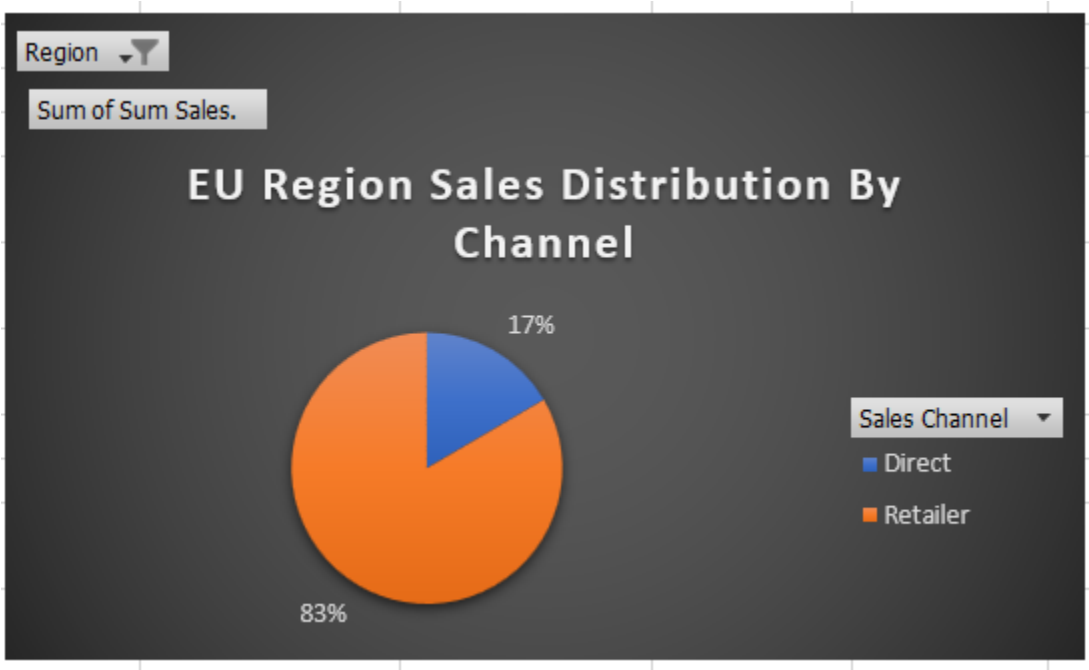
Not significantly different.

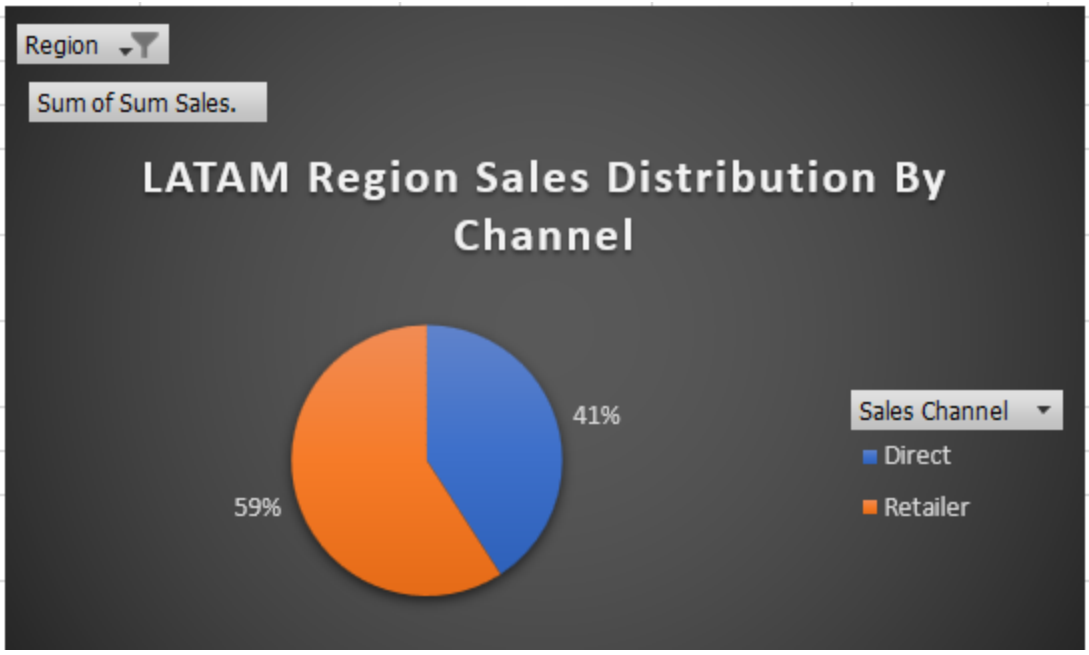
Channel Dominance

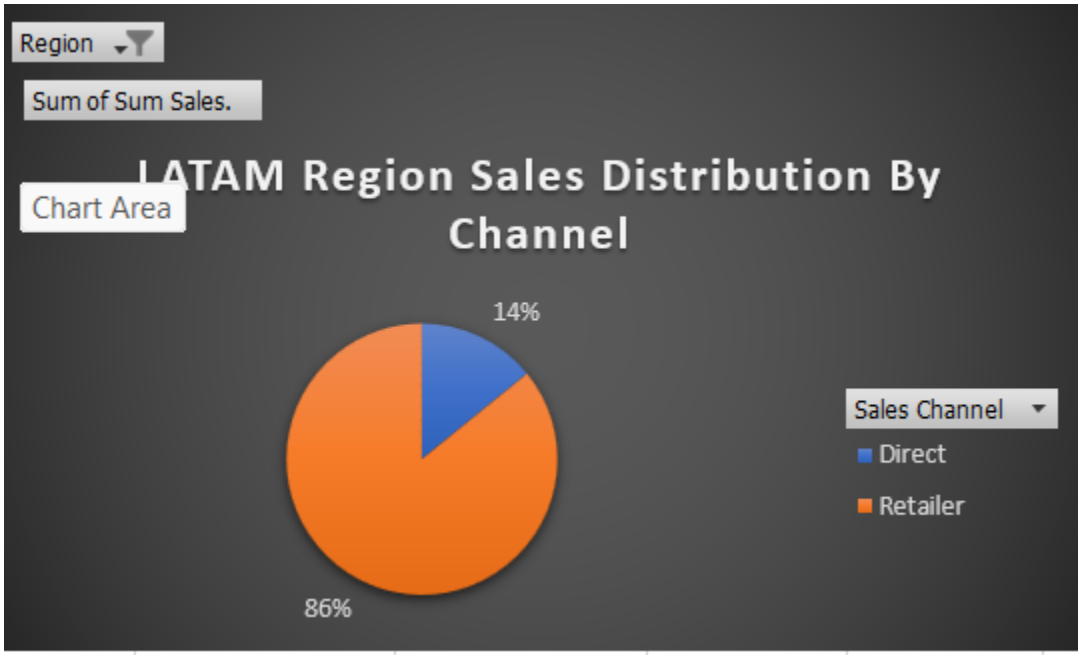
```
-- channel dominance

select region,channel,sum(sales_amount) as sales_amount from sales_table
group by region, channel;
```

region	channel	sales_amount
APAC	Direct	198800668.8528
EU	Direct	83824989.5280
NA	Direct	66980252.1878
LATAM	Direct	4357941.6991
APAC	Distributor	248465233.7650
APAC	Retailer	763572318.8344
EU	Retailer	421586528.3839
NA	Retailer	406687654.2340
LATAM	Retailer	6309457.9526


Region
Sum of Sum Sales.
APAC Region Sales Distribution By Channel
16%
21%
63%
Sales Channel
Direct
Distributor
Retailer


Region
Sum of Sum Sales.
EU Region Sales Distribution By Channel
17%
83%
Sales Channel
Direct
Retailer





Insights:

73% of total sales happens by retail channel.

All the channels showed growth in sales but the retailer channel grew the largest along with the distributor channel.

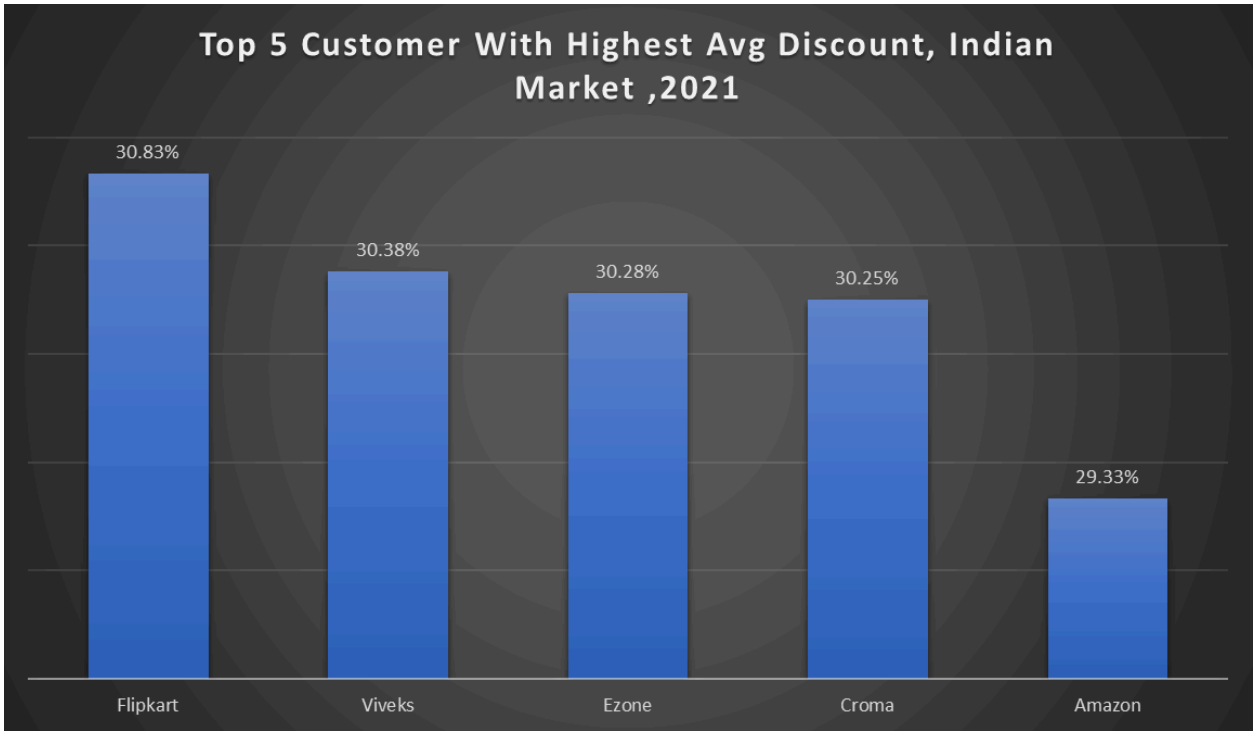
Discount Analysis

Generate a report which contains the top 5 customers who received an average high pre_invoice_discount_pct for the fiscal year 2021 and in the Indian market.

```
select t1.customer_code, t1.customer,
concat(round(avg(t2.pre_invoice_discount_pct)*100,2),' % ') as avg_pre_invoice_discount_prec
from dim_customer t1 inner join fact_pre_invoice_deductions t2
on t1.customer_code = t2.customer_code
where fiscal_year = 2021 and sub_zone = 'India'
group by t1.customer_code,t1.customer
order by avg(t2.pre_invoice_discount_pct) desc limit 5;
```

Output:

customer_code	customer	avg_pre_invoice_discount_prec
90002009	Flipkart	30.83 %
90002006	Viveks	30.38 %
90002003	Ezone	30.28 %
90002002	Croma	30.25 %
90002016	Amazon	29.33 %

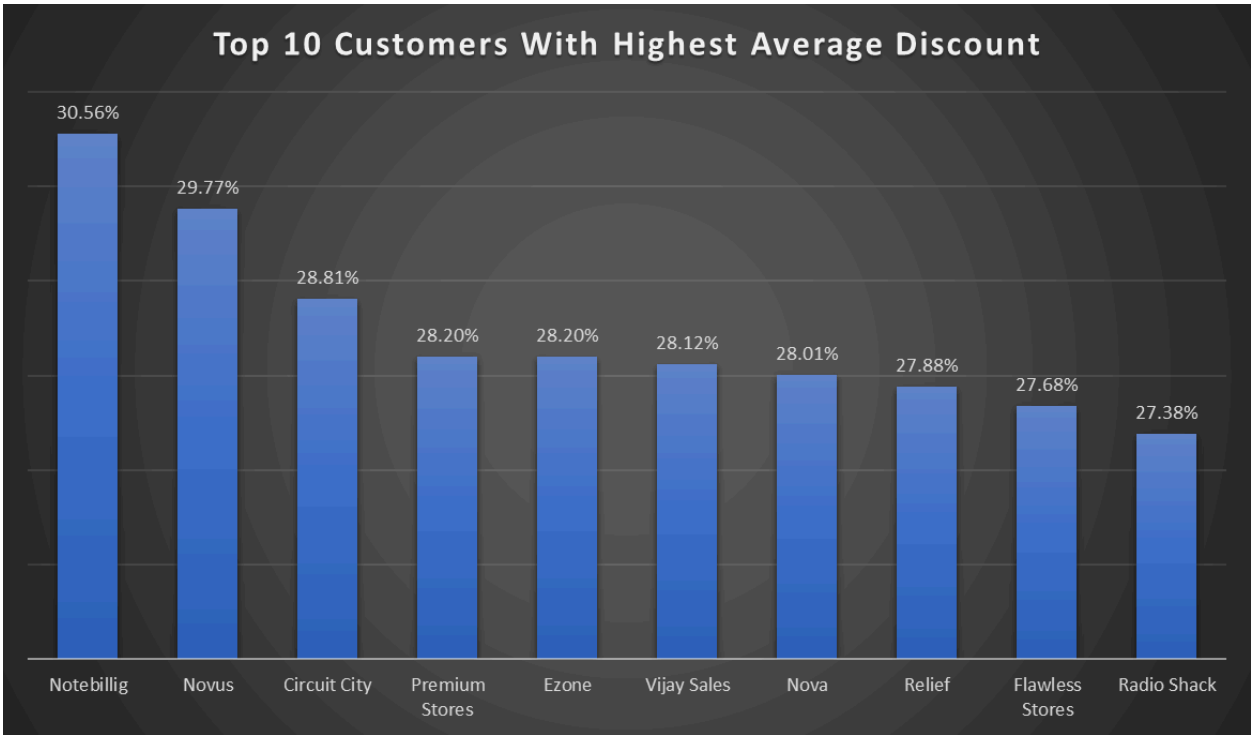


How much discounting are we doing by customer and region, and what is its impact on margins?

```
create view discount_info as
(select t1.*,
t2.customer, t2.region,t2.market
from fact_pre_invoice_deductions t1 inner join dim_customer t2 on t1.customer_code = t2.customer_code);

select customer, avg(pre_invoice_discount_pct) as avg_pre_invoice_discount_pct from discount_info
group by customer
order by avg(pre_invoice_discount_pct) desc limit 10;
```

Output:

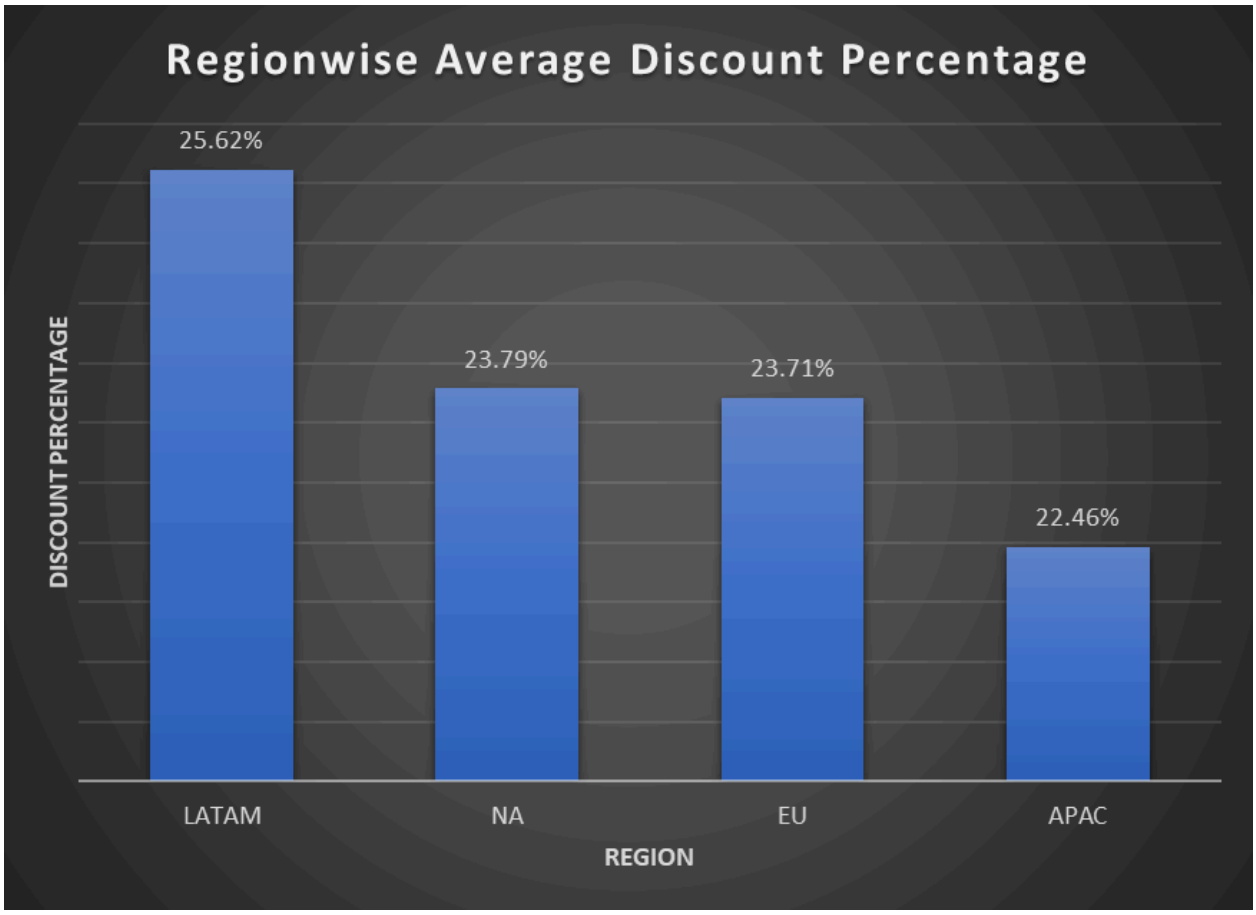
customer	avg_pre_invoice_discount_pct
Notebillig	0.30560000
Novus	0.29765000
Circuit City	0.28810000
Premium Stores	0.28197500
Ezone	0.28195000
Vijay Sales	0.28115000
Nova	0.28005000
Relief	0.27882500
Flawless Stores	0.27675000
Radio Shack	0.27380000



Discount and Region

```
-- discount and region.
select region, avg(pre_invoice_discount_pct) as avg_pre_invoice_discount_pct from discount_info
group by region
order by avg(pre_invoice_discount_pct) desc;
```

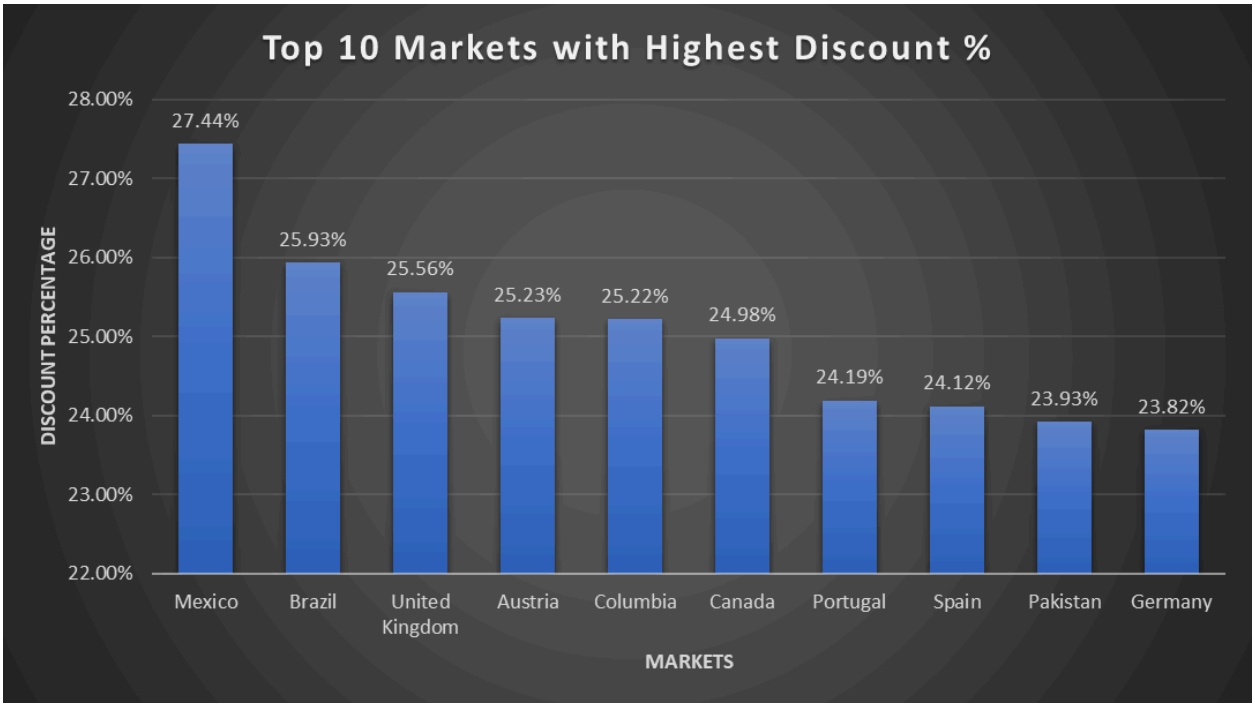
region	avg_pre_invoice_discount_pct
LATAM	0.25616429
NA	0.23792308
EU	0.23712000
APAC	0.22463451



Discount and Market.

```
-- discount and market
select market, avg(pre_invoice_discount_pct) as avg_pre_invoice_discount_pct from discount_info
group by market
order by avg(pre_invoice_discount_pct) desc limit 10;
```

market	avg_pre_invoice_discount_pct
Mexico	0.27440000
Brazil	0.25930000
United Kingdom	0.25562727
Austria	0.25234375
Columbia	0.25215000
Canada	0.24977727
Portugal	0.24189167
Spain	0.24116364
Pakistan	0.23925000
Germany	0.23815909

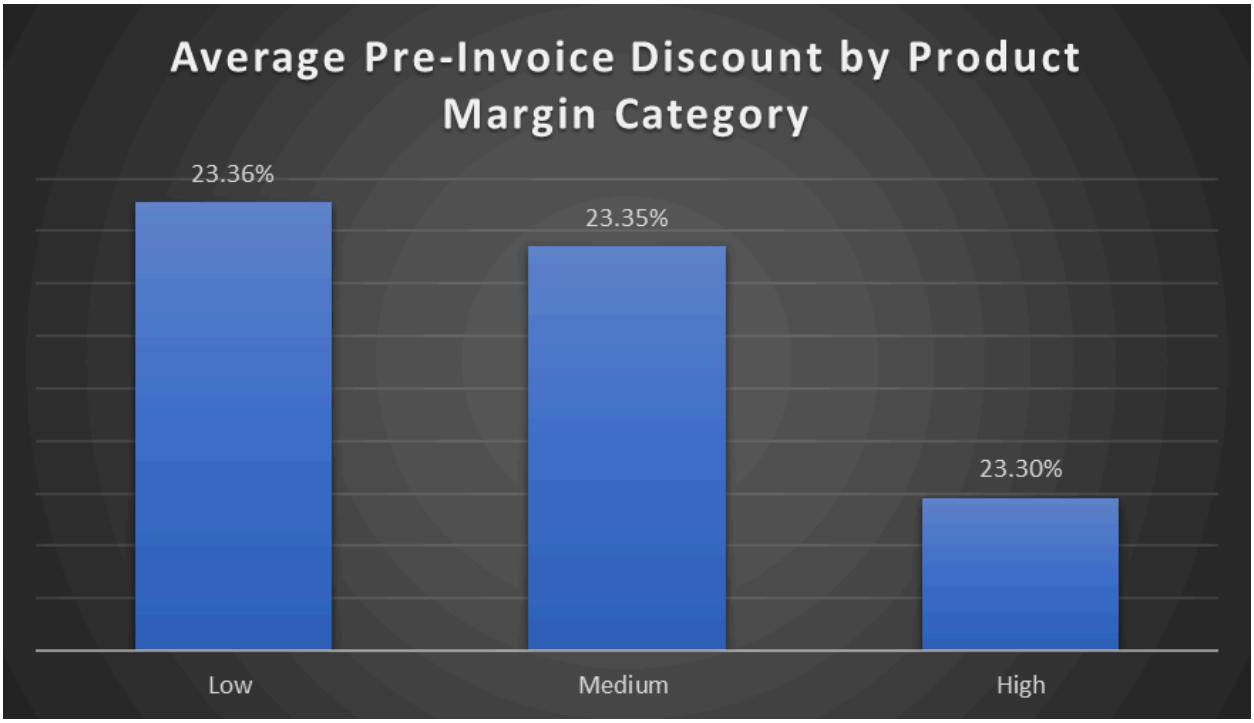


Discount and the product margin category.

```
select t2.margin_category, avg(t3.pre_invoice_discount_pct) as avg_discount_perc from sales_table t1 inner join pcode_margin_cat t2
on t1.product_code = t2.product_code and t1.fiscal_year = t2.fiscal_year inner join
fact_pre_invoice_deductions t3 on t1.customer_code = t3.customer_code and t1.fiscal_year = t3.fiscal_year
group by t2.margin_category order by avg_discount_perc desc;
```

Output:

margin_category	avg_discount_perc
low_maring_product	0.23355500
medium_margin_product	0.23347031
high_margin_product	0.23299212



Discount and Division

```
-- discount and division.
select t1.division,avg(t2.pre_invoice_discount_pct) as avg_discount_perc from sales_table t1 inner join fact_pre_invoice_deductions t2 on
t1.customer_code = t2.customer_code and t1.fiscal_year = t2.fiscal_year
group by t1.division order by avg_discount_perc desc;
```

Output:

division	avg_discount_perc
N & S	0.23364841
P & A	0.23354542
PC	0.23299637

Discount , Division and Market.

```
-- discount and division.
select t1.division,avg(t2.pre_invoice_discount_pct) as avg_discount_perc from sales_table t1
inner join fact_pre_invoice_deductions t2 on
t1.customer_code = t2.customer_code and t1.fiscal_year = t2.fiscal_year
group by t1.division order by avg_discount_perc desc;
```

Insight: For a market discount percentage remains nearly the same for different divisions.

Discount, Market and Product Category

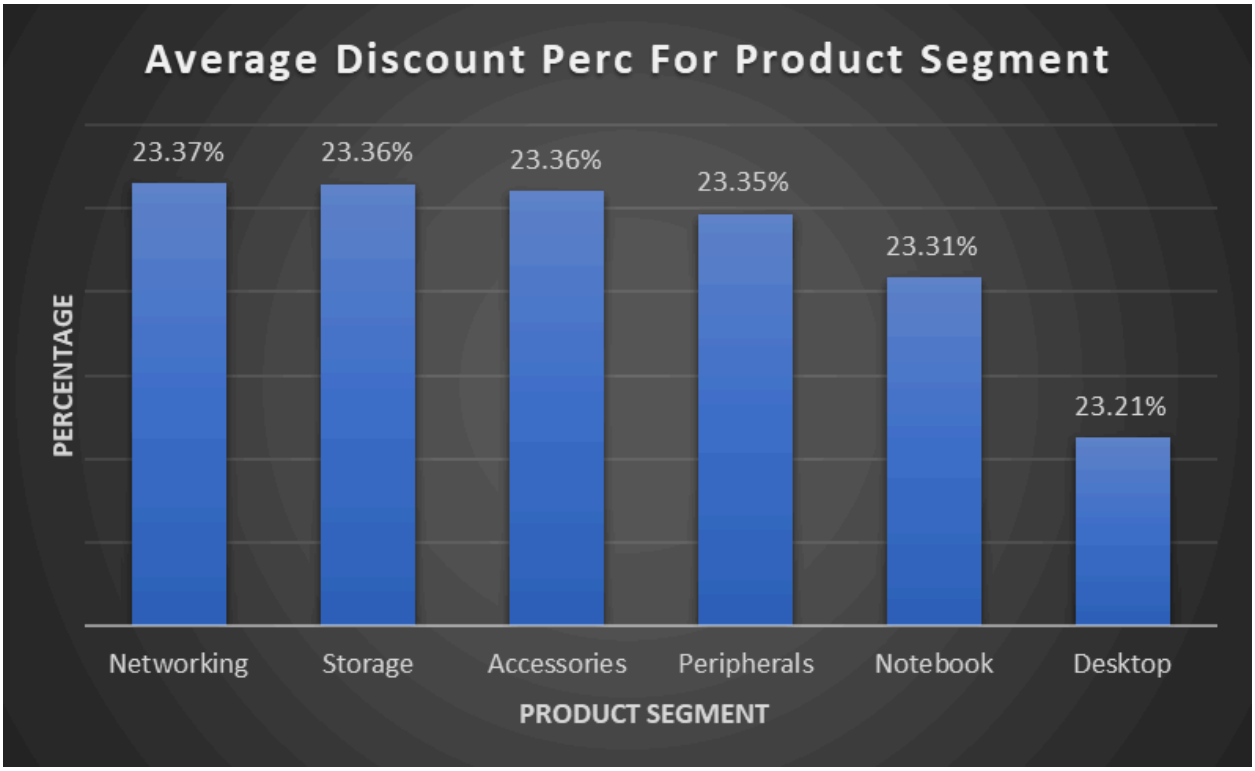
```
-- which product category gets the most discount in each market.
-- simply discount,category and market.
select t1.market,t1.category,avg(t2.pre_invoice_discount_pct) as avg_discount_perc from sales_table t1
inner join fact_pre_invoice_deductions t2 on
t1.customer_code = t2.customer_code and t1.fiscal_year = t2.fiscal_year
group by t1.category, t1.market order by t1.market asc;
```

Insight: For a market discount percentage remains nearly the same for different product categories. If changes it changes by around 1 or 2 percent mostly for computer peripherals devices.(External Solid State Drive) Categories.

Discount , Product Segment.

```
-- discount and product segment
select t1.segment,avg(t2.pre_invoice_discount_pct) as avg_discount_perc
from sales_table t1 inner join fact_pre_invoice_deductions t2 on
t1.customer_code = t2.customer_code and t1.fiscal_year = t2.fiscal_year
group by t1.segment order by avg_discount_perc desc;
```

segment	avg_discount_perc
Networking	0.23365025
Storage	0.23364747
Accessories	0.23360669
Peripherals	0.23346847
Notebook	0.23308913
Desktop	0.23212991



Discount and Product Category

```
-- discount and product category
select t1.category,avg(t2.pre_invoice_discount_pct) as avg_discount_perc
from sales_table t1 inner join fact_pre_invoice_deductions t2 on
t1.customer_code = t2.customer_code and t1.fiscal_year = t2.fiscal_year
group by t1.category order by avg_discount_perc desc;
```

Output:

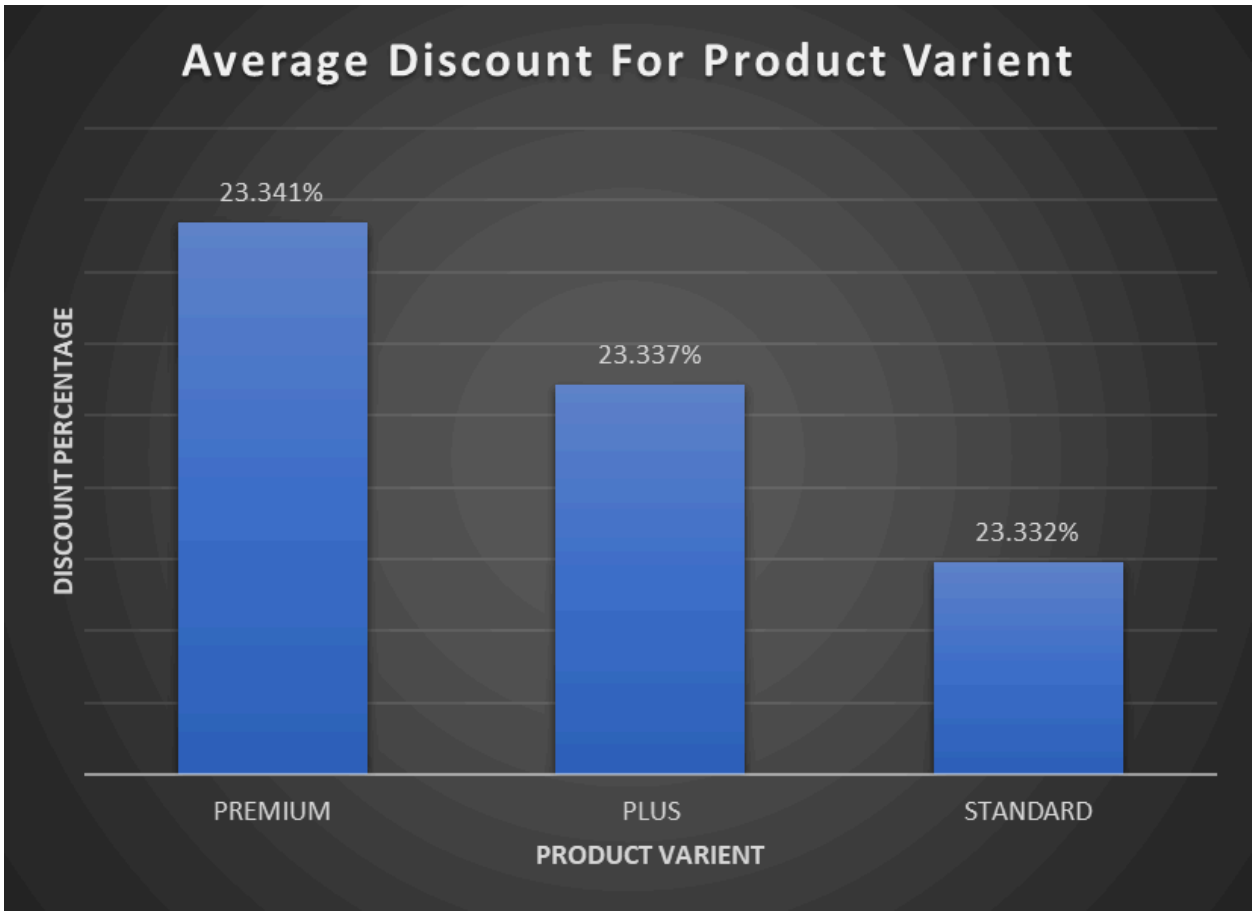
category	avg_discount_perc
USB Flash Drives	0.23372117
Mouse	0.23368441
Wi fi extender	0.23365025
Keyboard	0.23364332
External Solid State Drives	0.23362822
Internal HDD	0.23351948
Processors	0.23349529
Graphic Card	0.23345787
MotherBoard	0.23342986
Batteries	0.23332417
Personal Laptop	0.23331397
Business Laptop	0.23314023
Gaming Laptop	0.23277498
Personal Desktop	0.23094307

Discount and Standardised Variant.

```
-- discount and standardised_variant.
select t1.standardised_variant,avg(t2.pre_invoice_discount_pct) as avg_discount_perc
from sales_table t1 inner join fact_pre_invoice_deductions t2 on
t1.customer_code = t2.customer_code and t1.fiscal_year = t2.fiscal_year
group by t1.standardised_variant order by avg_discount_perc desc;
```

Output:

standardised_variant	avg_discount_perc
PREMIUM	0.23341358
PLUS	0.23336863
STANDARD	0.23331914



Discount Percentage, Product Variant ,Market.

Does the discount percentage vary for different variants of products across markets.

```
-- does discount percentage varies for different variant of products across markets.
-- simply discount,market,standardised_variant.
-- No doesnt vary
select t1.market,t1.standardised_variant,avg(t2.pre_invoice_discount_pct) as avg_discount_perc
from sales_table t1 inner join fact_pre_invoice_deductions t2 on
t1.customer_code = t2.customer_code and t1.fiscal_year = t2.fiscal_year
group by t1.standardised_variant, t1.market order by t1.market;
```

Insight No doesn't vary.

Discount and product segment, and product category.

```
-- discount and product segment, and product category.
select t1.segment,t1.category,avg(t2.pre_invoice_discount_pct) as avg_discount_perc
from sales_table t1 inner join fact_pre_invoice_deductions t2 on
t1.customer_code = t2.customer_code and t1.fiscal_year = t2.fiscal_year
group by t1.segment,t1.category order by t1.segment;
```

Output:

segment	category	avg_discount_perc
Accessories	Batteries	0.23332417
Accessories	Keyboard	0.23364332
Accessories	Mouse	0.23368441
Desktop	Business Laptop	0.23361989
Desktop	Personal Desktop	0.23094307
Networking	Wi fi extender	0.23365025
Notebook	Business Laptop	0.23305743
Notebook	Gaming Laptop	0.23277498
Notebook	Personal Laptop	0.23331397
Peripherals	Graphic Card	0.23345787
Peripherals	Internal HDD	0.23351948
Peripherals	MotherBoard	0.23342986
Peripherals	Processors	0.23349529
Storage	External Solid St...	0.23362822
Storage	USB Flash Drives	0.23372117

Insight: For different product categories across segments the avg discount percentage remains more or less the same.

Discount , Top 10 customers , Product Margin Category.

```
select t1.customer,t2.margin_category, avg(pre_invoice_discount_pct) as avg_percentage
from sales_table t1 inner join pcode_margin_cat t2
on t1.product_code = t2.product_code and t1.fiscal_year = t2.fiscal_year
inner join fact_pre_invoice_deductions t3 on
t1.customer_code = t3.customer_code and t1.fiscal_year = t3.fiscal_year
where t1.customer in (select distinct customer from top_10_customer_sales_table)
group by t1.customer,t2.margin_category
order by t1.customer;
```

Output:

customer	margin_category	avg_percentag
Amazon	high_margin_product	0.23746098
Amazon	low_maring_product	0.23820377
Amazon	medium_margin_product	0.23769137
Atliq e Store	high_margin_product	0.23850948
Atliq e Store	low_maring_product	0.23854701
Atliq e Store	medium_margin_product	0.23910850
Atliq Exclusive	high_margin_product	0.09712248
Atliq Exclusive	low_maring_product	0.09453823
Atliq Exclusive	medium_margin_product	0.09529062
Ebay	high_margin_product	0.22085288
Ebay	low_maring_product	0.22360156
Ebay	medium_margin_product	0.22214412
Electricalsocity	high_margin_product	0.23534615
Electricalsocity	low_maring_product	0.24288068
Electricalsocity	medium_margin_product	0.23998832
Flipkart	high_margin_product	0.26752529
Flipkart	low_maring_product	0.26826063
Flipkart	medium_margin_product	0.26902173
Leader	high_margin_product	0.24136911
Leader	low_maring_product	0.24153089
Leader	medium_margin_product	0.24050847
Neptune	high_margin_product	0.25459375
Neptune	low_maring_product	0.25052592
Neptune	medium_margin_product	0.25337574

Neptune	medium_margin_product	0.25337574

Sage	high_margin_product	0.23982940
Sage	low_maring_product	0.23711359
Sage	medium_margin_product	0.23803863
Synthetic	high_margin_product	0.24139633
Synthetic	low_maring_product	0.24902414
Synthetic	medium_margin_product	0.24698795

Discount, Region and Product Margin.

```
select t1.region,t2.margin_category, avg(pre_invoice_discount_pct) as avg_percentage
from sales_table t1 inner join pcode_margin_cat t2
on t1.product_code = t2.product_code and t1.fiscal_year = t2.fiscal_year
inner join fact_pre_invoice_deductions t3 on
t1.customer_code = t3.customer_code and t1.fiscal_year = t3.fiscal_year
group by t1.region,t2.margin_category
order by t1.region;
```

region	margin_category	avg_percentage
APAC	high_margin_product	0.22543529
APAC	low_maring_product	0.22489580
APAC	medium_margin_product	0.22513047
EU	high_margin_product	0.23724323
EU	low_maring_product	0.23721295
EU	medium_margin_product	0.23728168
LATAM	high_margin_product	0.25691577
LATAM	low_maring_product	0.25503249
LATAM	medium_margin_product	0.25387210
NA	high_margin_product	0.23573960
NA	low_maring_product	0.23732959
NA	medium_margin_product	0.23680758

Discount percentage doesn't really vary across margin categories for different regions.

Discount, Market , Product Margin Category.

```
-- discount , market , product margin category,
with temp_table as
(
select distinct market from sales_table group by market order by sum(sales_amount) desc limit 10
)
select t1.market,t2.margin_category, avg(pre_invoice_discount_pct) as avg_percentage
from sales_table t1 inner join pcode_margin_cat t2
on t1.product_code = t2.product_code and t1.fiscal_year = t2.fiscal_year
inner join fact_pre_invoice_deductions t3 on
t1.customer_code = t3.customer_code and t1.fiscal_year = t3.fiscal_year
where market in (select market from temp_table)
group by t1.market,t2.margin_category
order by t1.market;
```

Insight:
Discount Percentage doesn't vary with product margin categories across markets.

Customer, Segment and Discount Analysis

```
-- customer,segment, discount analysis.
with temp_table as
(
select distinct customer from sales_table group by customer order by sum(sales_amount) desc limit 10
)
select t1.customer,t1.segment,avg(pre_invoice_discount_pct) as discount_percentage from sales_table t1
inner join fact_pre_invoice_deductions t2
on t1.customer_code = t2.customer_code and t1.fiscal_year = t2.fiscal_year
where t1.customer in (select customer from temp_table)
group by t1.customer,t1.segment;
```

Output:
Discount % doesn't for different customers across different product segments.

Customer,Division and Discount Analysis

```
with temp_table as
(
select distinct customer from sales_table group by customer order by sum(sales_amount) desc limit 10
)
select t1.customer,t1.division,avg(pre_invoice_discount_pct) as discount_percentage from sales_table t1
inner join fact_pre_invoice_deductions t2
on t1.customer_code = t2.customer_code and t1.fiscal_year = t2.fiscal_year
where t1.customer in (select customer from temp_table)
group by t1.customer,t1.division;
```

Customer,Category ,Discount analysis

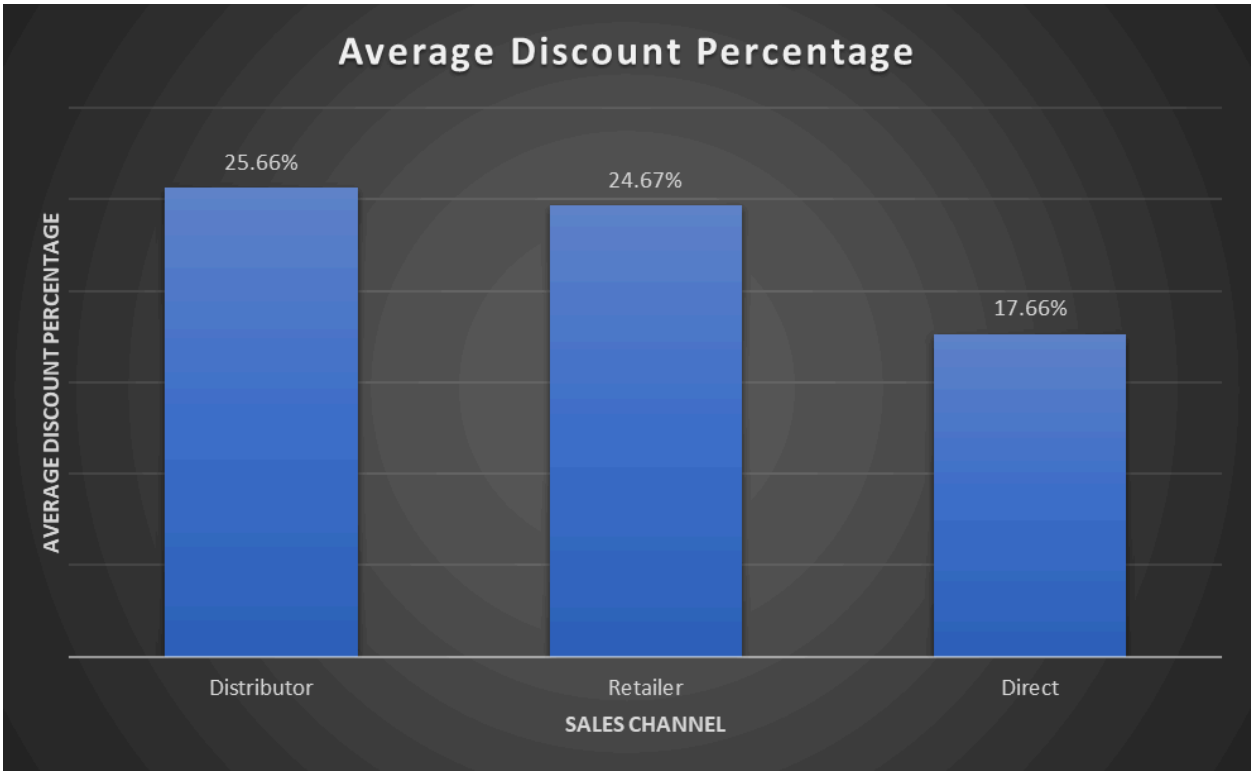
Nothing significant here.

Channel Discount Analysis

```
-- channel,discount,analysis

select t1.channel,avg(t2.pre_invoice_discount_pct) as avg_discount_percentage
from sales_table t1 inner join fact_pre_invoice_deductions t2
on t1.customer_code = t2.customer_code and
t1.fiscal_year = t2.fiscal_year
group by t1.channel;
```

Output:

channel	avg_discount_percentage
Distributor	0.25659734
Retailer	0.24674024
Direct	0.17655247

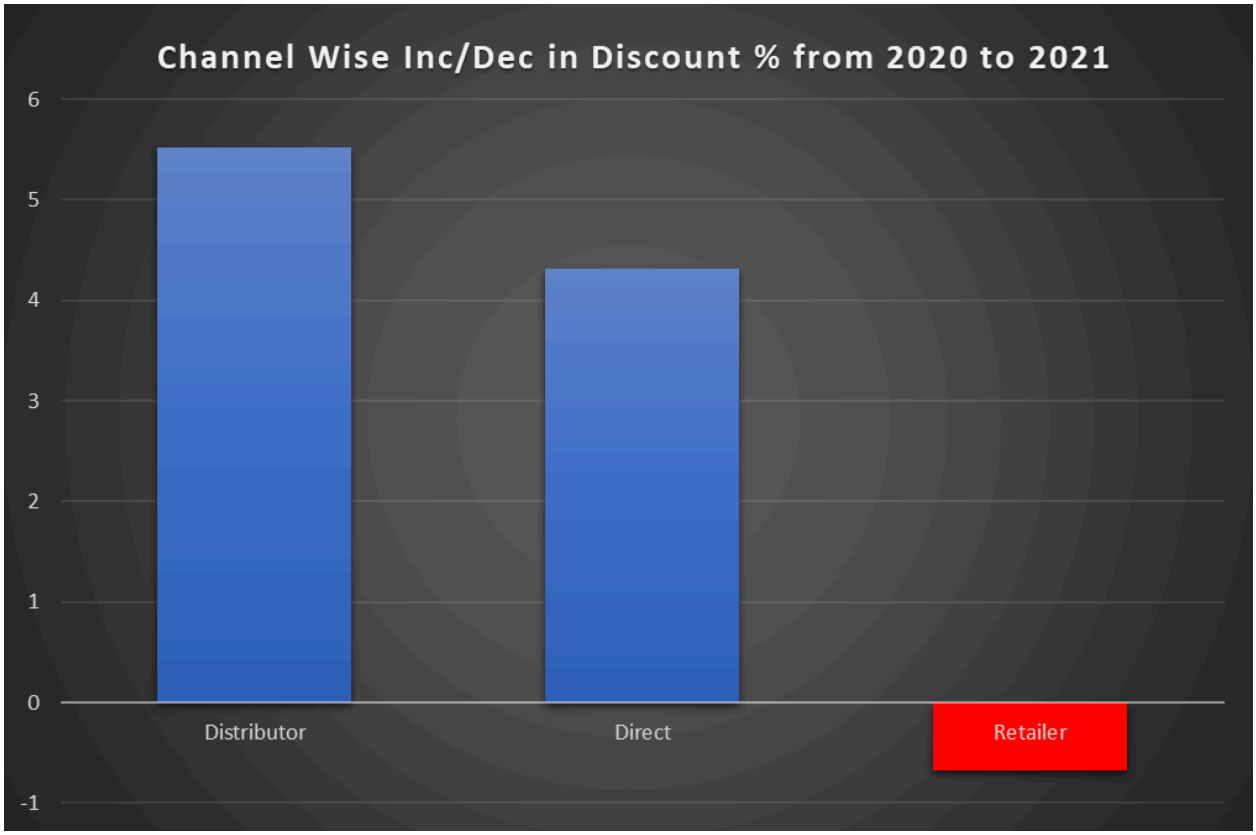


Channel wise discount percentage increase/decrease in 2020 and 2021.

```
-- channel and discount percentage increase/decrease in 2020 and 2021.
with discount_2020 as
(select t1.channel,avg(t2.pre_invoice_discount_pct) as avg_discount_percentage_2020
from sales_table t1 inner join fact_pre_invoice_deductions t2
on t1.customer_code = t2.customer_code and
t1.fiscal_year = t2.fiscal_year
where t1.fiscal_year = 2020
group by t1.channel),

discount_2021 as
(
select t1.channel,avg(t2.pre_invoice_discount_pct) as avg_discount_percentage_2021
from sales_table t1 inner join fact_pre_invoice_deductions t2
on t1.customer_code = t2.customer_code and
t1.fiscal_year = t2.fiscal_year
where t1.fiscal_year = 2021
group by t1.channel
)

select t1.channel,t1.avg_discount_percentage_2020, t2.avg_discount_percentage_2021,
 round(((t2.avg_discount_percentage_2021 - t1.avg_discount_percentage_2020)/(t1.avg_discount_percentage_2020)) * 100,2) as discount_pct_inc_dec_perc
from discount_2020 t1 inner join discount_2021 t2 on t1.channel = t2.channel
order by discount_pct_inc_dec_perc desc;
```

Output:

channel	avg_discount_percentage_2020	avg_discount_percentage_2021	discount_pct_inc_dec_perc
Distributor	0.24857948	0.26230000	5.52
Direct	0.17196792	0.17940164	4.32
Retailer	0.24777896	0.24612903	-0.67

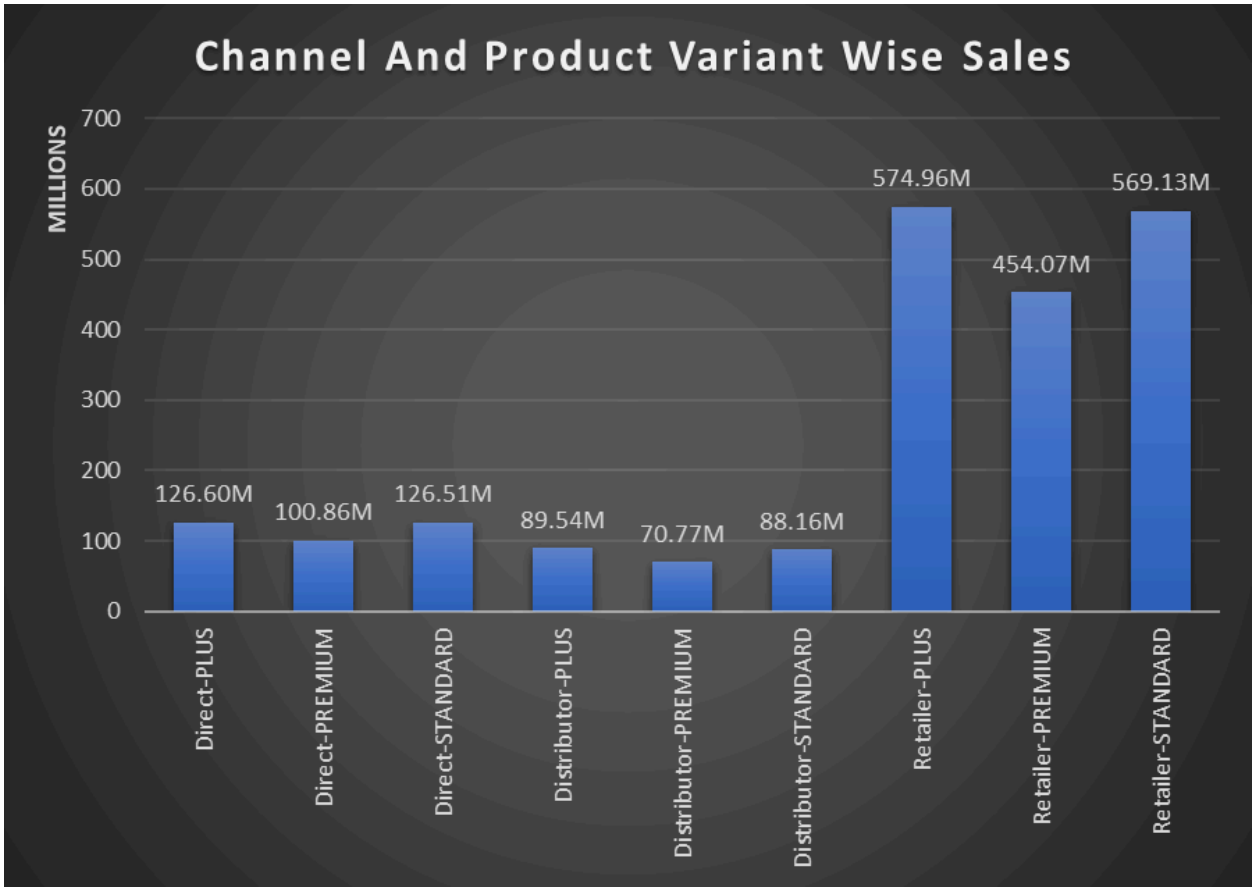


Channel and Variant Analysis.

```
-- channel variant analysis sales amt analysis.
select channel,standardised_variant,sum(sales_amount) as sales_amoint
from sales_table group by channel,standardised_variant
order by channel;
```

Output:

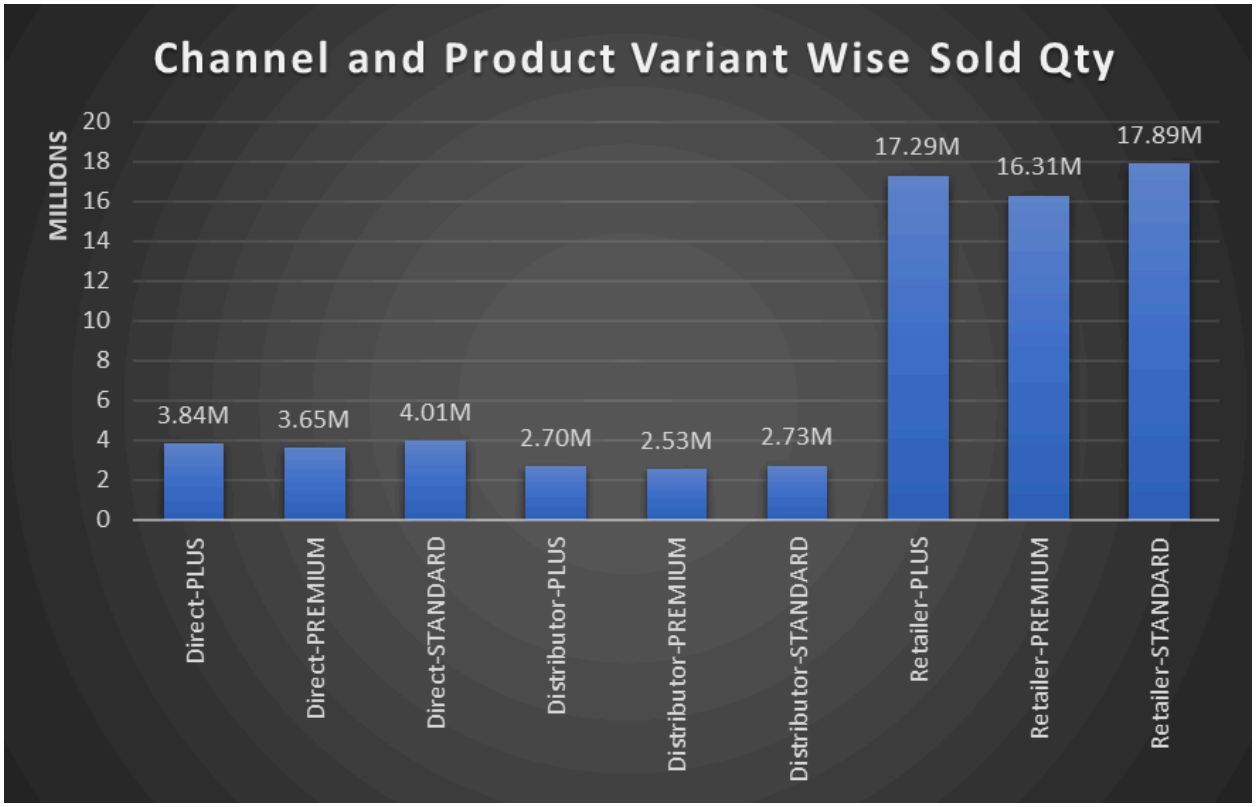
channel	standardised_variant	sales_amoint
Direct	PLUS	126597167.4928
Direct	PREMIUM	100856844.9805
Direct	STANDARD	126509839.7944
Distributor	PLUS	89542554.3750
Distributor	PREMIUM	70766214.1304
Distributor	STANDARD	88156465.2596
Retailer	PLUS	574957130.1471
Retailer	PREMIUM	454067767.3352
Retailer	STANDARD	569131061.9226



Channel And Product Variant Analysis By sold Quantity.

```
-- channel variant analysis sold qty analysis.
select channel,standardised_variant,sum(sold_quantity) as sold_quantity
from sales_table group by channel,standardised_variant
order by channel;
```

channel	standardised_variant	sold_quantity
Direct	PLUS	3839533
Direct	PREMIUM	3645222
Direct	STANDARD	4006068
Distributor	PLUS	2702941
Distributor	PREMIUM	2529903
Distributor	STANDARD	2728183
Retailer	PLUS	17292467
Retailer	PREMIUM	16307097
Retailer	STANDARD	17886057

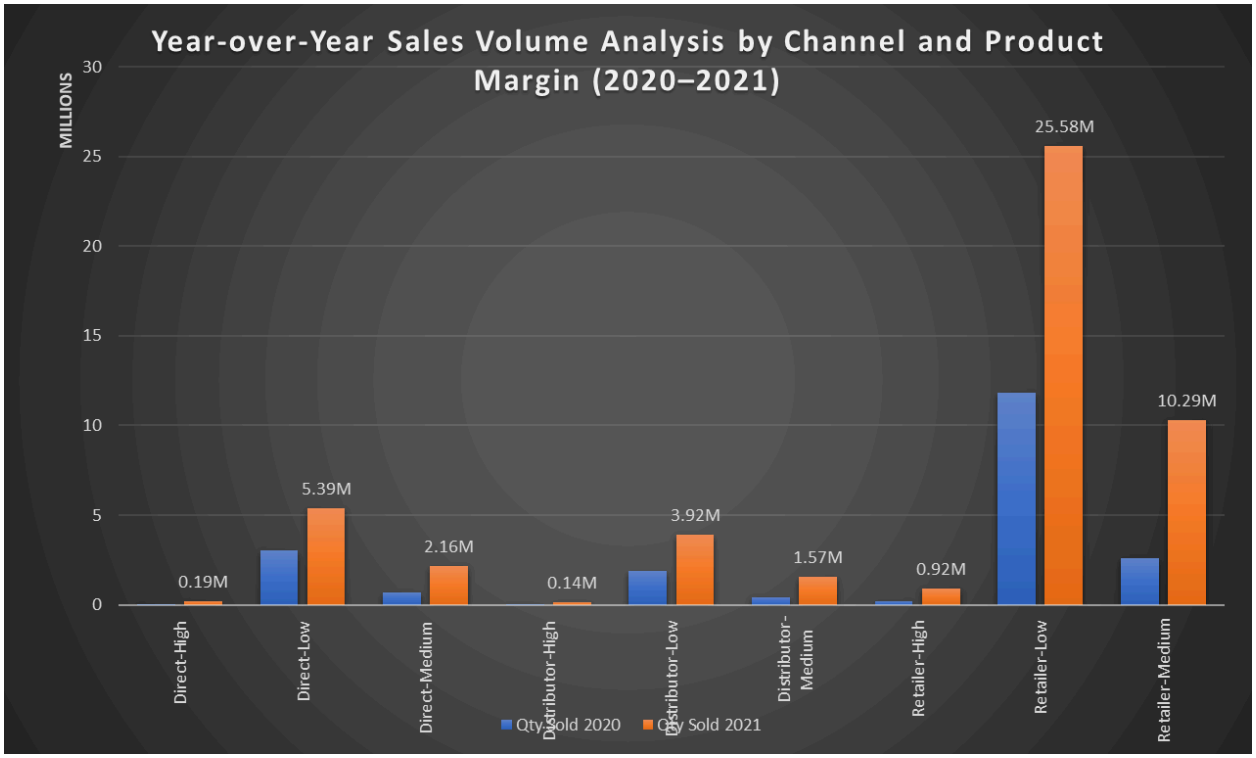


Channel , Product Margin Category And Sold Quantity.

```sql
-- channel, product_ margin category, yoy sold_qty.
with channel_sold_qty_2020 as
(select t1.channel,t2.margin_category,sum(sold_quantity) as sold_qty_2020
from sales_table t1 inner join pcode_margin_cat t2
on t1.product_code = t2.product_code and t1.fiscal_year = t2.fiscal_year
where t1.fiscal_year = 2020
group by t1.channel,t2.margin_category
order by t1.channel),
channel_sold_qty_2021 as
(select t1.channel,t2.margin_category,sum(sold_quantity) as sold_qty_2021
from sales_table t1 inner join pcode_margin_cat t2
on t1.product_code = t2.product_code and t1.fiscal_year = t2.fiscal_year
where t1.fiscal_year = 2021
group by t1.channel,t2.margin_category
order by t1.channel)
select t1.channel,t1.margin_category, t1.sold_qty_2020 , t2.sold_qty_2021,
t2.sold_qty_2021 - t1.sold_qty_2020 as qty_inc_dec,
round(((t2.sold_qty_2021 - t1.sold_qty_2020)/(t1.sold_qty_2020)) * 100,2) as perc_inc_or_dec
from channel_sold_qty_2020 t1 inner join channel_sold_qty_2021 t2
on t1.channel = t2.channel and t1.margin_category = t2.margin_category;
```

Output:

channel	margin_category	sold_qty_2020	sold_qty_2021	qty_inc_dec	perc_inc_or_dec
Direct	high_margin_product	55297	194095	138798	251.00
Direct	low_maring_product	3023864	5386922	2363058	78.15
Direct	medium_margin_product	668211	2162434	1494223	223.62
Distributor	high_margin_product	34858	141976	107118	307.30
Distributor	low_maring_product	1868159	3921585	2053426	109.92
Distributor	medium_margin_product	425624	1568825	1143201	268.59
Retailer	high_margin_product	218310	918492	700182	320.73
Retailer	low_maring_product	11850771	25580964	13730193	115.86
Retailer	medium_margin_product	2627795	10289289	7661494	291.56



Channel,Product Margin Category and Sales Amount

```
-- channel, product_ margin category sales analysis.
select t1.channel,t2.margin_category,sum(sales_amount) as sales_amount from sales_table t1 inner join
pcode_margin_cat t2 on t1.product_code = t2.product_code
and t1.fiscal_year = t2.fiscal_year
group by t1.channel,t2.margin_category
order by t1.channel;
```

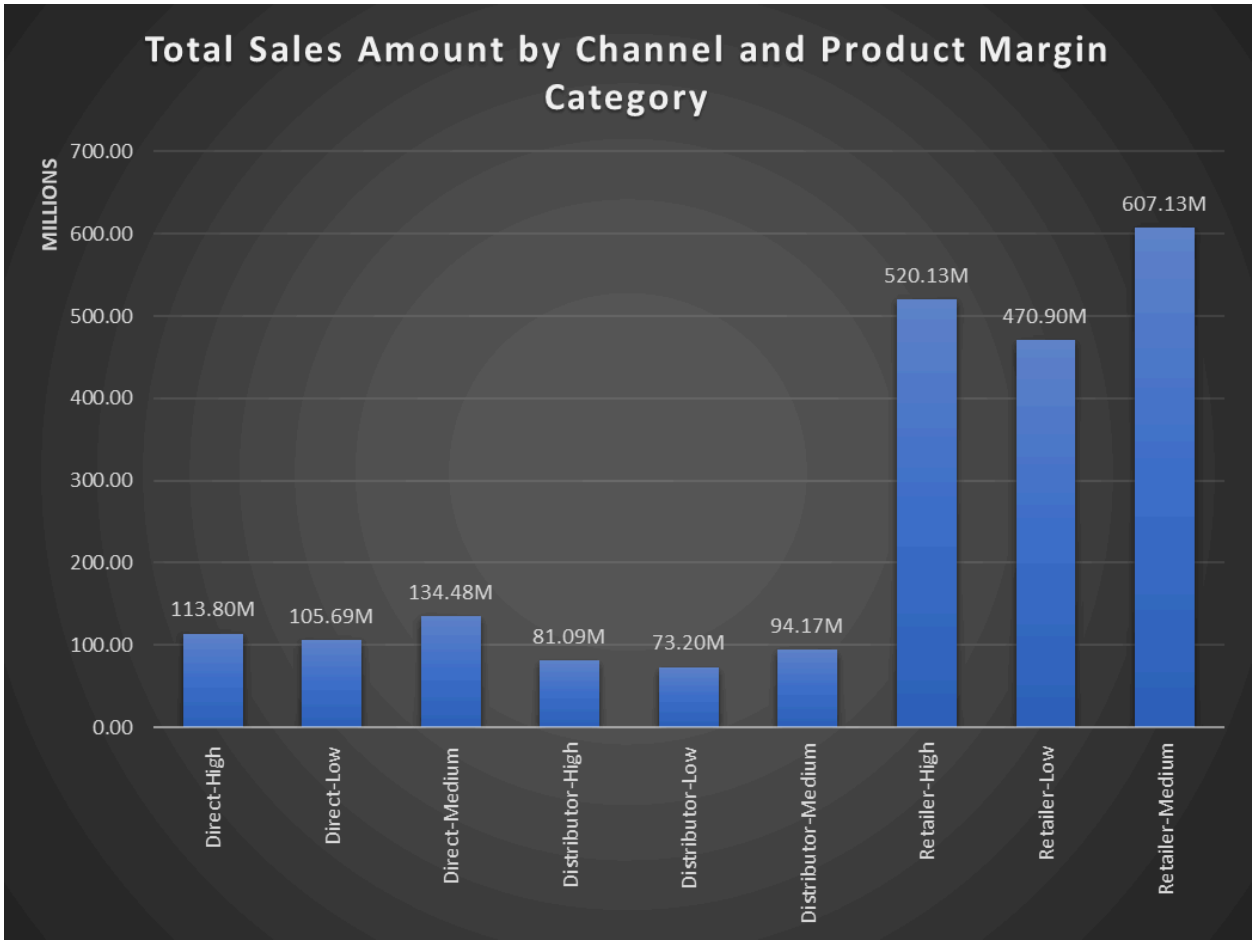
Output:

channel	margin_category	sales_amount
Direct	high_margin_product	113800252.8319
Direct	low_maring_product	105687295.6405
Direct	medium_margin_product	134476303.7953
Distributor	high_margin_product	81090089.4261
Distributor	low_maring_product	73202705.1507
Distributor	medium_margin_product	94172439.1882
Retailer	high_margin_product	520129200.3289
Retailer	low_maring_product	470897893.3317
Retailer	medium_margin_product	607128865.7443

```sql
-- channel, product_ margin category sales analysis.
select t1.channel,t2.margin_category,sum(sales_amount) as sales_amount from sales_table t1 inner join
pcode_margin_cat t2 on t1.product_code = t2.product_code
and t1.fiscal_year = t2.fiscal_year
group by t1.channel,t2.margin_category
order by t1.channel;
```

Output:

channel	margin_category	sales_amount
Direct	high_margin_product	113800252.8319
Direct	low_maring_product	105687295.6405
Direct	medium_margin_product	134476303.7953
Distributor	high_margin_product	81090089.4261
Distributor	low_maring_product	73202705.1507
Distributor	medium_margin_product	94172439.1882
Retailer	high_margin_product	520129200.3289
Retailer	low_maring_product	470897893.3317
Retailer	medium_margin_product	607128865.7443


Total Sales Amount by Channel and Product Margin Category
MILLIONS
700.00
600.00
500.00
400.00
300.00
200.00
100.00
0.00
113.80M
105.69M
134.48M
81.09M
73.20M
94.17M
520.13M
470.90M
607.13M
Direct-High
Direct-Low
Direct-Medium
Distributor-High
Distributor-Low
Distributor-Medium
Retailer-High
Retailer-Low
Retailer-Medium

Insights:
Latam- Latin America receives the most discount yet the sales have not grown significantly over there.

Discount is mostly the same for all kinds of product categories or product margin wise categories.

Discount is highest currently for the products of low margin category. Need to incentivize customers more for the higher and medium margin category products by introducing schemes or increasing discount percentage for bulk buyers.

This big discount for the low margin category might also be as they are getting bought in very huge quantities.

Average discount percentage for products getting sold from the direct channel is lower as compared to other sales channels.

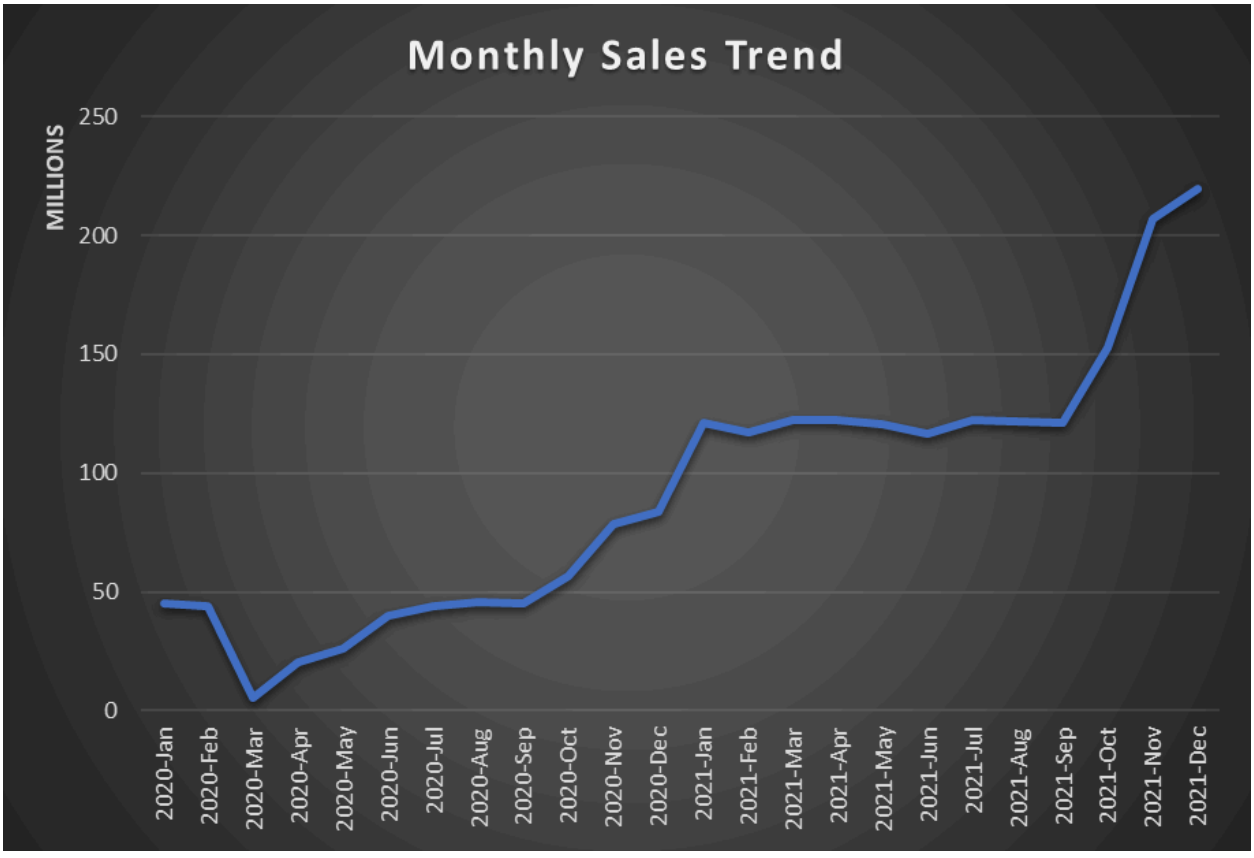
There is a decrease in the average discount percentage for retail sales channel From 2020 to 2021 of 0.67%.

Seasonal Analysis.

Monthly Sales Trend

```sql
select
    fiscal_year,
    month(date) as month_num,
    date_format(date, '%b') as month_name,
    sum(sales_amount) as total_sales
from sales_table
group by fiscal_year, month_num, month_name
order by fiscal_year, month_num;
```

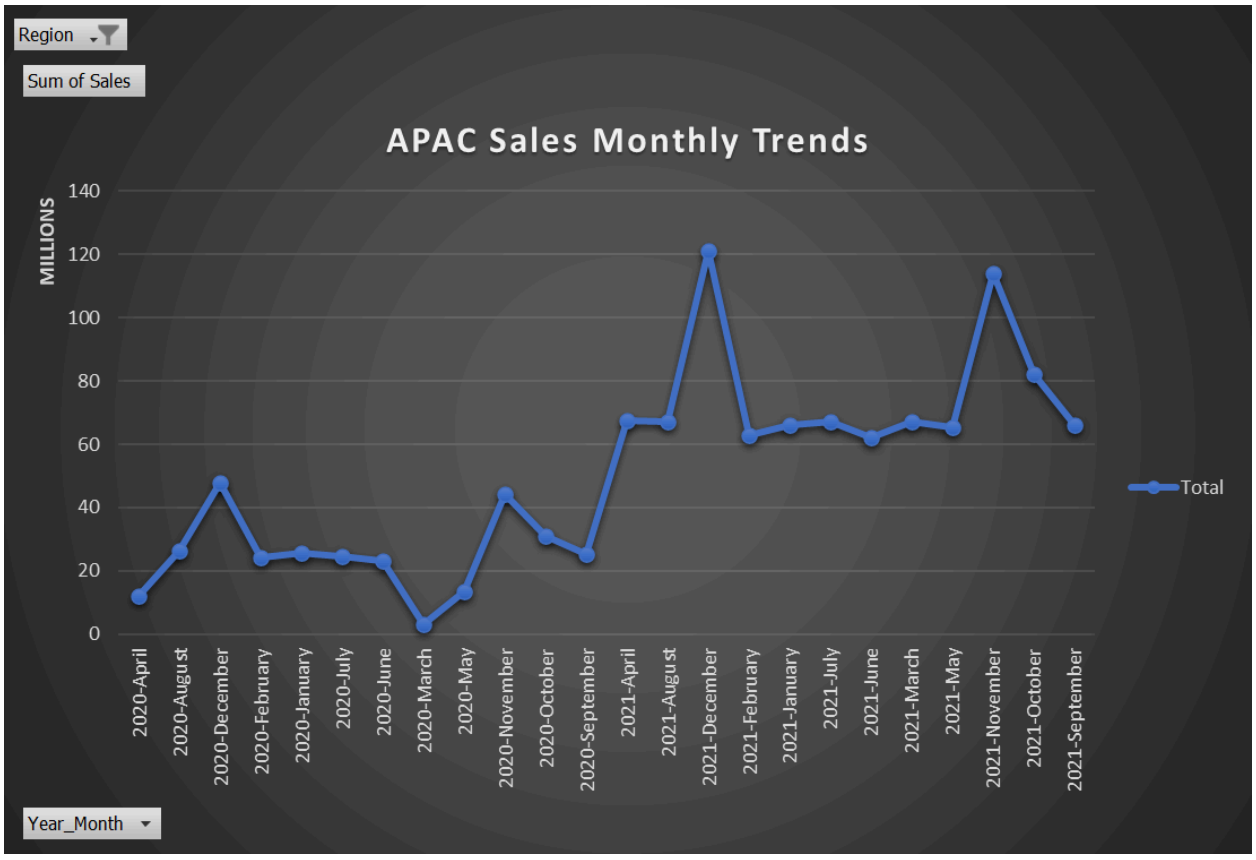
Output:

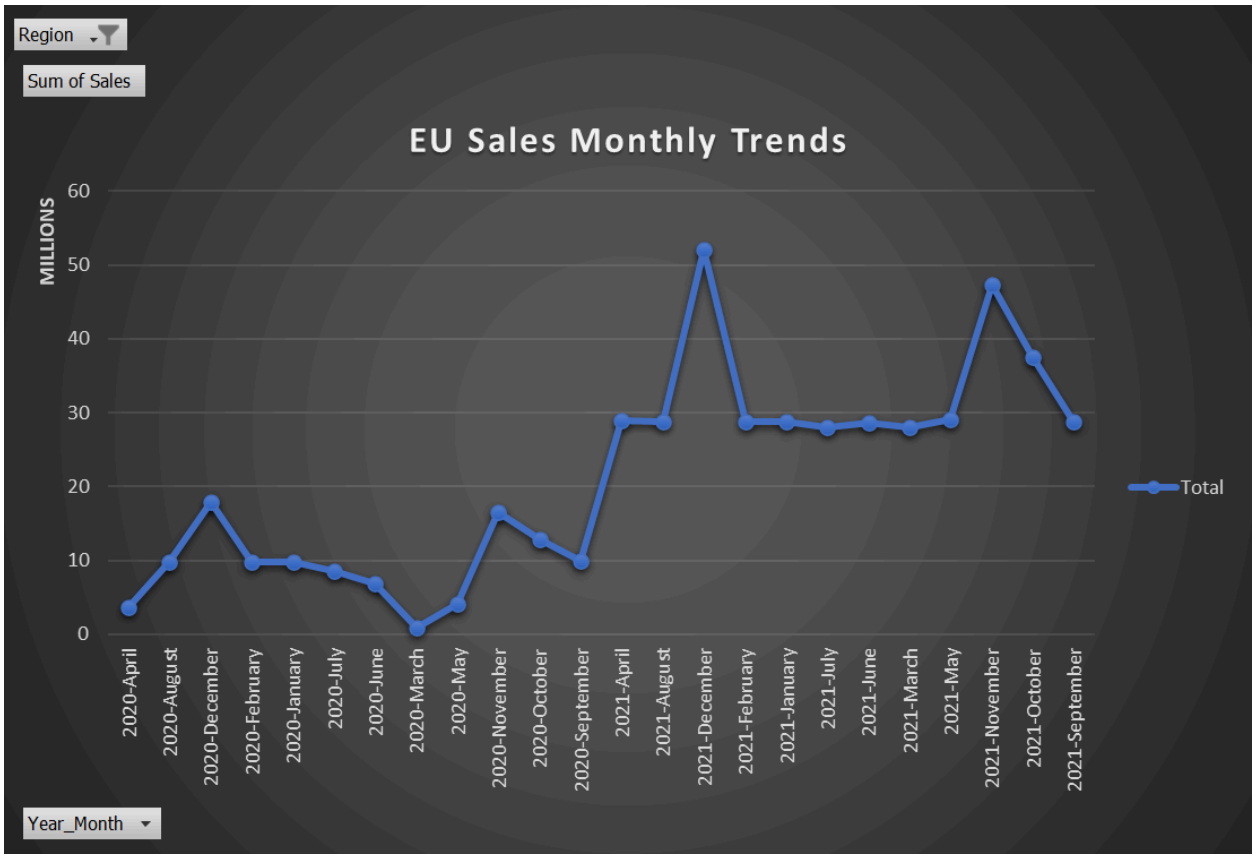


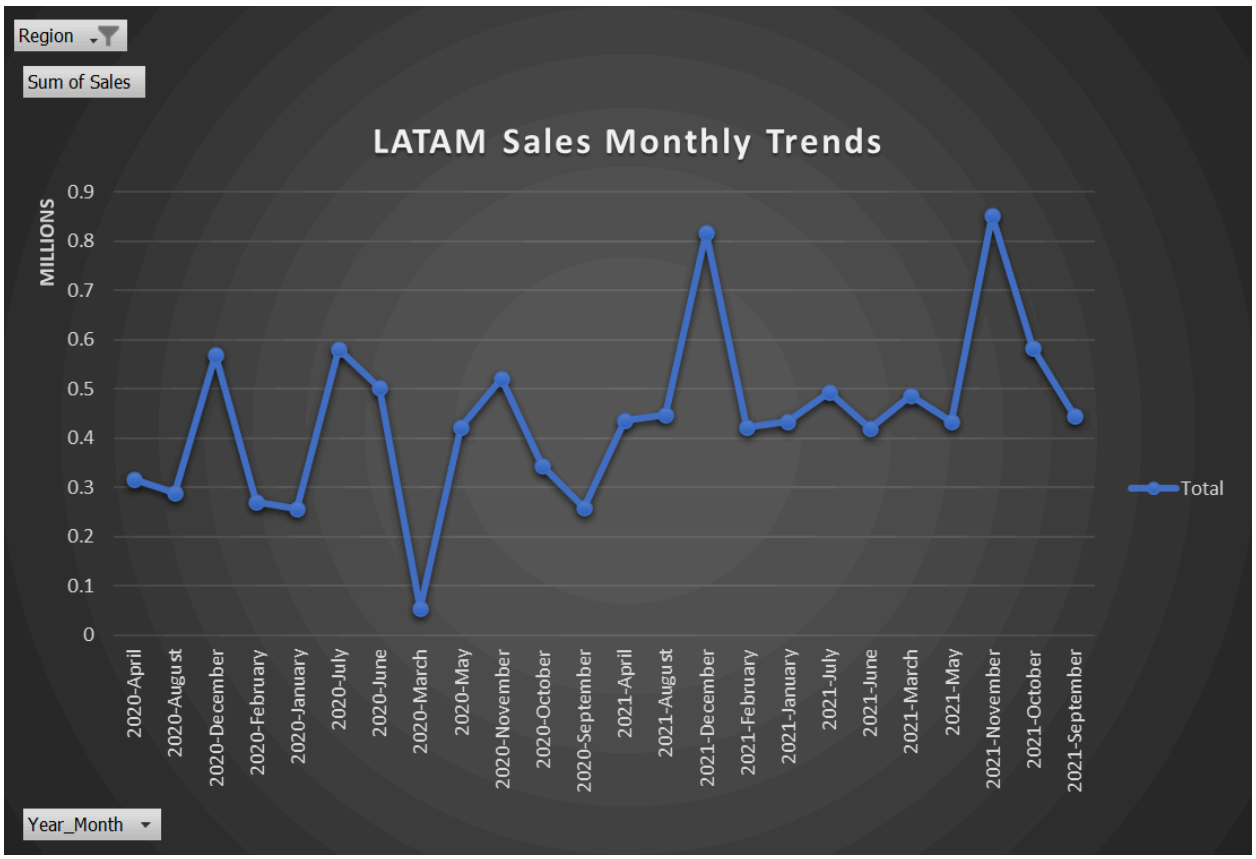
Regionwise Monthly Sales Trends

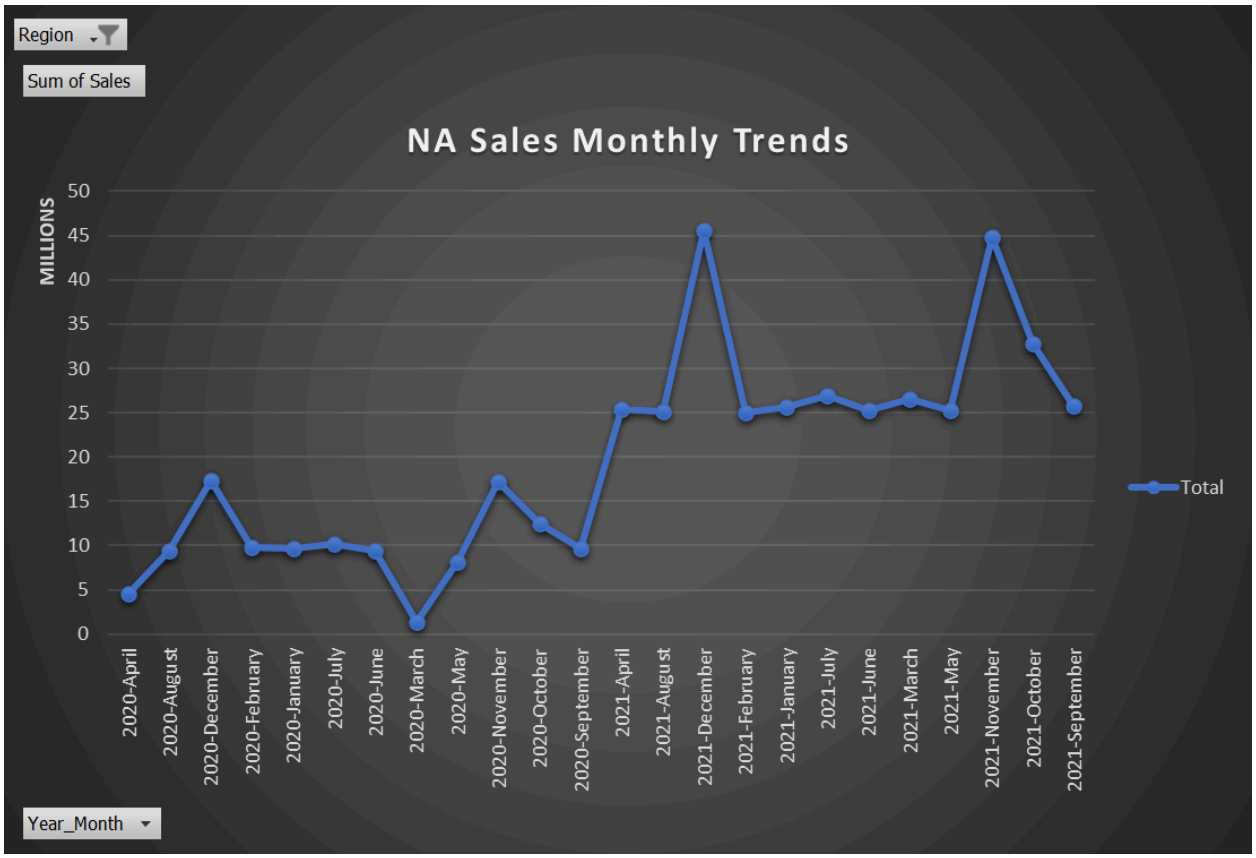
```sql
-- seasonal sales by region monthly.
select region,
concat(fiscal_year,"-",monthname(`date`)) as `year_month`,
sum(sales_amount) as total_sales
from sales_table
group by region, `year_month`
order by region, `year_month`;
```

Output:



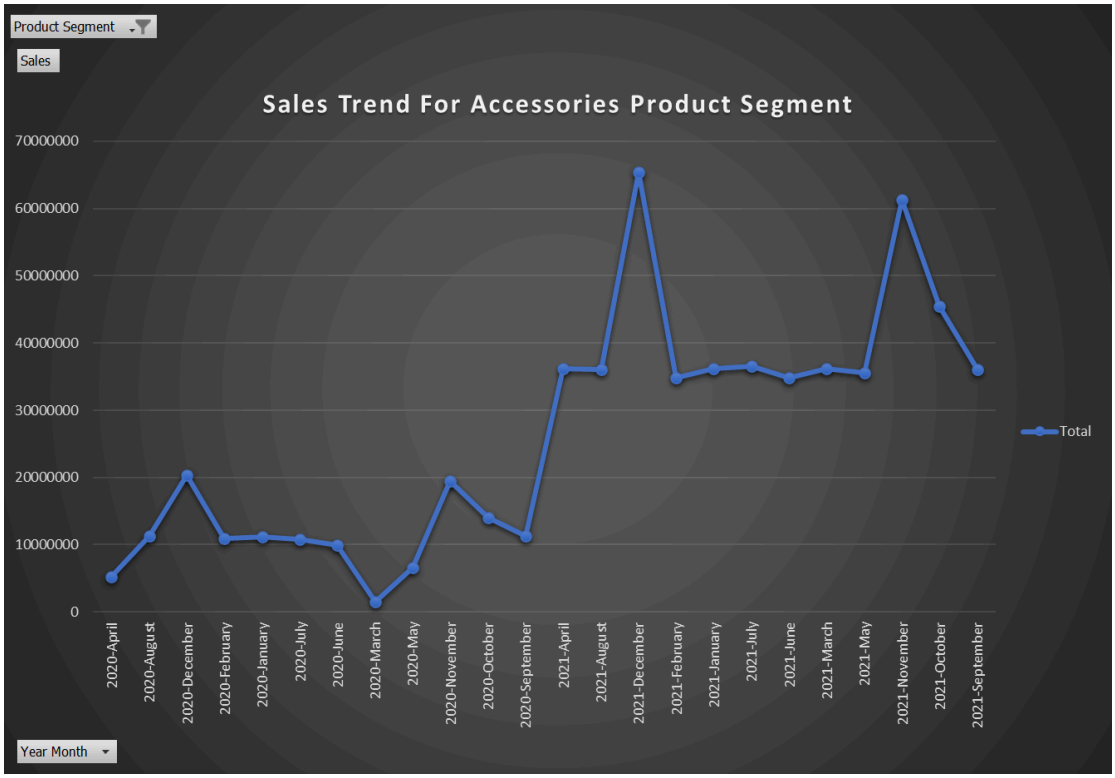

Region
Sum of Sales
EU Sales Monthly Trends
MILLIONS
60
50
40
30
20
10
0
2020-April
2020-August
2020-December
2020-February
2020-January
2020-July
2020-June
2020-March
2020-May
2020-November
2020-October
2020-September
2021-April
2021-August
2021-December
2021-February
2021-January
2021-July
2021-June
2021-March
2021-May
2021-November
2021-October
2021-September
Total
Year_Month

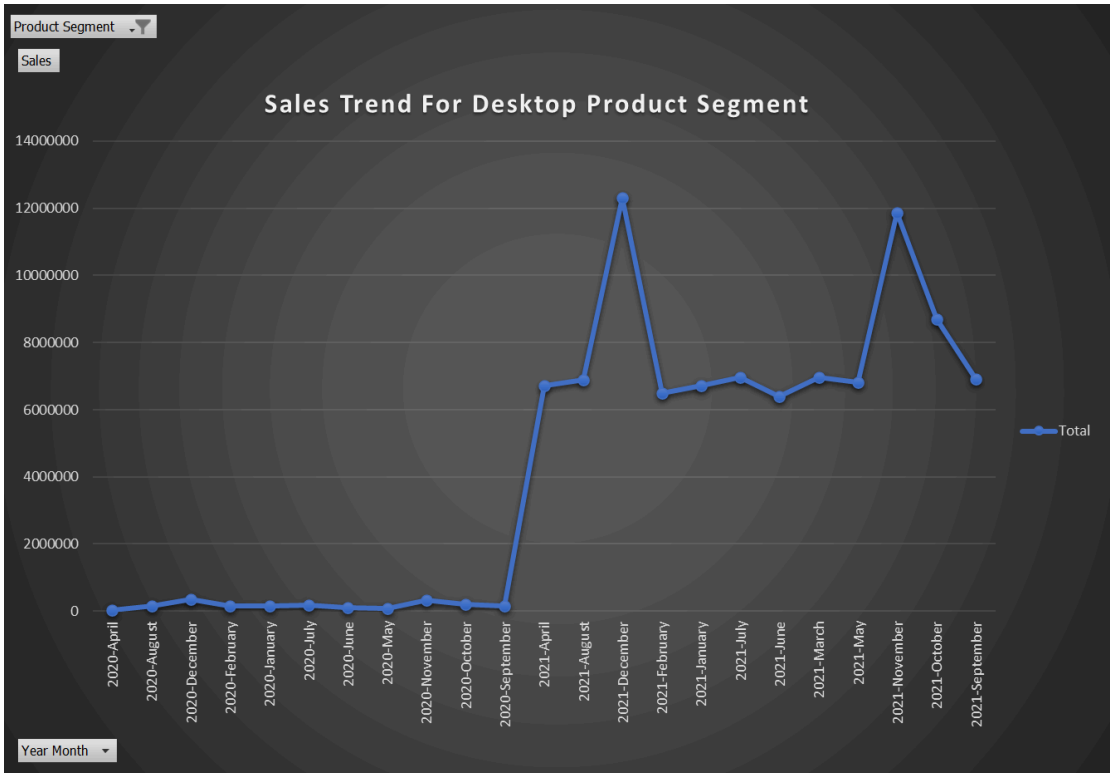

Region
Sum of Sales
LATAM Sales Monthly Trends
MILLIONS
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
2020-April
2020-August
2020-December
2020-February
2020-January
2020-July
2020-June
2020-March
2020-May
2020-November
2020-October
2020-September
2021-April
2021-August
2021-December
2021-February
2021-January
2021-July
2021-June
2021-March
2021-May
2021-November
2021-October
2021-September
Total
Year_Month

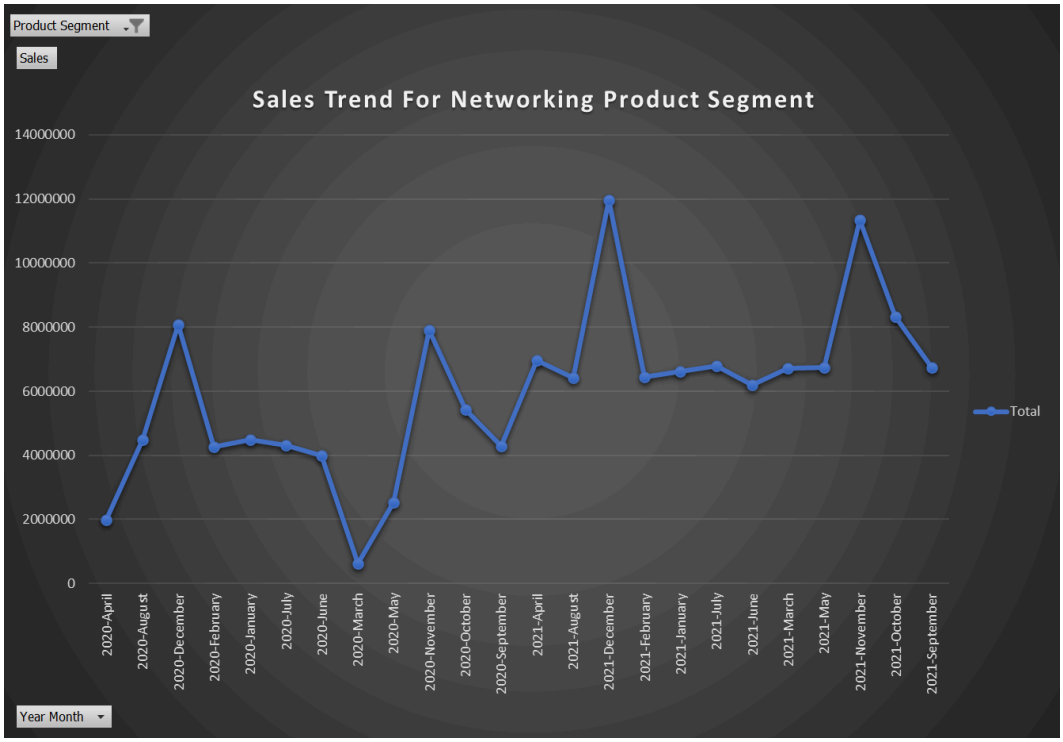


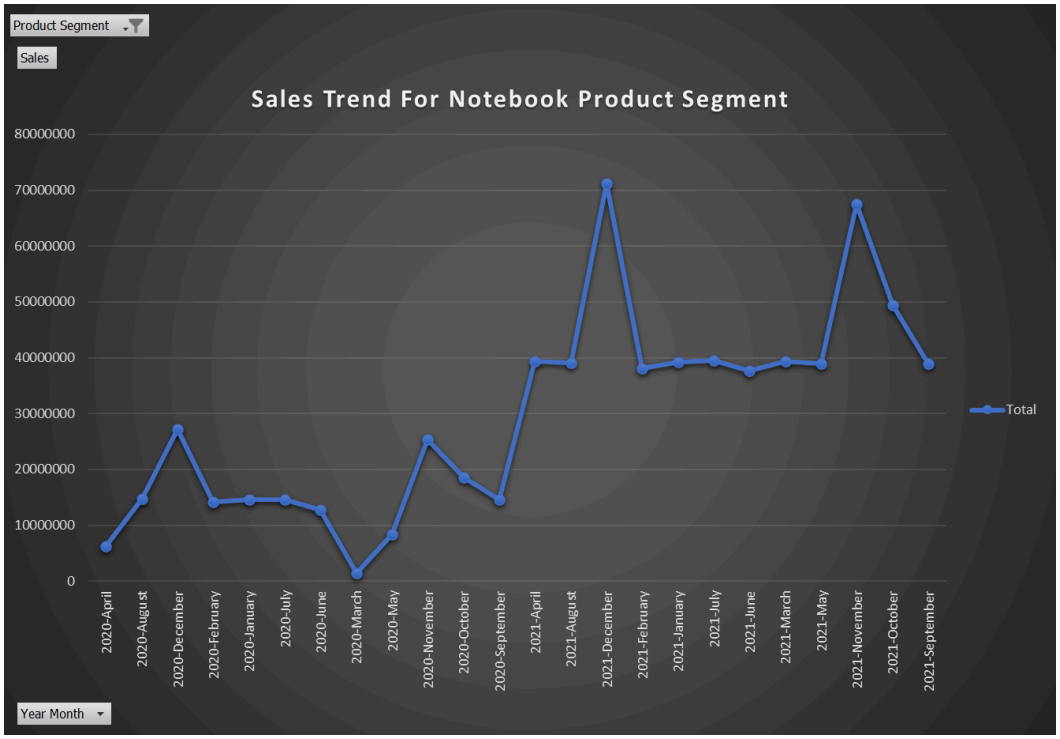
Seasonal Sales By Product Segment

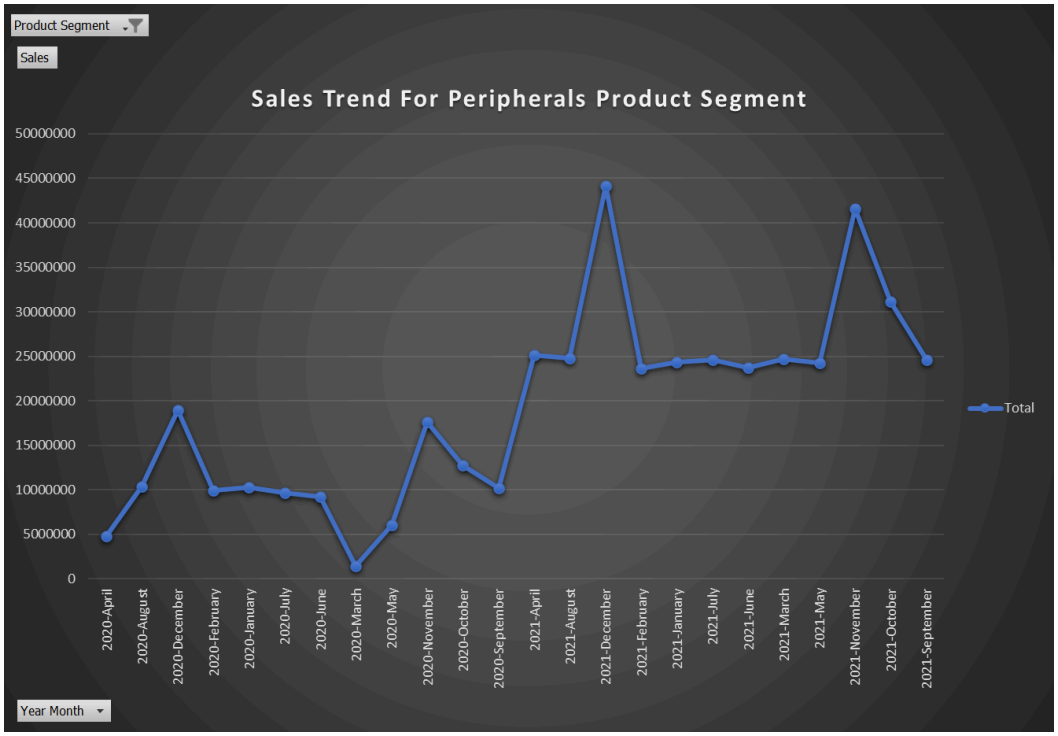
```sql
-- Seasonal Sales by Product Segment
select
    segment,
    concat(fiscal_year,"-",monthname(date)) AS `year_month`,
    SUM(sales_amount) AS total_sales
FROM sales_table
GROUP BY segment, `year_month`
ORDER BY segment, `year_month`;
```

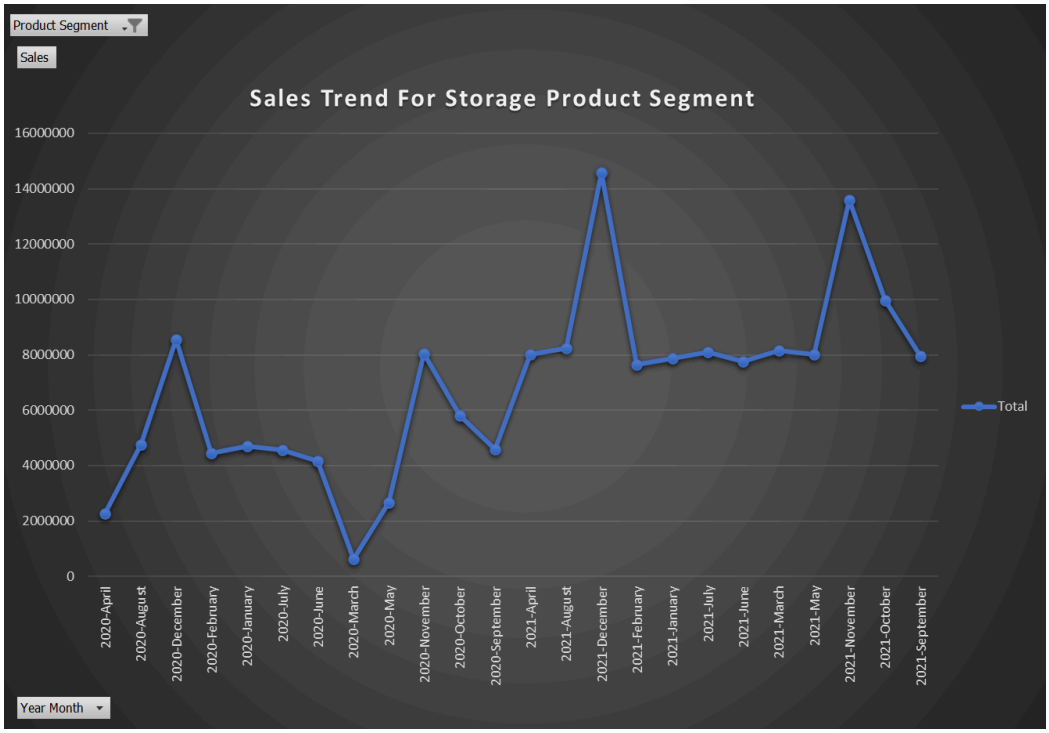

Product Segment
Sales
Sales Trend For Accessories Product Segment
70000000
60000000
50000000
40000000
30000000
20000000
10000000
0
2020-April
2020-August
2020-December
2020-February
2020-January
2020-July
2020-June
2020-March
2020-May
2020-November
2020-October
2020-September
2021-April
2021-August
2021-December
2021-February
2021-January
2021-July
2021-June
2021-March
2021-May
2021-November
2021-October
2021-September
Total
Year Month


Product Segment
Sales
Sales Trend For Desktop Product Segment
14000000
12000000
10000000
8000000
6000000
4000000
2000000
0
2020-April
2020-August
2020-December
2020-February
2020-January
2020-July
2020-June
2020-May
2020-November
2020-October
2020-September
2021-April
2021-August
2021-December
2021-February
2021-January
2021-July
2021-June
2021-March
2021-May
2021-November
2021-October
2021-September
Total
Year Month


Product Segment
Sales
Sales Trend For Networking Product Segment
14000000
12000000
10000000
8000000
6000000
4000000
2000000
0
2020-April
2020-August
2020-December
2020-February
2020-January
2020-July
2020-June
2020-March
2020-May
2020-November
2020-October
2020-September
2021-April
2021-August
2021-December
2021-February
2021-January
2021-July
2021-June
2021-March
2021-May
2021-November
2021-October
2021-September
Total
Year Month


Product Segment
Sales
Sales Trend For Notebook Product Segment
80000000
70000000
60000000
50000000
40000000
30000000
20000000
10000000
0
2020-April
2020-August
2020-December
2020-February
2020-January
2020-July
2020-June
2020-March
2020-May
2020-November
2020-October
2020-September
2021-April
2021-August
2021-December
2021-February
2021-January
2021-July
2021-June
2021-March
2021-May
2021-November
2021-October
2021-September
Total
Year Month


Product Segment
Sales
Sales Trend For Peripherals Product Segment
50000000
45000000
40000000
35000000
30000000
25000000
20000000
15000000
10000000
5000000
0
2020-April
2020-August
2020-December
2020-February
2020-January
2020-July
2020-June
2020-March
2020-May
2020-November
2020-October
2020-September
2021-April
2021-August
2021-December
2021-February
2021-January
2021-July
2021-June
2021-March
2021-May
2021-November
2021-October
2021-September
Total
Year Month


Product Segment
Sales
Sales Trend For Storage Product Segment
16000000
14000000
12000000
10000000
8000000
6000000
4000000
2000000
0
2020-April
2020-August
2020-December
2020-February
2020-January
2020-July
2020-June
2020-March
2020-May
2020-November
2020-October
2020-September
2021-April
2021-August
2021-December
2021-February
2021-January
2021-July
2021-June
2021-March
2021-May
2021-November
2021-October
2021-September
Total
Year Month

Sales Growth YoY by Quarter by Sales Amount.

```sql
-- season growth yoy by quarter by sales amount
with quarterly_sales_2020 as
(select quarter(date) as quarter_2020 ,sum(sales_amount) as quarterly_sales_2020 from sales_table
where fiscal_year = 2020
group by quarter(date)
),
quarterly_sales_2021 as
(select quarter(date) as quarter_2021 ,sum(sales_amount) as quarterly_sales_2021 from sales_table
where fiscal_year = 2021
group by quarter(date)
)
select t1.quarter_2020,t1.quarterly_sales_2020,t2.quarterly_sales_2021,
t2.quarterly_sales_2021 - t1.quarterly_sales_2020 as inc_or_dec,
((t2.quarterly_sales_2021 - t1.quarterly_sales_2020)/(t1.quarterly_sales_2020)) * 100 as inc_or_dec_percentage
from quarterly_sales_2020 t1 inner join quarterly_sales_2021 t2
on t1.quarter_2020 = t2.quarter_2021
order by t1.quarter_2020;
```

Output:

quarter_2020	quarterly_sold_qty_2020	quarterly_sold_qty_2021	inc_or_dec	inc_or_dec_percentage
1	3704398	10861386	7156988	193.2025
2	3395899	10831277	7435378	218.9517
3	5246770	11024794	5778024	110.1254
4	8425822	17447125	9021303	107.0673

Quarter Wise Analysis the sales this year quarter than last year has increased.

Products Sold YoY by Quarter by Qty Sold. as per fiscal year.

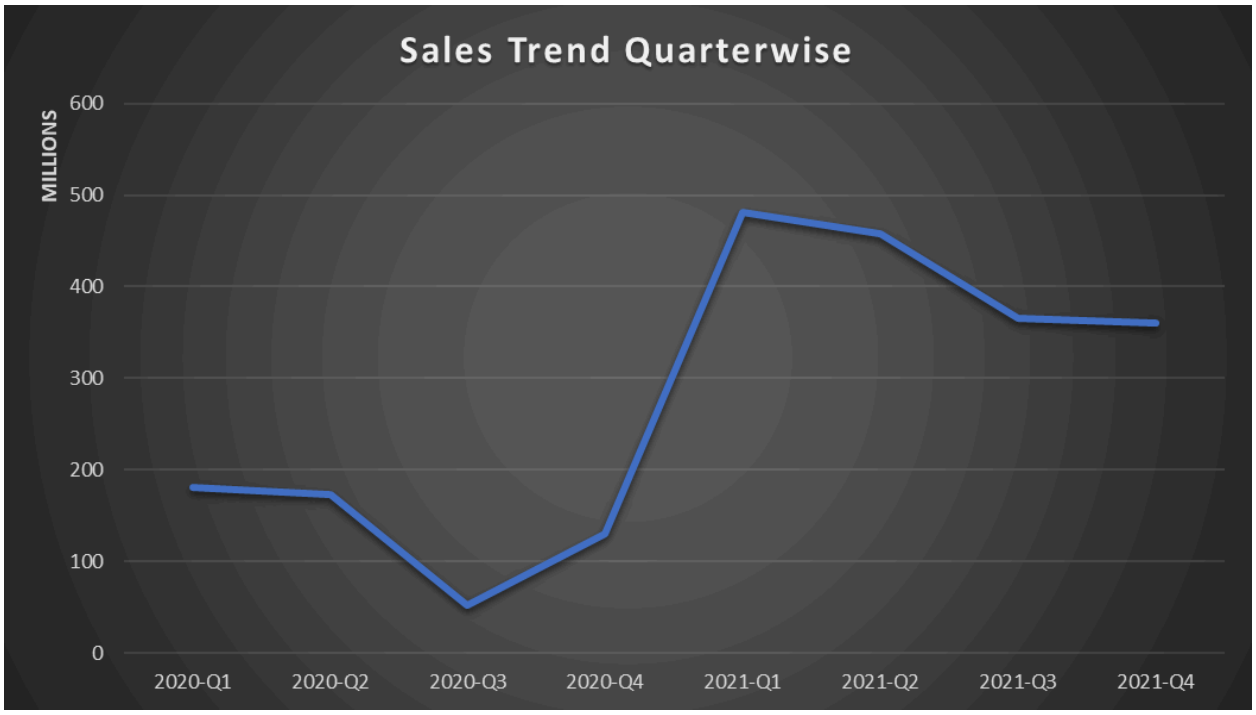
```sql
with quarterly_sold_qty_2020 as
(select quarter(date) as quarter_2020 ,sum(sold_quantity) as quarterly_sold_qty_2020 from sales_table
where fiscal_year = 2020
group by quarter(date)
),
quarterly_sold_qty_2021 as
(select quarter(date) as quarter_2021 ,sum(sold_quantity) as quarterly_sold_qty_2021 from sales_table
where fiscal_year = 2021
group by quarter(date)
)
select t1.quarter_2020,t1.quarterly_sold_qty_2020,t2.quarterly_sold_qty_2021,
t2.quarterly_sold_qty_2021 - t1.quarterly_sold_qty_2020 as inc_or_dec,
((t2.quarterly_sold_qty_2021 - t1.quarterly_sold_qty_2020)/(t1.quarterly_sold_qty_2020)) * 100 as inc_or_dec_percentage
from quarterly_sold_qty_2020 t1 inner join quarterly_sold_qty_2021 t2
on t1.quarter_2020 = t2.quarter_2021
order by t1.quarter_2020;
-- for quarterly the sold quantity has increased quarter over quarter.
```

Output:

quarter_2020	quarterly_sold_qty_2020	quarterly_sold_qty_2021	inc_or_dec	inc_or_dec_percentage
1	3704398	10861386	7156988	193.2025
2	3395899	10831277	7435378	218.9517
3	5246770	11024794	5778024	110.1254
4	8425822	17447125	9021303	107.0673

Sold quantity for 2021 fiscal year has increased for every quarter as compared to the fiscal year 2020.

Quarter Wise Analysis of Sales.



QuarterWise sales analysis seems to be decreasing.
It increased from around 2020 - Q3 till 2021 - Q1 and has been decreasing till 2021-Q4.

Additional Analysis

Regional Profitability By Sales growth

```
-- abosoulte growth in net_sales from the 2020 to 2021 regionwise.
with net_sales_t_2020 as
(select region, sum(net_sales) as net_sales_2020 from regional_profitability
where fiscal_year = 2020
group by region ),
net_sales_t_2021 as
(
select region, sum(net_sales) as net_sales_2021 from regional_profitability
where fiscal_year = 2021
group by region)
select *,
net_sales_2021 - net_sales_2020 as sales_growth,
((net_sales_2021 - net_sales_2020)/(net_sales_2020)) * 100 as sales_growth_pct
from net_sales_t_2020 t1 inner join net_sales_t_2021 t2
on t1.region = t2.region;
```

region	net_sales_2020	region	net_sales_2021	sales_growth	sales_growth_pct
APAC	232798444.50253549	APAC	693781163.92750890	460982719.42497341	198.017955150019
EU	84426507.83255441	EU	301264613.06740311	216838105.23484870	256.836520663522
NA	90748534.62394322	NA	272435164.51616139	181686629.89221817	200.208885625666
LATAM	3277130.28102746	LATAM	4651014.94169879	1373884.66067133	41.923406848524

Growth By Sales Amount
From fiscal year 2020 to 2021. By sales.
APAC - 198 %
EU - 256%
NA - 200%
LATAM - 41.95%
This shows that the EU actually **had the highest relative growth**, though from a smaller base.

Regional Profitability By Net Profit Growth

```sql
-- abosoulte growth in net_profit from the 2020 to 2021 regionwise.
with net_sales_t_2020 as
(select region, sum(net_sales) as net_sales_2020, sum(net_profit) as net_profit_2020 from regional_profitability
where fiscal_year = 2020
group by region ),
net_sales_t_2021 as
(
select region, sum(net_sales) as net_sales_2021, sum(net_profit) as net_profit_2021 from regional_profitability
where fiscal_year = 2021
group by region)
select t1.region,t1.net_sales_2020,t2.net_sales_2021,t1.net_profit_2020,t2.net_profit_2021,
net_sales_2021 - net_sales_2020 as sales_growth,
((net_sales_2021 - net_sales_2020)/(net_sales_2020)) * 100 as sales_growth_pct,
net_profit_2021 - net_profit_2020 as net_profit_growth,
((net_profit_2021 - net_profit_2020)/(net_profit_2020)) * 100 as net_profit_growth_pct
from net_sales_t_2020 t1 inner join net_sales_t_2021 t2
on t1.region = t2.region;
```

Output:

region	net_sales_2020	net_sales_2021	net_profit_2020	net_profit_2021	sales_growth	sales_growth_pct	net_profit_growth	net_profit_growth_pct
APAC	232798444.50253549	693781163.92750890	142634007.16893549	421851238.60800890	460982719.42497341	198.017955150019	279217231.43907341	195.757825907793
EU	84426507.83255441	301264613.06740311	51323558.17595441	183224255.58930311	216838105.23484870	256.836520663522	131900697.41334870	256.998349493129
NA	90748534.62394322	272435164.51616139	55029479.16054322	166506459.75506139	181686629.89221817	200.208885625666	111476980.59451817	202.576841167794
LATAM	3277130.28102746	4651014.94169879	1963834.41722746	2773017.40629879	1373884.66067133	41.923406848524	809182.98907133	41.204237076858

Net Profit grew highest for the EU Region.
EU Region Net profit growth - 256 %
APAC Region Net profit growth - 195 %
NA Region Net profit growth - 202 %
LATAM Region Net profit growth - 41%

Channel Profitability By Sales

```
-- channel profitability analysis by sales for 2020 and 2021
with g_s_2020 as
(select channel,sum(gross_sales) as gross_sales_2020 from channel_profitability where fiscal_year = 2020
group by channel),
g_s_2021 as
(select channel,sum(gross_sales) as gross_sales_2021 from channel_profitability where fiscal_year = 2021
group by channel)
select t1.channel, t1.gross_sales_2020, t2.gross_sales_2021,
t2.gross_sales_2021 - t1.gross_sales_2020,
((t2.gross_sales_2021 - t1.gross_sales_2020)/(t1.gross_sales_2020)) * 100 as gross_sales_pct
from g_s_2020 t1 inner join g_s_2021 t2 on
t1.channel = t2.channel;
```

Output:

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

channel	gross_sales_2020	gross_sales_2021	increase_or_decrease	gross_sales_pct
Direct	96431849.6141	257532002.6536	161100153.0395	167.06114596
Distributor	60439602.8302	188025630.9348	127586028.1046	211.09673481
Retailer	379074319.4577	1219081639.9472	840007320.4895	221.59436221

Retailer Channel grew the largest by sales from 2020 to 2021 by 221%.
Distributor Channel grew by 211%.
Direct Channel grew by 167%.

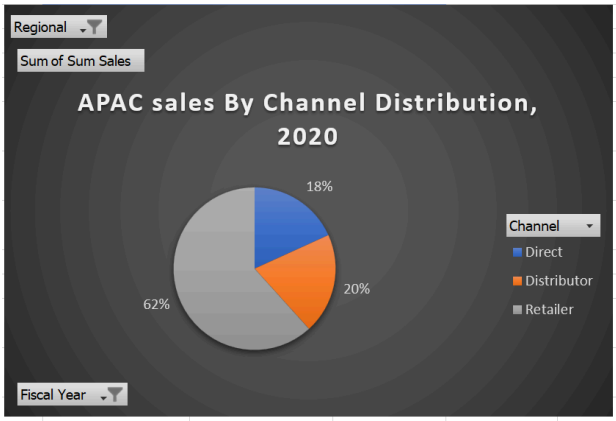
Channel Profitability By Net profit

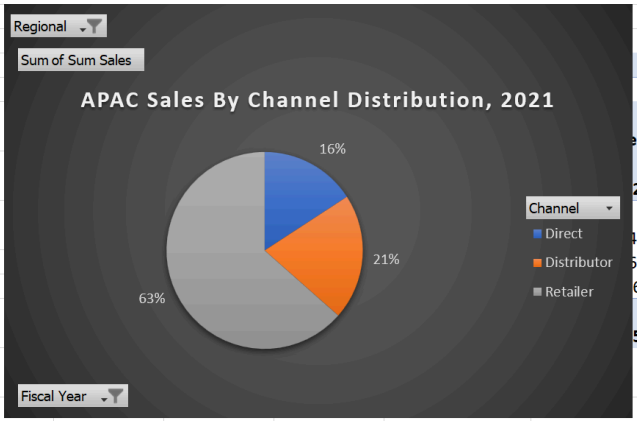
```
-- channel profitability analysis by sales for 2020 and 2021 by np.
with g_s_2020 as
(select channel,sum(gross_sales) as gross_sales_2020, sum(net_profit) as net_profit_2020 from channel_profitability where fiscal_year = 2020
group by channel),
g_s_2021 as
(select channel,sum(gross_sales) as gross_sales_2021, sum(net_profit) as net_profit_2021 from channel_profitability where fiscal_year = 2021
group by channel)
select t1.channel, t1.gross_sales_2020, t2.gross_sales_2021,
t1.net_profit_2020, t2.net_profit_2021,
t2.gross_sales_2021 - t1.gross_sales_2020 as inc_dec_g_sales,
((t2.gross_sales_2021 - t1.gross_sales_2020)/(t1.gross_sales_2020)) * 100 as gross_sales_pct_inc_or_dec,
t2.net_profit_2021 - t1.net_profit_2020 as inc_dec_net_profit,
((t2.net_profit_2021 - t1.net_profit_2020)/(t1.net_profit_2020)) * 100 as np_inc_dec_perc
from g_s_2020 t1 inner join g_s_2021 t2 on
t1.channel = t2.channel;
```

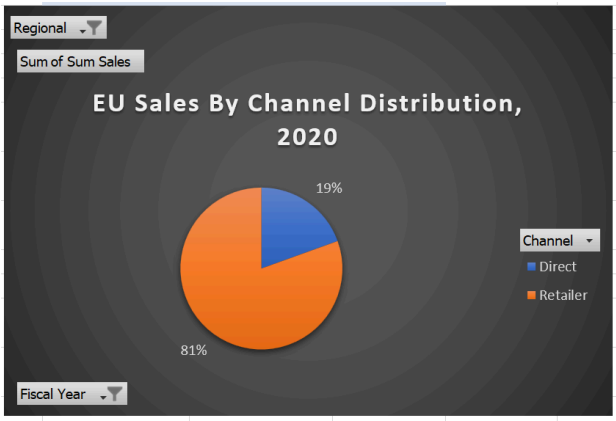
channel	gross_sales_2020	gross_sales_2021	net_profit_2020	net_profit_2021	inc_dec_g_sales	gross_sales_pct_inc_or_dec	inc_dec_net_profit	np_inc_dec_perc
Direct	96431849.6141	257532002.6536	50051111.05625722	139357434.99541961	161100153.0395	167.06114596	89306323.93916239	178.430252704645
Distributor	60439602.8302	188025630.9348	27210686.39915776	82155732.50397782	127586028.1046	211.09673481	54945046.10482006	201.924513401914
Retailer	379074319.4577	1219081639.9472	173689081.46724560	552841803.85927476	840007320.4895	221.59436221	379152722.39202916	218.293930274212

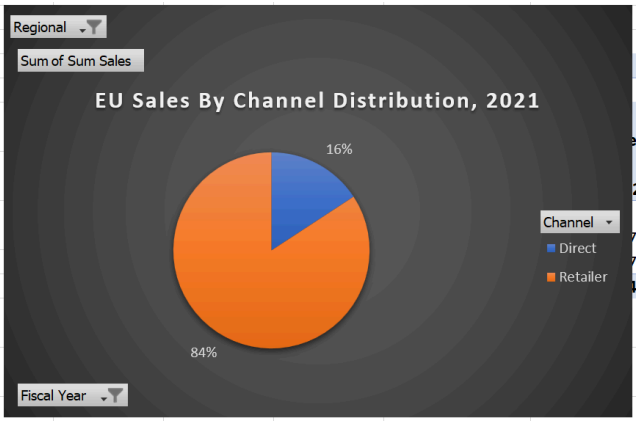
Retailer Channel got the most increase in net profit from 2020 to 2021 followed by Distributor and then Direct Channel.
Retailer channel net profit increase was 218%.
Distributor channel net profit increase was 201%.
Direct Channel net profit increase was by 178%.

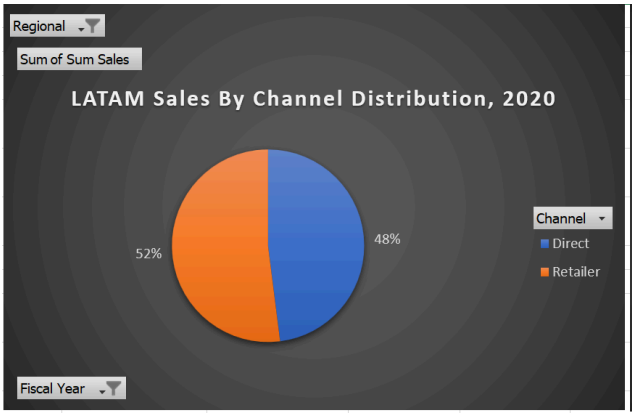
Channel And Region Mix

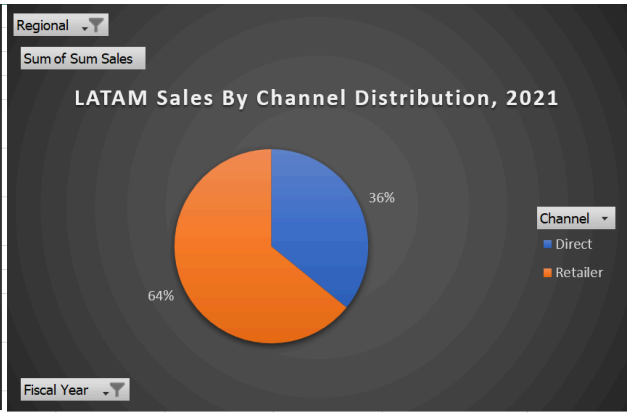


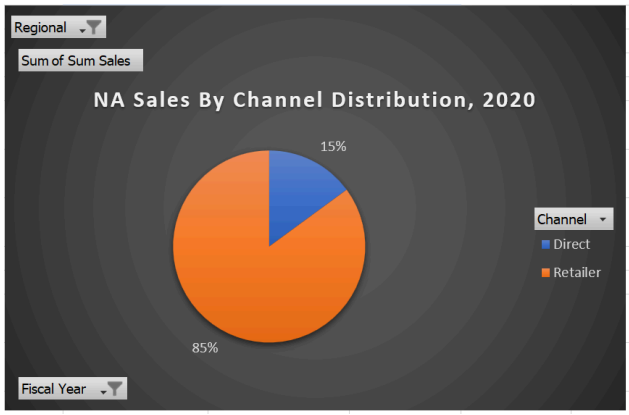


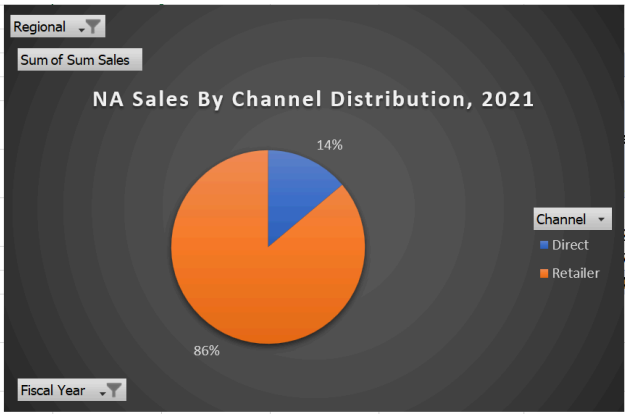












– Channel Dominance.