

```
In [1]: import numpy as np
```

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
In [2]: df_Tran=pd.read_excel("C:\\\\Users\\\\LENOVO\\\\Downloads\\\\QVI_transaction_data.xlsx")
```

```
df_Tran.head(5)
```

```
Out[2]:
```

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES
0	43390	1	1000	1	5	Natural Chip Comnpy SeaSalt175g	2	6.0
1	43599	1	1307	348	66	CCs Nacho Cheese 175g	3	6.3
2	43605	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	2.9
3	43329	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5	15.0
4	43330	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpmo Chili 150g	3	13.8

```
In [3]: df_Tran.shape
```

```
Out[3]: (264836, 8)
```

```
In [4]: df_purchase_bhe=pd.read_csv("C:\\\\Users\\\\LENOVO\\\\Downloads\\\\QVI_purchase_behaviour.csv")
```

```
df_purchase_bhe.head(5)
```

```
Out[4]:
```

	LYLTY_CARD_NBR	LIFESTAGE	PREMIUM_CUSTOMER
0	1000	YOUNG SINGLES/COUPLES	Premium
1	1002	YOUNG SINGLES/COUPLES	Mainstream
2	1003	YOUNG FAMILIES	Budget
3	1004	OLDER SINGLES/COUPLES	Mainstream
4	1005	MIDAGE SINGLES/COUPLES	Mainstream

```
In [5]: df_purchase_bhe.shape
```

```
Out[5]: (72637, 3)
```

```
In [6]: df_Tran.isnull().sum()
```

```
Out[6]:
```

DATE	0
STORE_NBR	0
LYLTY_CARD_NBR	0
TXN_ID	0
PROD_NBR	0
PROD_NAME	0
PROD_QTY	0
TOT_SALES	0
dtype: int64	

```
In [7]: df_purchase_bhe.isnull().sum()
```

```
Out[7]:
```

LYLTY_CARD_NBR	0
LIFESTAGE	0
PREMIUM_CUSTOMER	0
dtype: int64	

There is no null values in this dataset

```
In [8]: df_Tran["DATE"] = pd.to_datetime(df_Tran["DATE"], unit="D", origin="1899-12-30")
```

```
In [9]: df_Tran["year"] = df_Tran["DATE"].dt.year
```

```
In [10]: df_Tran["month_name"] = df_Tran["DATE"].dt.strftime("%b")
```

Here I decoded the date column in date format and then extract the month and year

```
In [11]: df_Tran["PROD_NAME"] = df_Tran["PROD_NAME"].str.replace("Burger Rings 220g", "Smit's Burger Rings 220g")
```

```
In [12]: df_Tran["PROD_NAME"] = df_Tran["PROD_NAME"].str.replace("Kettle 135g Swt Pot Sea Salt", "Kettle Swt Pot Sea Salt 135g")
```

```
In [13]: df_Tran["PROD_NAME"] = df_Tran["PROD_NAME"].str.replace("Old El Paso Salsa ", "OldElPasoSalsa ")
```

```
In [14]: df_Tran["PROD_NAME"] = df_Tran["PROD_NAME"].str.replace("Twisties Chicken270g", "Twisties Chicken 270g ")
```

```
In [15]: df_Tran["PROD_NAME"] = df_Tran["PROD_NAME"].str.replace("Natural Chip Comnpy SeaSalt175g", "Natural Chip Comnpy SeaSalt 175g ")
```

```
In [16]: df_Tran["Packet_size(gm)"] = df_Tran["PROD_NAME"].str.extract(r"(\d+(?:(?G)))")
```

```
In [17]: df_Tran["Company_names"] = df_Tran["PROD_NAME"].str.extract(r"(\w+)")
```

```
In [18]: df_Tran['PROD_NAME'] = df_Tran['PROD_NAME'].str.strip().str.replace(r'\s+', ' ', regex=True)
```

```
In [19]: df_Tran["product_n"] = df_Tran["PROD_NAME"].apply(lambda x: ' '.join(x.split(' ')[1:-1]))
```

Change the Date column into Actual Date.

Extract month name , year, company names and packet size from it.

Change some product name for simple analysis purpose.

```
In [20]: df_Tran.head(10)
```

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	year	month_name	Packet_size(gm)
0	2018-10-17	1	1000	1	5	Natural Chip Comnpy SeaSalt 175g	2	6.0	2018	Oct	175g
1	2019-05-14	1	1307	348	66	CCs Nacho Cheese 175g	3	6.3	2019	May	175g
2	2019-05-20	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	2.9	2019	May	170g
3	2018-08-17	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5	15.0	2018	Aug	175g
4	2018-08-18	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpo Chili 150g	3	13.8	2018	Aug	150g
5	2019-05-19	4	4074	2982	57	OldElPasoSalsa Dip Tomato Mild 300g	1	5.1	2019	May	300g
6	2019-05-16	4	4149	3333	16	Smiths Crinkle Chips Salt & Vinegar 330g	1	5.7	2019	May	330g
7	2019-05-16	4	4196	3539	24	Grain Waves Sweet Chilli 210g	1	3.6	2019	May	210g
8	2018-08-20	5	5026	4525	42	Doritos Corn Chip Mexican Jalapeno 150g	1	3.9	2018	Aug	150g
9	2018-08-18	7	7150	6900	52	Grain Waves Sour Cream&Chives 210G	2	7.2	2018	Aug	210G

In [21]: `pd.set_option("display.max_columns",None)`

```
pd.set_option("display.max_rows",None)
```

```
In [22]: pd.DataFrame(df_Tran["product_n"].unique())
```

Out[22]:

0

- |    |                              |
|----|------------------------------|
| 0  | Chip Comnpy SeaSalt          |
| 1  | Nacho Cheese                 |
| 2  | Crinkle Cut Chips Chicken    |
| 3  | Chip Thinly S/Cream&Onion    |
| 4  | Tortilla ChpsHny&Jlpno Chili |
| 5  | Dip Tomato Mild              |
| 6  | Crinkle Chips Salt & Vinegar |
| 7  | Waves Sweet Chilli           |
| 8  | Corn Chip Mexican Jalapeno   |
| 9  | Waves Sour Cream&Chives      |
| 10 | Sensations Siracha Lime      |
| 11 | Cheese                       |
| 12 | Crinkle Cut Chicken          |
| 13 | Chips Light& Tangy           |
| 14 | Original                     |
| 15 | Burger Rings                 |
| 16 | Sour Cream & Garden Chives   |
| 17 | Corn Chip Southern Chicken   |
| 18 | Cheese Box                   |
| 19 | Crinkle Original             |
| 20 | Crn Crnchers Tangy Gcamole   |
| 21 | Sea Salt And Vinegar         |
| 22 | Chip Thinly Cut Original     |
| 23 | Rock Deli Thai Chilli&Lime   |
| 24 | Sthrn FriedChicken           |

**0**

- |           |                              |
|-----------|------------------------------|
| <b>25</b> | Sweet&Spicy BBQ              |
| <b>26</b> | Rock Deli SR Salsa & Mzrrlla |
| <b>27</b> | Chips Originl salted         |
| <b>28</b> | Rock Deli Sp Salt & Truffle  |
| <b>29</b> | Thinly Swt                   |
| <b>30</b> | Chilli                       |
| <b>31</b> | Mexicana                     |
| <b>32</b> | Crinkle Cut French OnionDip  |
| <b>33</b> | ChipCo Honey Soy             |
| <b>34</b> | Corn Chp Supreme             |
| <b>35</b> | Chicken                      |
| <b>36</b> | Thinly Cut Roast Chicken     |
| <b>37</b> | Crinkle Cut Tomato Salsa     |
| <b>38</b> | Mozzarella Basil & Pesto     |
| <b>39</b> | Thai SweetChili PotatoMix    |
| <b>40</b> | Sensations Camembert & Fig   |
| <b>41</b> | Crinkle Cut Mac N Cheese     |
| <b>42</b> | Honey Soy Chicken            |
| <b>43</b> | Chips Seasonedchicken        |
| <b>44</b> | Crinkle Cut Salt & Vinegar   |
| <b>45</b> | BBQ Rib Prawn Crackers       |
| <b>46</b> | Plus Btroot & Chilli Jam     |
| <b>47</b> | Crisps Lightly Salted        |
| <b>48</b> | Sweet Chilli And Sour Cream  |
| <b>49</b> | Salsa Medium                 |

**0**

- |           |                                |
|-----------|--------------------------------|
| <b>50</b> | Swt Pot Sea Salt               |
| <b>51</b> | SourCream Onion                |
| <b>52</b> | Corn Chips Original            |
| <b>53</b> | Cheese Burger                  |
| <b>54</b> | Dip Chnky Tom                  |
| <b>55</b> | Popd Swt/Chlli &Sr/Cream Chips |
| <b>56</b> | Mild Salsa                     |
| <b>57</b> | Chip Co Tmato Hrb&Spce         |
| <b>58</b> | Crinkle Cut Chips Original     |
| <b>59</b> | Popd Sea Salt Chips            |
| <b>60</b> | Crinkle Cut Chips              |
| <b>61</b> | Fries Potato Chips             |
| <b>62</b> | Dip Tomato Med                 |
| <b>63</b> | Corn Chips Cheese Supreme      |
| <b>64</b> | Original Crisps                |
| <b>65</b> | Chilli& Coconut                |
| <b>66</b> | Original Corn Chips            |
| <b>67</b> | Potato Chips Hot & Spicy       |
| <b>68</b> | Popd Sour Crm &Chives Chips    |
| <b>69</b> | Crnkle Chip Orgnl Big Bag      |
| <b>70</b> | Corn Chips Nacho Cheese        |
| <b>71</b> | Sensations BBQ&Maple           |
| <b>72</b> | D/Style Chip Sea Salt          |
| <b>73</b> | Chicken Salt Crips             |
| <b>74</b> | Original Stacked Chips         |

**0**

- |           |                             |
|-----------|-----------------------------|
| <b>75</b> | Chip Thinly                 |
| <b>76</b> | Lightly Salted              |
| <b>77</b> | Chips Salt & Vinegar        |
| <b>78</b> | Crinkle Cut Chips Barbecue  |
| <b>79</b> | Puffs                       |
| <b>80</b> | Sweet Chilli & Sour Cream   |
| <b>81</b> | Crinkle Cut Original        |
| <b>82</b> | Splash Of Lime              |
| <b>83</b> | Medium Salsa                |
| <b>84</b> | Tortilla ChpsBtroot&Ricotta |
| <b>85</b> | Tasty Cheese                |
| <b>86</b> | Cheese Rings                |
| <b>87</b> | Smoked Chipotle             |
| <b>88</b> | Barbeque                    |
| <b>89</b> | Supreme Cheese Corn Chips   |
| <b>90</b> | Mystery Flavour             |
| <b>91</b> | Crisps Ched & Chives        |
| <b>92</b> | Whlgrn Crisps Cheddr&Mstrd  |
| <b>93</b> | Chs & Bacon Balls           |
| <b>94</b> | Slt Vingar                  |
| <b>95</b> | SourCream&Herbs Veg Strws   |
| <b>96</b> | Tortilla ChpsFeta&Garlic    |
| <b>97</b> | Mango Chutny Papadums       |
| <b>98</b> | Steak & Chimuchurri         |
| <b>99</b> | Whlegrn Crisps Frch/Onin    |

0	
100	Salt & Vinegar
101	Cheese Supreme
102	Crinkle Cut Snag&Sauce
103	Sour Cream &OnionStacked Chips
104	Lime & Pepper
105	ChipCo Sea Salt & Vinegr
106	Rock Deli Chikn&Garlic Aioli
107	SR Slow Rst Pork Belly
108	Pc Sea Salt
109	Crinkle Cut Bolognese
110	Salsa Mild

```
In [23]: df_Tran["Packet_size(gm)"] = df_Tran["Packet_size(gm)"].str.extract(r'(\d+)').astype(int) # get the packet size a numeric format
```

```
In [24]: #df Tran.head(10)
```

```
In [25]: new_data=df_Tran.merge(df_purchase_bhe)# merge the both data
```

```
In [26]: new_df=new_data.drop(columns="TXN_ID") # drop the TXN_ID column
```

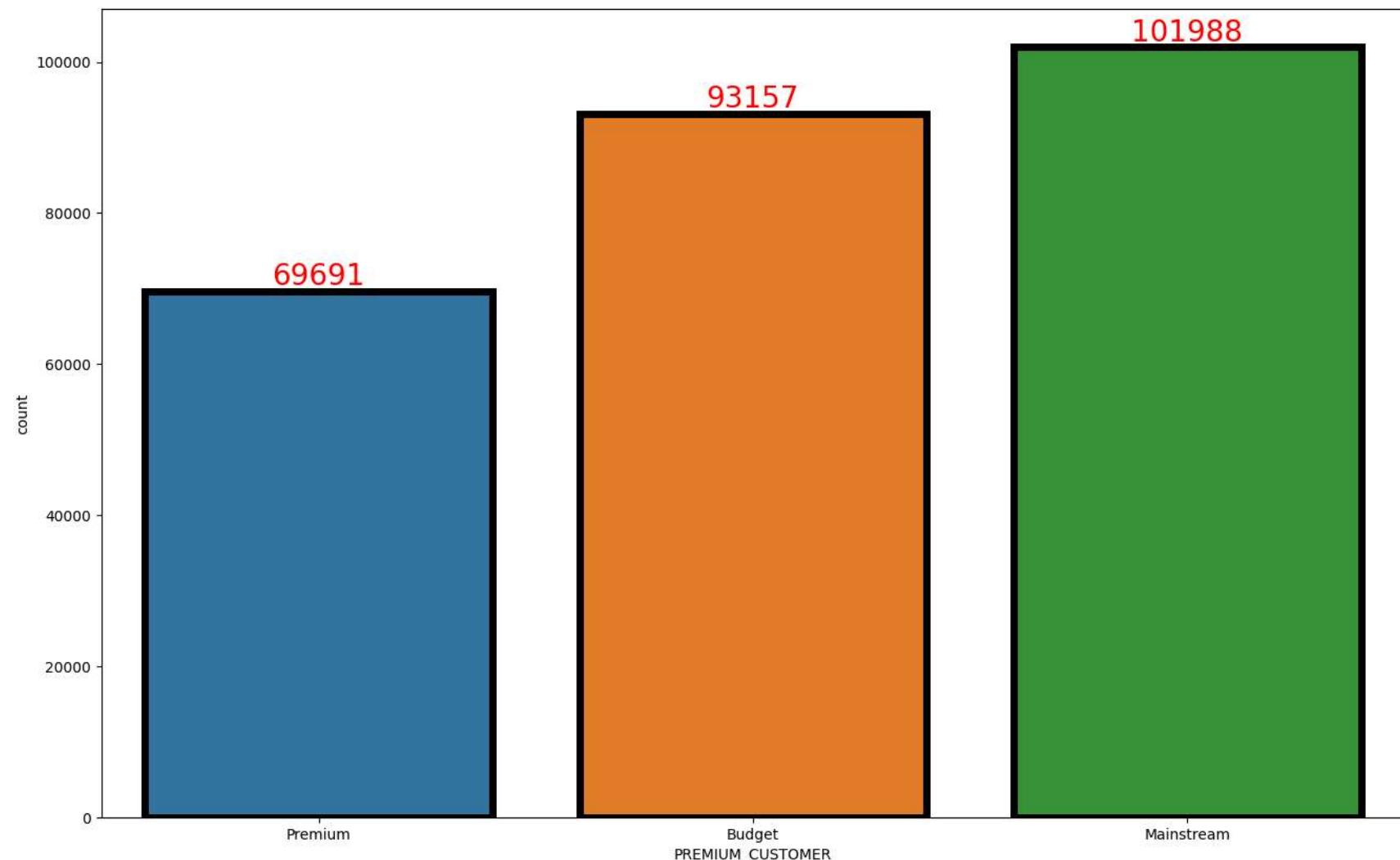
```
In [27]: new_df=new_data.drop(columns="PROD_NAME") # drop the PROD_NAME column
```

```
In [28]: new_df["Price"] = new_df["TOT_SALES"] / new_df["PROD_QTY"]
```

```
In [29]: #new_df.tail(10)
```

# Visualization

```
In [30]:  
plt.figure(figsize=(16,10))  
ax=sns.countplot(new_data,x=new_df[ "PREMIUM_CUSTOMER"],edgecolor="k", linewidth=5)  
ax.bar_label(ax.containers[0],size=20,color="r")  
plt.show()
```



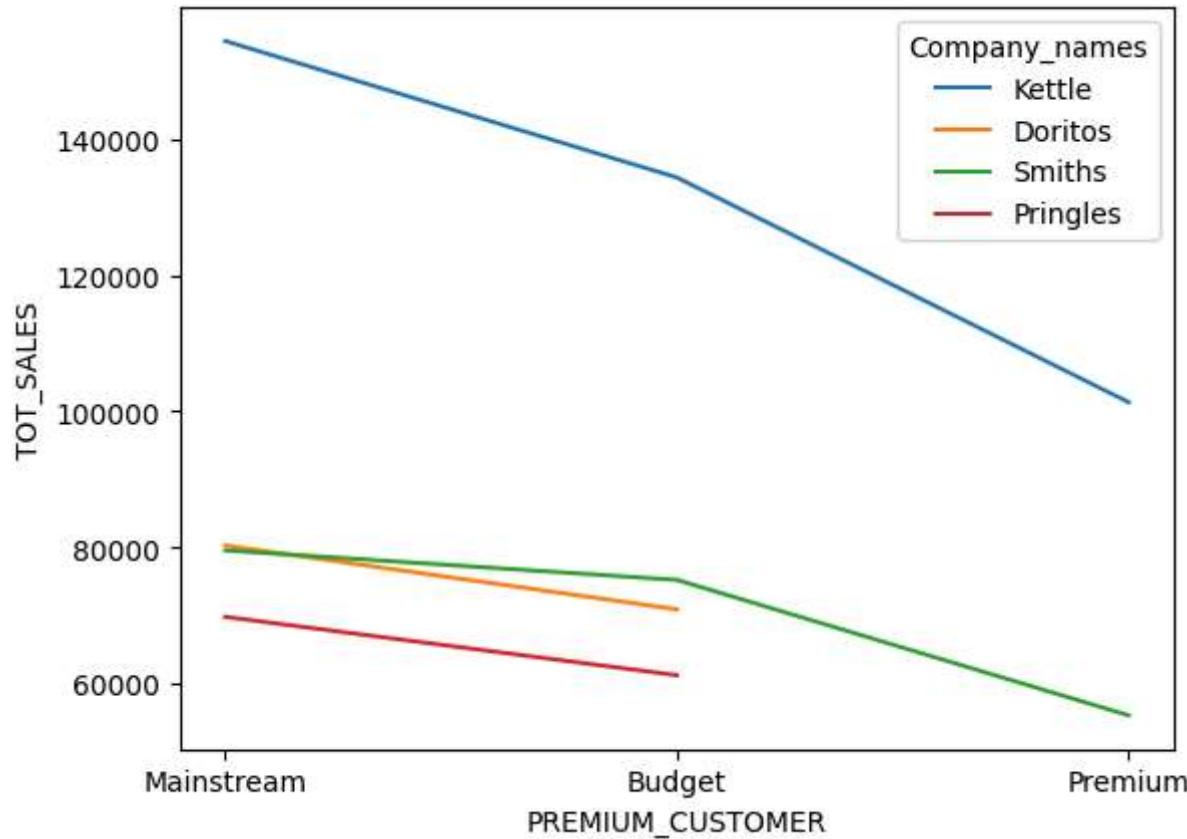
```
In [31]: Top_companies_highest_sales=new_data.groupby(["Company_names","PREMIUM_CUSTOMER"]).agg({"TOT_SALES": "sum"}).reset_index().sort_values("TOT_SALES", ascending=False)
Top_10_companies=Top_companies_highest_sales.nlargest(10,"TOT_SALES")
Top_10_companies ## in top 10 companies kettle, doritos , smiths, pringles are cover in top 10
```

```
Out[31]:
```

	Company_names	PREMIUM_CUSTOMER	TOT_SALES
34	Kettle	Mainstream	154477.0
33	Kettle	Budget	134407.8
35	Kettle	Premium	101355.0
16	Doritos	Mainstream	80272.9
61	Smiths	Mainstream	79560.0
60	Smiths	Budget	75219.0
15	Doritos	Budget	70893.0
46	Pringles	Mainstream	69785.7
45	Pringles	Budget	61201.7
62	Smiths	Premium	55297.8

```
In [32]: sns.lineplot(x=Top_10_companies["PREMIUM_CUSTOMER"],y=Top_10_companies["TOT_SALES"],hue=Top_10_companies["PREMIUM_CUSTOMER"])
```

```
Out[32]: <Axes: xlabel='PREMIUM_CUSTOMER', ylabel='TOT_SALES'>
```



```
In [33]: Top_companies_lowest_sales=new_data.groupby(["Company_names","PREMIUM_CUSTOMER"]).agg({"TOT_SALES": "sum"}).reset_index().sort_values("TOT_SALES", ascending=True)
Top_20_companies=Top_companies_lowest_sales.nsmallest(20,"TOT_SALES")
Top_20_companies
```

```
Out[33]:
```

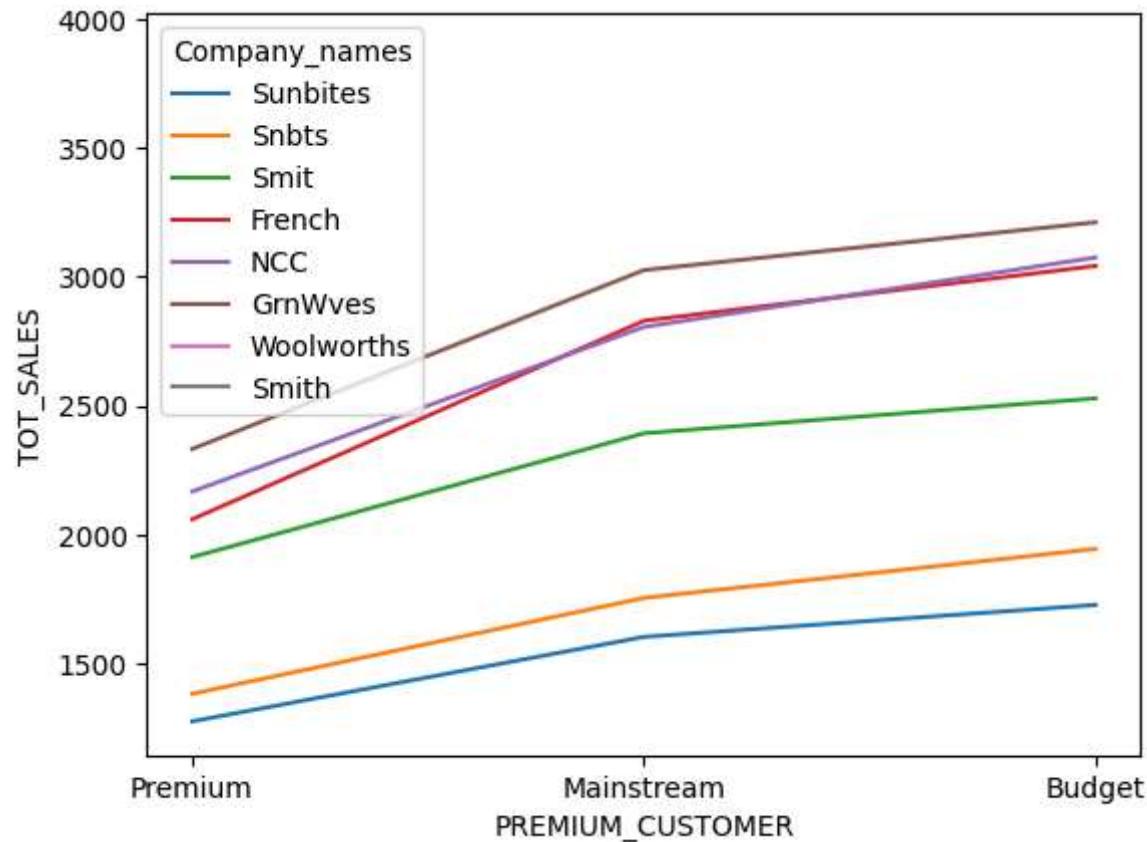
	Company_names	PREMIUM_CUSTOMER	TOT_SALES
68	Sunbites	Premium	1273.3
65	Snbts	Premium	1380.4
67	Sunbites	Mainstream	1601.4
66	Sunbites	Budget	1725.5
64	Snbts	Mainstream	1752.7
56	Smit	Premium	1911.3
63	Snbts	Budget	1943.1
20	French	Premium	2058.0
38	NCC	Premium	2166.0
26	GrnWves	Premium	2331.2
55	Smit	Mainstream	2392.0
54	Smit	Budget	2527.7
37	NCC	Mainstream	2805.0
19	French	Mainstream	2829.0
25	GrnWves	Mainstream	3025.6
18	French	Budget	3042.0
36	NCC	Budget	3075.0
24	GrnWves	Budget	3211.6
86	Woolworths	Premium	3712.5
59	Smith	Premium	3892.2

```
In [34]:
```

```
sns.lineplot(x=Top_20_companies["PREMIUM_CUSTOMER"],y=Top_20_companies["TOT_SALES"],hue=Top_20_companies["
```

```
Out[34]:
```

```
<Axes: xlabel='PREMIUM_CUSTOMER', ylabel='TOT_SALES'>
```



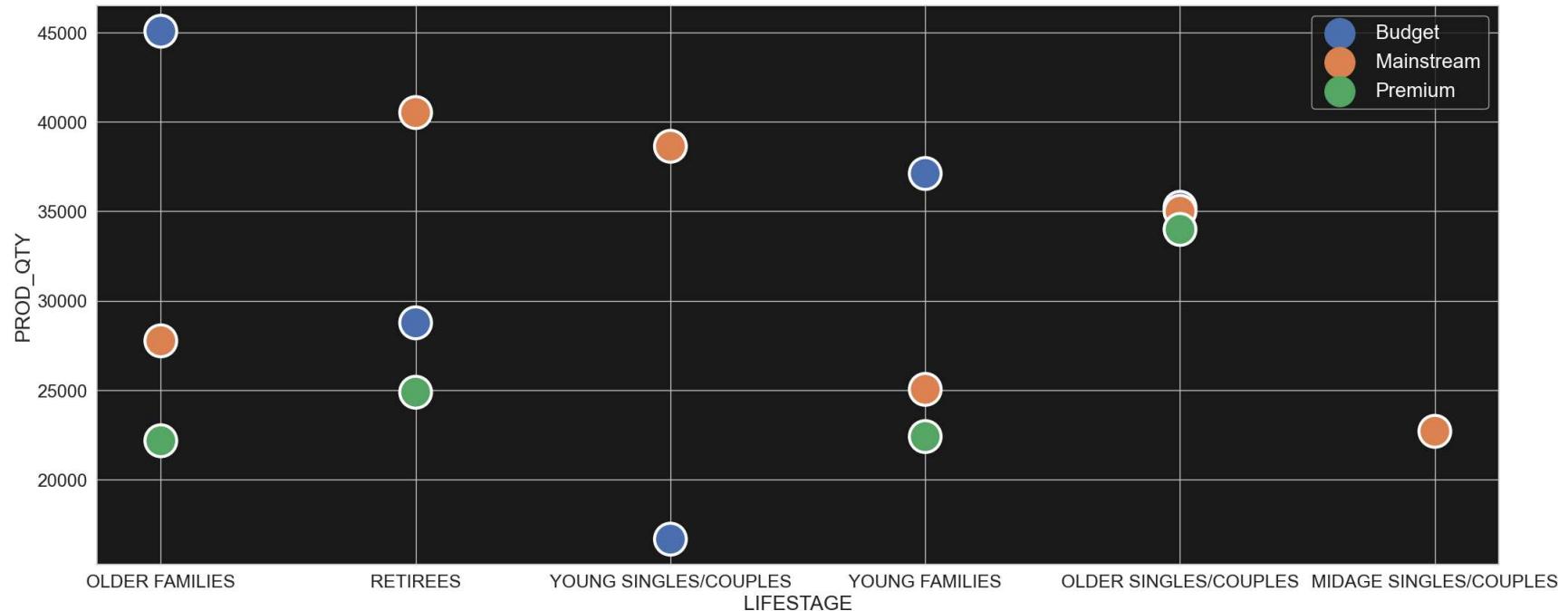
```
In [35]: highest_quantity=new_data.groupby(["LIFESTAGE","PREMIUM_CUSTOMER"]).agg({"PROD_QTY":"sum"}).reset_index().  
quantity=highest_quantity.nlargest(15,"PROD_QTY")  
quantity
```

Out[35]:

	LIFESTAGE	PREMIUM_CUSTOMER	PROD_QTY
6	OLDER FAMILIES	Budget	45065
13	RETIREES	Mainstream	40518
19	YOUNG SINGLES/COUPLES	Mainstream	38632
15	YOUNG FAMILIES	Budget	37111
9	OLDER SINGLES/COUPLES	Budget	35220
10	OLDER SINGLES/COUPLES	Mainstream	34997
11	OLDER SINGLES/COUPLES	Premium	33986
12	RETIREES	Budget	28764
7	OLDER FAMILIES	Mainstream	27756
16	YOUNG FAMILIES	Mainstream	25044
14	RETIREES	Premium	24884
1	MIDAGE SINGLES/COUPLES	Mainstream	22699
17	YOUNG FAMILIES	Premium	22406
8	OLDER FAMILIES	Premium	22171
18	YOUNG SINGLES/COUPLES	Budget	16671

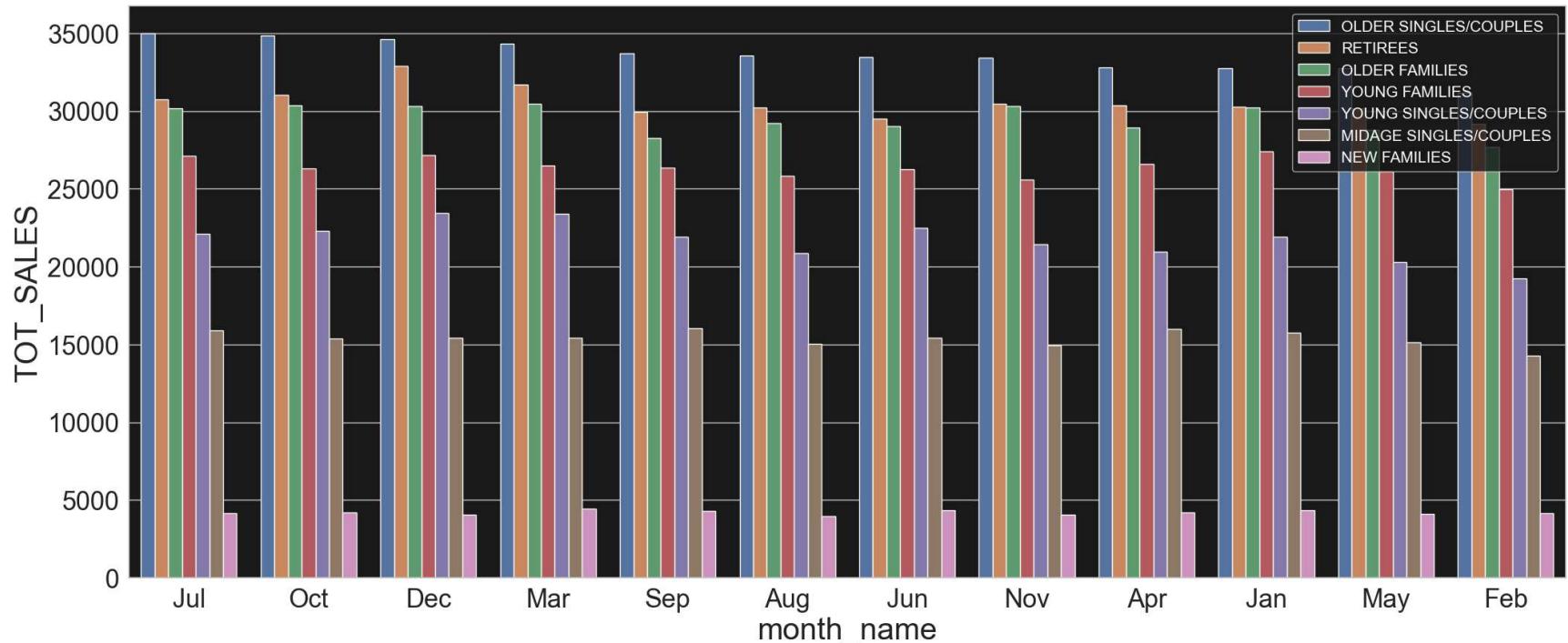
In [36]:

```
plt.figure(figsize=(25,10))
sns.set(style="whitegrid")
ax=sns.scatterplot(x=quantity["LIFESTAGE"],y=quantity["PROD_QTY"],hue=quantity["PREMIUM_CUSTOMER"],s=1000,
                    edgecolor="w", linewidth=3)
ax.set_facecolor("k")
plt.tick_params(axis="both",labelsize=18)
plt.legend(fontsize=20,facecolor="k",labelcolor="w",markerscale=5)
plt.xlabel("LIFESTAGE",size=20)
plt.ylabel("PROD_QTY",size=20)
plt.show()
```



```
In [37]: month_sales=new_data.groupby(["month_name","LIFESTAGE"]).agg({"TOT_SALES":"sum"}).reset_index().sort_values("TOT_SALES", ascending=False)  
#month_sales
```

```
In [38]: plt.figure(figsize=(25,10))  
sns.set(style="whitegrid")  
ax=sns.barplot(x=month_sales["month_name"],y=month_sales["TOT_SALES"],hue=month_sales["LIFESTAGE"],errorbar=None)  
plt.tick_params(axis="both",labelsize=18)  
ax.set_facecolor("k")  
plt.legend(fontsize=15,facecolor="k",labelcolor="w",markerscale=5)  
plt.tick_params(axis="both",labelsize=25)  
plt.ylabel("TOT_SALES",size=30)  
plt.xlabel("month_name",size=30)  
plt.show()
```



```
In [39]: Top_products=new_df.groupby(["LIFESTAGE","product_n"]).agg({"TOT_SALES":"sum"}).sort_values("TOT_SALES",ascending=False)
Top_products_10=Top_products.nlargest(5,"TOT_SALES")
Top_products_10# these age people like older, young families or couples or singles like cheese .
```

Out[39]:

### TOT\_SALES

LIFESTAGE	product_n	TOT_SALES
OLDER SINGLES/COUPLES	Cheese	12768.1
RETIREES	Cheese	12077.2
OLDER FAMILIES	Cheese	10478.6
YOUNG FAMILIES	Cheese	9980.0
YOUNG SINGLES/COUPLES	Cheese	8665.9

```
In [40]: Top_products=new_df.groupby(["LIFESTAGE","product_n","PREMIUM_CUSTOMER"]).agg({"TOT_SALES":"sum"}).sort_values("TOT_SALES", ascending=False)
Top_products_10=Top_products.nsmallest(10,"TOT_SALES")
Top_products_10
## our Lowest sales contribution in these product like medium salsa , chilli coconut
##comes from who are the newfamilies but mostly premium .
```

Out[40]:

LIFESTAGE	product_n	PREMIUM_CUSTOMER	TOT_SALES
NEW FAMILIES	Crinkle Cut Chicken	Premium	5.1
	Whlgrn Crisps Cheddr&Mstrd	Premium	11.9
	Fries Potato Chips	Premium	15.0
	Mild Salsa	Premium	15.0
	Whlegrn Crisps Frch/Onin	Mainstream	15.3
	Chilli& Coconut	Premium	16.2
	Medium Salsa	Mainstream	16.5
		Premium	16.5
	Chip Compy SeaSalt	Premium	18.0
	Medium Salsa	Budget	19.5

```
In [41]: top_size=new_df.groupby(["Packet_size(gm)"]).agg({"TOT_SALES":"sum"}).sort_values("TOT_SALES",ascending=False)
top_size_5=top_size.nlargest(5,"TOT_SALES")
top_size_5 # our highest packet size sales mosly within 175,150,134
```

Out[41]:

### TOT\_SALES

#### Packet\_size(gm)

<b>175</b>	485437.4
<b>150</b>	304288.5
<b>134</b>	177655.5
<b>110</b>	162765.4
<b>170</b>	146673.0

In [42]:

```
Top_companies_quantity=new_df.groupby([ "Company_names", "product_n"]).agg({"PROD_QTY":"sum"}).sort_values('PROD_QTY', ascending=False).nlargest(5, "PROD_QTY")  
TOP=Top_companies_quantity  
TOP
```

Out[42]:

### PROD\_QTY

#### Company\_names

#### product\_n

<b>Dorito</b>	<b>Corn Chp Supreme</b>	6509
<b>Kettle</b>	<b>Mozzarella Basil &amp; Pesto</b>	6381
	<b>Tortilla ChpsHny&amp;Jlpno Chili</b>	6309
<b>Cobs</b>	<b>Popd Sea Salt Chips</b>	6277
	<b>Popd Swt/Chlli &amp;Sr/Cream Chips</b>	6256

In [43]:

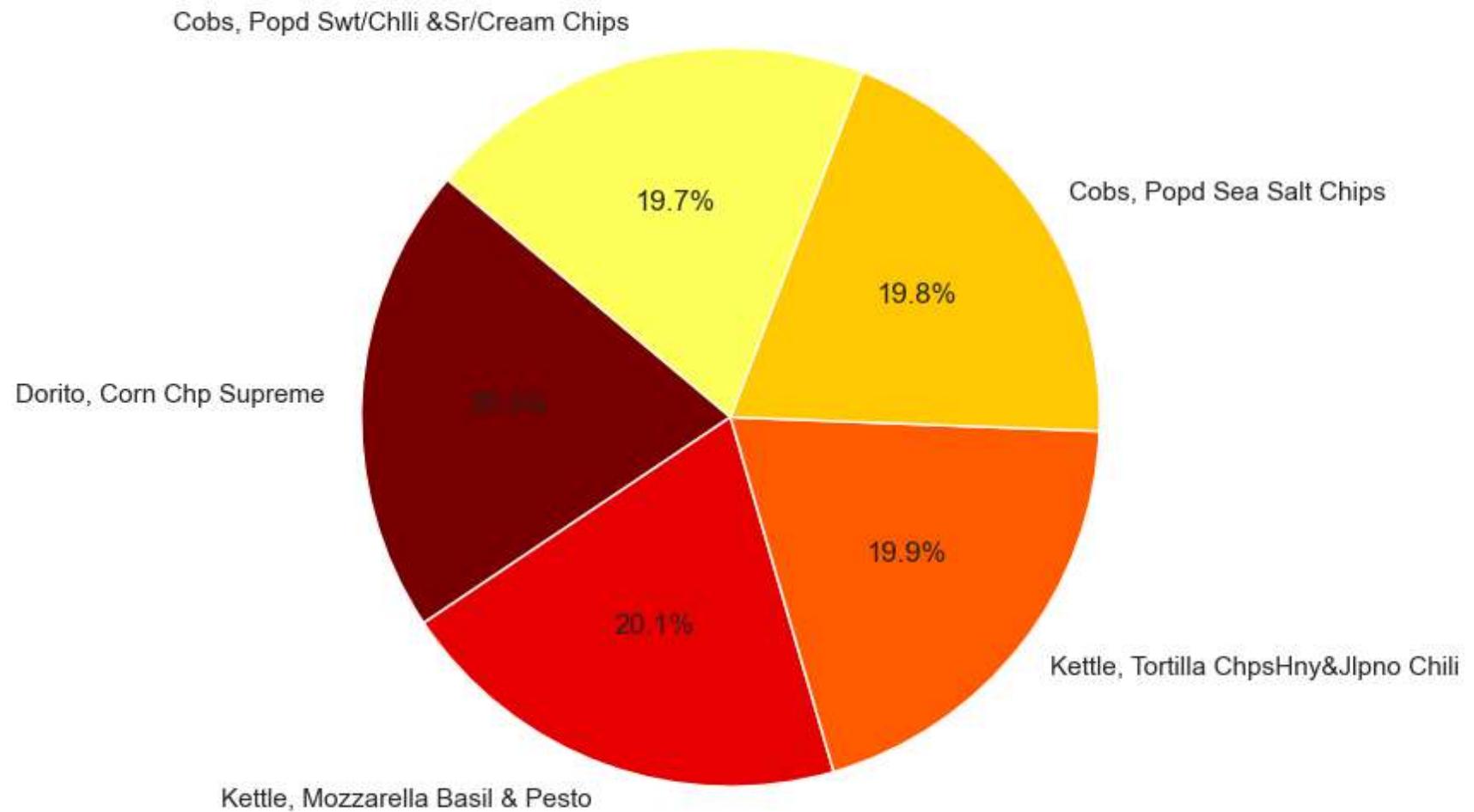
```
labels = [f'{company}, {product}' for company, product in TOP.index]  
labels
```

Out[43]:

```
['Dorito, Corn Chp Supreme',  
'Kettle, Mozzarella Basil & Pesto',  
'Kettle, Tortilla ChpsHny&Jlpno Chili',  
'Cobs, Popd Sea Salt Chips',  
'Cobs, Popd Swt/Chlli &Sr/Cream Chips']
```

```
In [44]: plt.figure(figsize=(10, 7))
TOP['PROD_QTY'].plot.pie(autopct='%0.1f%%', startangle=140, colors=sns.color_palette("hot", 5),
labels=labels)
plt.title('Top Products with Highest Sales')
plt.ylabel('') # Hide the y-label
plt.show()
```

### Top Products with Highest Sales



```
In [45]: new_df.head(5)
```

Out[45]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_QTY	TOT_SALES	year	month_name	Packet_size(gm)	Company_names
0	2018-10-17	1	1000	1	5	2	6.0	2018	Oct	175	Natural
1	2019-05-14	1	1307	348	66	3	6.3	2019	May	175	CCs
2	2018-11-10	1	1307	346	96	2	3.8	2018	Nov	160	WW
3	2019-03-09	1	1307	347	54	1	2.1	2019	Mar	175	CCs
4	2019-05-20	1	1343	383	61	2	2.9	2019	May	170	Smiths

In [46]:

```
filtered_df = new_df[new_df['year'] == 2018]
```

In [47]:

```
year_base=filtered_df.groupby(["year","PREMIUM_CUSTOMER","product_n"]).agg({"TOT_SALES":"sum"}).sort_values("TOT_SALES", ascending=False)
year_base
```

Out[47]:

year	PREMIUM_CUSTOMER	TOT_SALES	
		product_n	
2018	Mainstream	Cheese	12343.5
		Budget	10493.1
	Mainstream	Honey Soy Chicken	8319.0
	Premium	Cheese	7818.7
	Mainstream	Crinkle Chip Orgnl Big Bag	7392.7

In [48]:

```
filtered_df_2 = new_df[new_df['year'] == 2019]
```

```
In [49]: year_base_2019=filtered_df_2.groupby(["year","PREMIUM_CUSTOMER","product_n"]).agg({"TOT_SALES":"sum"}).sort_values("TOT_SALES", ascending=False)
year_base_2019## based on the last two years .in top 5 product are cheese honey soy chicken ,but for
mainstreem
## (middle class) it become corn chip supreme instead of crinkle chip origin big bag
```

Out[49]:

year	PREMIUM_CUSTOMER	TOT_SALES	
		product_n	
2019	Mainstream	Cheese	12054.20
		Budget	10796.40
		Premium	8363.40
	Mainstream	Corn Chp Supreme	7978.75
		Honey Soy Chicken	7850.40

```
In [50]: Companies_Yearly_sales=new_df.groupby(["year","Company_names"]).agg({"TOT_SALES":"sum"}).sort_values("TOT_SALES", ascending=False)
Companies_Yearly_sales
```

Out[50]:

**TOT\_SALES**

year	Company_names	TOT_SALES
2018	Kettle	197532.4
2019	Kettle	192707.4
2018	Smiths	105289.3
2019	Smiths	104787.5
2018	Doritos	102479.1
2019	Doritos	99059.8
2018	Pringles	90453.9
2019	Pringles	87201.6
2018	OldElPasoSalsa	46323.3
	Thins	45777.6
2019	OldElPasoSalsa	44461.8
	Thins	43074.9
2018	Twisties	41350.8
2019	Twisties	40171.3
2018	Tostitos	39996.0

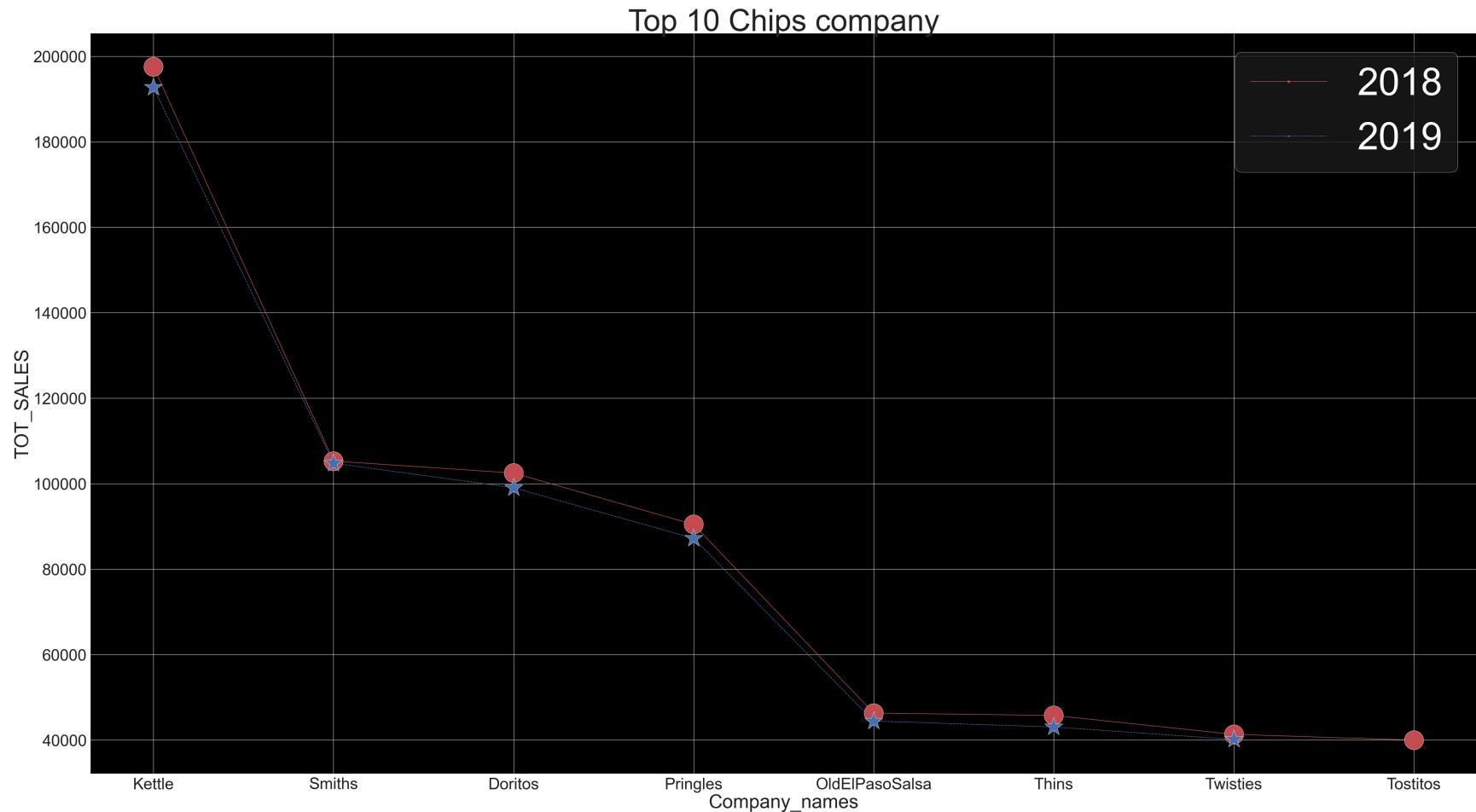
In [51]:

```
plt.figure(figsize=(65,35))
sns.set(style="darkgrid")
ax=sns.lineplot(data=Companies_Yearly_sales, x="Company_names",y="TOT_SALES",hue="year",markers=
["o","*"],
                  style="year",markersize=50,palette=["r","b"],alpha=1,errorbar=None)
ax.set_facecolor("black")
plt.legend(fontsize=100,facecolor="k",labelcolor="w")
plt.xlabel("Company_names",size=50,)
plt.ylabel("TOT_SALES",size=50)
```

```

plt.tick_params(axis="both",labelsize=42)
plt.title("Top 10 Chips company",size=80)
plt.show() ## here I can Tell that by kettle, pringals and Thins sales got slightly down compare the previous year.
## Tostios doesn't come around in year top 15 sales in the year 2019

```



# Potato Chips Market Segmentation Analysis

**Let's Dive into this data where you can taste  
different flavours .**

## Customer Segmentation and Sales Trends



# Customer Insights

- **Mainstream(Middle-Class) customers have the highest count and purchase more chips.**
- **They mostly prefer kettle brand chips. As the Kettle sales got high after october**
- **Products like sunbites, NCC, Smith these products are low sales to all segment of customers.**

# Demographic Insights



- Middle -aged , single/couples from the mainstream have chip transaction quantities below 25000.
- older families from the poor class have a high contribution to chip quantity , around 50000 out of 505124.
- Older singles/Couples from the middle class or rich class have total quantity purchase between 35000 to 40000.
- Retires and young couples they have contributed total sales around 40000 to 45000.

# Seasonal Trends



- Retired people from Middle class and older singles /couples have high purchases in December ,October and July,



# Product Sales

- Premium Customers prefer top packet size of 330G and 380g.
- New families from Premium (Rich class) their lowest buy are Chili coconut and salsa.
- Cheese which is highest sales are popular all the aged people except Middle age and new families.



- Based on the last two years , the top 5 products are cheese, honey Soy chicken , but for mainstream (MiddleClass), It becomes Corn chip supreme instead of Crinkle chip origin Big Bag.

*Top products*

# Key Factors



**High Demand For  
Cheese chips**

Premium Customers  
prefer larger packet size .

**Middle aged couples  
and older families are  
health Conscious.  
Develop new flavors.**

**Oct, Nov, Dec these months has  
highest sales**  
  
Dorito's, Corn Supreme,  
Kettle's, Mozzarella Basil & Pesto  
Kettle's, Tortilla ChpsHny&Jlno Chili',  
Cobs's, Popd Sea Salt Chips,  
Cobs's, Popd Swt/Chlli &Sr/Cream  
Chips. These Chips has highest  
quantity sales.

# Suggestions

- Ensure Sufficient inventory For this products
- Focus middle class customers as they are largest segment.
- Expand the cheese products line.
- Promote Corn chip supreme.

# Thank you!

