In [54]:
```
import pandas as pd
from tabulate import tabulate as tb
import numpy as np
import plotly.express as px
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.graph_objects as go
from plotly.subplots import make_subplots
from IPython.display import display_html
```

In [2]:
```
df1 = pd.read_csv('C:/Users/sahil/Downloads/pizza_order_details.csv')
df2 = pd.read_csv('C:/Users/sahil/Downloads/orders.csv')
# df3 = pd.read_csv('C:/Users/sahil/Downloads/pizza_types.csv')
df4 = pd.read_csv('C:/Users/sahil/Downloads/pizzas.csv')
```

In [3]:
```
# Try reading the file with a specific encoding
try:
    df3 = pd.read_csv(('C:/Users/sahil/Downloads/pizza_types.csv'), encoding='latin
except UnicodeDecodeError:
    # Try another encoding if the first one fails
    df3 = pd.read_csv(('C:/Users/sahil/Downloads/pizza_types.csv'), encoding='iso-8
```

In [4]:
```
df3
```

Out[4]:

	pizza_type_id	name	category	ingredients
0	bbq_ckn	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, Red Peppers, Green Peppers,...
1	cali_ckn	The California Chicken Pizza	Chicken	Chicken, Artichoke, Spinach, Garlic, Jalapeno ...
2	ckn_alfredo	The Chicken Alfredo Pizza	Chicken	Chicken, Red Onions, Red Peppers, Mushrooms, A...
3	ckn_pesto	The Chicken Pesto Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Spinach, Garli...
4	southw_ckn	The Southwest Chicken Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Red Onions, Ja...
5	thai_ckn	The Thai Chicken Pizza	Chicken	Chicken, Pineapple, Tomatoes, Red Peppers, Tha...
6	big_meat	The Big Meat Pizza	Classic	Bacon, Pepperoni, Italian Sausage, Chorizo Sau...
7	classic_dlx	The Classic Deluxe Pizza	Classic	Pepperoni, Mushrooms, Red Onions, Red Peppers,...
8	hawaiian	The Hawaiian Pizza	Classic	Sliced Ham, Pineapple, Mozzarella Cheese
9	ital_cpcllo	The Italian Capocollo Pizza	Classic	Capocollo, Red Peppers, Tomatoes, Goat Cheese,...
10	napolitana	The Napolitana Pizza	Classic	Tomatoes, Anchovies, Green Olives, Red Onions,...
11	pep_msh_pep	The Pepperoni, Mushroom, and Peppers Pizza	Classic	Pepperoni, Mushrooms, Green Peppers
12	pepperoni	The Pepperoni Pizza	Classic	Mozzarella Cheese, Pepperoni
13	the_greek	The Greek Pizza	Classic	Kalamata Olives, Feta Cheese, Tomatoes, Garlic...
14	brie_carre	The Brie Carre Pizza	Supreme	Brie Carre Cheese, Prosciutto, Caramelized Oni...
15	calabrese	The Calabrese Pizza	Supreme	□Nduja Salami, Pancetta, Tomatoes, Red Onions,...
16	ital_supr	The Italian Supreme Pizza	Supreme	Calabrese Salami, Capocollo, Tomatoes, Red Oni...
17	peppr_salami	The Pepper Salami Pizza	Supreme	Genoa Salami, Capocollo, Pepperoni, Tomatoes, ...
18	prsc_argla	The Prosciutto and Arugula Pizza	Supreme	Prosciutto di San Daniele, Arugula, Mozzarella...
19	sicilian	The Sicilian Pizza	Supreme	Coarse Sicilian Salami, Tomatoes, Green Olives...

	pizza_type_id	name	category	ingredients
20	soppressata	The Soppressata Pizza	Supreme	Soppressata Salami, Fontina Cheese, Mozzarella...
21	spicy_ital	The Spicy Italian Pizza	Supreme	Capocollo, Tomatoes, Goat Cheese, Artichokes, ...
22	spinach_supr	The Spinach Supreme Pizza	Supreme	Spinach, Red Onions, Pepperoni, Tomatoes, Arti...
23	five_cheese	The Five Cheese Pizza	Veggie	Mozzarella Cheese, Provolone Cheese, Smoked Go...
24	four_cheese	The Four Cheese Pizza	Veggie	Ricotta Cheese, Gorgonzola Piccante Cheese, Mo...
25	green_garden	The Green Garden Pizza	Veggie	Spinach, Mushrooms, Tomatoes, Green Olives, Fe...
26	ital_veggie	The Italian Vegetables Pizza	Veggie	Eggplant, Artichokes, Tomatoes, Zucchini, Red ...
27	mediterraneo	The Mediterranean Pizza	Veggie	Spinach, Artichokes, Kalamata Olives, Sun-drie...
28	mexicana	The Mexicana Pizza	Veggie	Tomatoes, Red Peppers, Jalapeno Peppers, Red O...
29	spin_pesto	The Spinach Pesto Pizza	Veggie	Spinach, Artichokes, Tomatoes, Sun-dried Tomat...
30	spinach_fet	The Spinach and Feta Pizza	Veggie	Spinach, Mushrooms, Red Onions, Feta Cheese, G...
31	veggie_veg	The Vegetables + Vegetables Pizza	Veggie	Mushrooms, Tomatoes, Red Peppers, Green Pepper...

In [5]:
```
a = df1.merge(df4, on='pizza_id', how='left')
```

In [6]:
```
b = a.merge(df2, on='order_id', how='left')
```

In [7]:
```
merged = b.merge(df3, on='pizza_type_id', how='left')
```

In [8]:
```
merged.head(3)
```

Out[8]:

	order_details_id	order_id	pizza_id	quantity	pizza_type_id	size	price	date	ti
0	1	1	hawaiian_m	1	hawaiian	M	13.25	01-01-2023	11:38:
1	2	2	classic_dlx_m	1	classic_dlx	M	16.00	01-01-2023	11:57:
2	3	2	five_cheese_l	1	five_cheese	L	18.50	01-01-2023	11:57:

In [9]:
```
merged.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48655 entries, 0 to 48654
Data columns (total 12 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   order_details_id  48655 non-null  int64  
 1   order_id          48655 non-null  int64  
 2   pizza_id          48655 non-null  object 
 3   quantity          48655 non-null  int64  
 4   pizza_type_id     48655 non-null  object 
 5   size              48655 non-null  object 
 6   price             48655 non-null  float64
 7   date              48633 non-null  object 
 8   time              48633 non-null  object 
 9   name              48655 non-null  object 
 10  category          48655 non-null  object 
 11  ingredients       48655 non-null  object 
dtypes: float64(1), int64(3), object(8)
memory usage: 4.5+ MB
```

In [10]:
```
merged['revenue'] = merged['quantity'] * merged['price']
```

In [11]:
```
merged['revenue'].sum()
```

Out[11]:
```
818452.7
```

In [12]:
```
e = merged.groupby(['price','size'])['name'].unique()
e.reset_index().sort_values(by='price',ascending=False).head(5)
```

Out[12]:

	price	size	name
26	35.95	XXL	[The Greek Pizza]
25	25.50	XL	[The Greek Pizza]
24	23.65	S	[The Brie Carre Pizza]
23	21.00	L	[The Italian Vegetables Pizza]
22	20.75	L	[The Italian Supreme Pizza, The Thai Chicken P...

In [164...
```
merged.groupby('size').agg({'order_id':'count','quantity':'sum','revenue':'sum'})
```

Out[164...

size	order_id	quantity	revenue
L	18539	18970	375601.85
M	15394	15644	249531.00
S	14150	14416	178237.25
XL	544	552	14076.00
XXL	28	28	1006.60

In [63]:
```
merged.groupby('size')['revenue'].sum().sort_values(ascending=False).reset_index()
```

Out[63]:

	size	revenue
0	L	375601.85
1	M	249531.00
2	S	178237.25
3	XL	14076.00
4	XXL	1006.60

In [64]:
```
cate_total = merged.groupby('category')['quantity'].sum().sort_values(ascending=Fal
cate_total
```

Out[64]:

	category	quantity
0	Classic	14892
1	Supreme	11997
2	Veggie	11657
3	Chicken	11064

In [74]:
```
px.pie(data_frame=cate_total, names='category', values='quantity', width=550,title=
      color='category',color_discrete_sequence=["grey", "greenyellow", "green", "li
```

In [65]:
```
cate_total1 = merged.groupby('category')['revenue'].sum().sort_values(ascending=Fal
cate_total1
```

Out[65]:

	category	revenue
0	Classic	220112.1
1	Supreme	208379.0
2	Chicken	196142.0
3	Veggie	193819.6

In [78]:
```
px.pie(data_frame=cate_total1, names='category', values='revenue', width=550,title=
      color_discrete_sequence=['aqua','blue','lime','blueviolet'])
```

In [19]:
```
pizza_total = merged.groupby('name')['quantity'].sum().reset_index()
pizza_total.sort_values(by='quantity',ascending=False)
```

Out[19]:

	name	quantity
7	The Classic Deluxe Pizza	2454
0	The Barbecue Chicken Pizza	2434
12	The Hawaiian Pizza	2423
20	The Pepperoni Pizza	2418
30	The Thai Chicken Pizza	2374
4	The California Chicken Pizza	2374
23	The Sicilian Pizza	1938
26	The Spicy Italian Pizza	1927
25	The Southwest Chicken Pizza	1919
1	The Big Meat Pizza	1914
9	The Four Cheese Pizza	1904
14	The Italian Supreme Pizza	1885
31	The Vegetables + Vegetables Pizza	1528
17	The Mexicana Pizza	1484
18	The Napolitana Pizza	1465
22	The Prosciutto and Arugula Pizza	1458
19	The Pepper Salami Pizza	1449
29	The Spinach and Feta Pizza	1446
13	The Italian Capocollo Pizza	1438
10	The Greek Pizza	1421
8	The Five Cheese Pizza	1409
21	The Pepperoni, Mushroom, and Peppers Pizza	1359
11	The Green Garden Pizza	998
5	The Chicken Alfredo Pizza	988
15	The Italian Vegetables Pizza	982
6	The Chicken Pesto Pizza	975
27	The Spinach Pesto Pizza	970
24	The Soppressata Pizza	962
28	The Spinach Supreme Pizza	950
3	The Calabrese Pizza	938

	name	quantity
16	The Mediterranean Pizza	936
2	The Brie Carre Pizza	490

In [20]:
```
pizza_total1 = pizza_total.sort_values(by='quantity',ascending=False).head(16)
pizza_total1
```

Out[20]:

	name	quantity
7	The Classic Deluxe Pizza	2454
0	The Barbecue Chicken Pizza	2434
12	The Hawaiian Pizza	2423
20	The Pepperoni Pizza	2418
30	The Thai Chicken Pizza	2374
4	The California Chicken Pizza	2374
23	The Sicilian Pizza	1938
26	The Spicy Italian Pizza	1927
25	The Southwest Chicken Pizza	1919
1	The Big Meat Pizza	1914
9	The Four Cheese Pizza	1904
14	The Italian Supreme Pizza	1885
31	The Vegetables + Vegetables Pizza	1528
17	The Mexicana Pizza	1484
18	The Napolitana Pizza	1465
22	The Prosciutto and Arugula Pizza	1458

In [21]:
```
d = {'name': "Other", 'quantity': 17711}
pizza_total1 = pizza_total1._append(d, ignore_index=True)
```

In [22]:
```
px.pie(data_frame=pizza_total1, names='name', values='quantity',color_discrete_sequ
       hole=0.6
      ).update_layout(margin=dict(l=150, r=10, t=50, b=10))
```

In [79]:
```
pizza_total2 = pizza_total
px.bar(pizza_total2,'name','quantity', color='quantity' ,color_continuous_scale='Vi
      ).update_layout(xaxis_showticklabels=False)
```

In [24]:
```
f = merged.groupby('name')['revenue'].sum().reset_index()
```

In [25]:
```
px.bar(f,'name','revenue',color='revenue',color_continuous_scale='matter',title='Re
      ).update_layout(xaxis_showticklabels=False)
```

In [153…

```
px.pie(merged,names='name',values='revenue',hole=0.6,title='Pizzas Contribution to
    margin=dict(l=20, r=20, t=20, b=20),template='ggplot2',title_x=0.01,title_y=0.9
```

In [230…

```
px.treemap(merged, path=[px.Constant("Pizzas"), 'category','size','name'],values='r
          ).update_layout(margin=dict(l=20, r=20, t=20, b=20),template='ggplot2')
```

In [28]:
```
merged
```

Out[28]:

	order_details_id	order_id	pizza_id	quantity	pizza_type_id	size	price	date
0	1	1	hawaiian_m	1	hawaiian	M	13.25	01-01-2023
1	2	2	classic_dlx_m	1	classic_dlx	M	16.00	01-01-2023
2	3	2	five_cheese_l	1	five_cheese	L	18.50	01-01-2023
3	4	2	ital_supr_l	1	ital_supr	L	20.75	01-01-2023
4	5	2	mexicana_m	1	mexicana	M	16.00	01-01-2023
...	...	...	...	...	...	...	...	...
48650	48616	21348	ckn_alfredo_m	1	ckn_alfredo	M	16.75	31-12-2023
48651	48617	21348	four_cheese_l	1	four_cheese	L	17.95	31-12-2023
48652	48618	21348	napolitana_s	1	napolitana	S	12.00	31-12-2023
48653	48619	21349	mexicana_l	1	mexicana	L	20.25	31-12-2023

	order_details_id	order_id	pizza_id	quantity	pizza_type_id	size	price	date
48654	48620	21350	bbq_ckn_s	1	bbq_ckn	S	12.75	31-12-2023

48655 rows × 13 columns

In [29]:
```
g = pd.to_datetime(merged['time'],format='%H:%M:%S').reset_index()
g
```

Out[29]:

	index	time
0	0	1900-01-01 11:38:36
1	1	1900-01-01 11:57:40
2	2	1900-01-01 11:57:40
3	3	1900-01-01 11:57:40
4	4	1900-01-01 11:57:40
...	...	...
48650	48650	1900-01-01 21:23:10
48651	48651	1900-01-01 21:23:10
48652	48652	1900-01-01 21:23:10
48653	48653	1900-01-01 22:09:54
48654	48654	1900-01-01 23:02:05

48655 rows × 2 columns

In [30]:
```
g1 = g['time'].dt.hour
g1 = pd.DataFrame(g1)
g1.rename(columns={'time':'hour'},inplace=True)
g1
```

Out[30]:

	hour
0	11.0
1	11.0
2	11.0
3	11.0
4	11.0
...	...
48650	21.0
48651	21.0
48652	21.0
48653	22.0
48654	23.0

48655 rows × 1 columns

In [31]:
```
h = pd.concat([merged,g1],axis=1)
```

In [32]:
```
h['hour'].value_counts()
```

Out[32]:
```
hour
12.0    6545
13.0    6214
18.0    5359
17.0    5143
19.0    4350
16.0    4185
14.0    3521
20.0    3487
15.0    3170
11.0    2672
21.0    2528
22.0    1370
23.0      68
10.0      17
9.0        4
Name: count, dtype: int64
```

In [125…
```
i = h.groupby('hour')['quantity'].sum().sort_values(ascending=False).reset_index()
i
```

Out[125...

	hour	quantity
0	12.0	6778
1	13.0	6425
2	18.0	5417
3	17.0	5211
4	19.0	4406
5	16.0	4239
6	14.0	3613
7	20.0	3534
8	15.0	3216
9	11.0	2728
10	21.0	2545
11	22.0	1386
12	23.0	68
13	10.0	18
14	9.0	4

In [35]:
```
print(h.loc[(h['order_details_id'] >=0) & (h['order_details_id'] <=4155), 'quantity
print(h.loc[(h['order_details_id'] >=4156) & (h['order_details_id'] <=8048), 'quant
print(h.loc[(h['order_details_id'] >=8049) & (h['order_details_id'] <=12234), 'quan

print(h.loc[(h['order_details_id'] >=12235) & (h['order_details_id'] <=16301), 'qua
print(h.loc[(h['order_details_id'] >=16302) & (h['order_details_id'] <=20540), 'qua
print(h.loc[(h['order_details_id'] >=20541) & (h['order_details_id'] <=24565), 'qua

print(h.loc[(h['order_details_id'] >=24566) & (h['order_details_id'] <=28878), 'qua
print(h.loc[(h['order_details_id'] >=28879) & (h['order_details_id'] <=32972), 'qua
print(h.loc[(h['order_details_id'] >=32973) & (h['order_details_id'] <=36791), 'qua

print(h.loc[(h['order_details_id'] >=36792) & (h['order_details_id'] <=40588), 'qua
print(h.loc[(h['order_details_id'] >=40589) & (h['order_details_id'] <=44773), 'qua
print(h.loc[(h['order_details_id'] >=44774) & (h['order_details_id'] <=48655), 'qua
```

```
4231
3963
4261
4151
4328
4107
4417
4168
3890
3883
4266
3945
```

In [36]:
```
print(h.loc[(h['order_details_id'] >=0) & (h['order_details_id'] <=12234), 'quantit
print(h.loc[(h['order_details_id'] >=12235) & (h['order_details_id'] <=24565), 'qua
print(h.loc[(h['order_details_id'] >=24566) & (h['order_details_id'] <=36791), 'qua
print(h.loc[(h['order_details_id'] >=36792) & (h['order_details_id'] <=48655), 'qua
```

```
12455
12586
12475
12094
```

In [37]:
```
j = h.loc[(h['order_details_id'] >=0) & (h['order_details_id'] <=4155)]
j1=j.groupby('hour')['quantity'].sum().reset_index()
```

In [38]:
```
j2 = h.loc[(h['order_details_id'] >=4156) & (h['order_details_id'] <=8048)]
j3 = j2.groupby('hour')['quantity'].sum().reset_index()
```

In [39]:
```
j4 = h.loc[(h['order_details_id'] >=8049) & (h['order_details_id'] <=12234)]
j5 = j4.groupby('hour')['quantity'].sum().reset_index()
```

In [40]:
```
fig = make_subplots(rows=3, cols=1, subplot_titles=("Jan", "Feb", "Mar"))

fig.add_trace(
    go.Scatter(x=j1['hour'], y=j1['quantity'], mode='lines'),
    row=1, col=1
)

fig.add_trace(
    go.Scatter(x=j3['hour'], y=j3['quantity'], mode='lines'),
    row=2, col=1
)

fig.add_trace(
    go.Scatter(x=j5['hour'], y=j5['quantity'], mode='lines',line_shape='spline'),
    row=3, col=1
)

fig.update_layout(height=900, width=1000, title_text="Time-Series Analysis",)

fig.show()
```

In [42]:
```
fig = go.Figure()
```

```
fig.add_trace(
    go.Scatter(x=j1['hour'], y=j1['quantity'], mode='lines+markers',name='Jan',mark
)

fig.add_trace(
    go.Scatter(x=j3['hour'], y=j3['quantity'], mode='lines+markers',name='Feb',mark
)

fig.add_trace(
    go.Scatter(x=j5['hour'], y=j5['quantity'], mode='lines+markers',name='Mar',mark
)

fig.update_xaxes(nticks=25)

fig.update_layout(height=600, width=1160, title_text="Time-Series Analysis",
                  xaxis_title='Hour',yaxis_title='Quantity',showlegend=True,
    legend=dict(
        title="Months",
        x=0.8,
        y=1,
        traceorder="normal"))

fig.show()
```

In [43]:
```
j6 = h.loc[(h['order_details_id'] >=0) & (h['order_details_id'] <=12234)]
j7 = h.loc[(h['order_details_id'] >=12235) & (h['order_details_id'] <=24565)]
j8 = h.loc[(h['order_details_id'] >=24566) & (h['order_details_id'] <=36791)]
j9 = h.loc[(h['order_details_id'] >=36792) & (h['order_details_id'] <=48655)]
```

In [44]:
```
j10 = j6.groupby('hour')['quantity'].sum().reset_index()
j11 = j7.groupby('hour')['quantity'].sum().reset_index()
j12 = j8.groupby('hour')['quantity'].sum().reset_index()
j13 = j9.groupby('hour')['quantity'].sum().reset_index()
```

In [45]:
```
fig = go.Figure()

fig.add_trace(go.Scatter(x=j10['hour'],y=j10['quantity'],mode='lines+markers',name=
fig.add_trace(go.Scatter(x=j11['hour'],y=j11['quantity'],mode='lines+markers',name=
fig.add_trace(go.Scatter(x=j12['hour'],y=j12['quantity'],mode='lines+markers',name=
fig.add_trace(go.Scatter(x=j13['hour'],y=j13['quantity'],mode='lines+markers',name=

fig.update_xaxes(nticks=25)
fig.update_yaxes(nticks=20)
```

```
fig.update_layout(title_text='Quaterly Time-Series Analysis',height=700,width=1160,
                  legend=dict(title='Quarters',x=0.8,y=1))
fig.show()
```

In [46]:
```
k = h.groupby('price')['revenue'].sum().reset_index()
```

In [47]:
```
px.pie(k,'price','revenue',title='Contribution to Revenue by Pizza Price').update_t
                                                                           ).update_
    margin=dict(l=20, r=100, t=30, b=20),legend=dict(x=0.76,y=1),
    font=dict(
        family="Balto",
        size=15))
```

In [136…

```
px.scatter(k,'price','revenue',title='Price Revenue Relation')
```

In [137…

```
px.scatter(h,'price','quantity',color='quantity',color_continuous_scale='hsv',title
```

In [50]:
```
k1 = h.groupby('price')['quantity'].sum().reset_index()
```

In [138…
```
px.scatter(k1,'price','quantity',title='Total Quantity sold by Price')
```

In [55]:
```
df1_style = k.sort_values(by='revenue',ascending=False).head(5).style.set_table_att

df2_style = k1.sort_values(by='quantity',ascending=False).head(5).style.set_table_a
                                                ).set_caption('top 5 highest quantity sold

display_html(df1_style._repr_html_() + df2_style._repr_html_(), raw=True)
```

top 5 highest revenue

	price	revenue
20	20.750000	184654.250000
14	16.750000	73448.750000
11	16.000000	72384.000000
3	12.000000	68964.000000
13	16.500000	67864.500000

top 5 highest quantity sold

	price	quantity
20	20.750000	8899
3	12.000000	5747
11	16.000000	4524
14	16.750000	4385
13	16.500000	4113

In [56]:
```
h['date'] = pd.to_datetime(h['date'],format='%d-%m-%Y')
h['month'] = h['date'].dt.strftime('%b-%Y')
print(h['month'])
```

```
0        Jan-2023
1        Jan-2023
2        Jan-2023
3        Jan-2023
4        Jan-2023
           ...
48650    Dec-2023
48651    Dec-2023
48652    Dec-2023
48653    Dec-2023
48654    Dec-2023
Name: month, Length: 48655, dtype: object
```

In [118…
```
l = h.groupby('month')['revenue'].sum().reset_index()
```

In [148…
```
month_order = ["Jan-2023", "Feb-2023", "Mar-2023", "Apr-2023", "May-2023", "Jun-202
               , "Jul-2023", "Aug-2023", "Sep-2023", "Oct-2023", "Nov-2023", "Dec-2
month_type = pd.CategoricalDtype(categories=month_order, ordered=True)
l['month'] = l['month'].astype(month_type)
l = l.sort_values('month').reset_index(drop=True)
```

In [202…
```
px.line(l,'month','revenue',markers='circle',color_discrete_sequence=['limegreen'],
```

In [60]:
```
h['size'].value_counts()
```

Out[60]:
```
size
L      18539
M      15394
S      14150
XL       544
XXL       28
Name: count, dtype: int64
```

In [121…
```
m = h.groupby(['size','name'])['quantity'].sum().reset_index()
```

In [122…
```
m1 = m.sort_values(by=['size','quantity'],ascending=[True,False]).groupby('size').h
m1
```

Out[122…

	size	name	quantity
28	L	The Thai Chicken Pizza	1411
6	L	The Five Cheese Pizza	1409
7	L	The Four Cheese Pizza	1318
35	M	The Classic Deluxe Pizza	1181
30	M	The Barbecue Chicken Pizza	957
32	M	The California Chicken Pizza	945
60	S	The Big Meat Pizza	1914
69	S	The Hawaiian Pizza	1021
66	S	The Classic Deluxe Pizza	799
89	XL	The Greek Pizza	552
90	XXL	The Greek Pizza	28

In [203…
```
px.bar(m1,'size','quantity',color='name',barmode='group',title='Top 3 most sold Piz
```

In [186…
```
m2 = h.groupby(['category','name'])[['revenue','quantity']].sum().reset_index()
```

In [200…
```
m3 = m2.sort_values(by=['category','revenue'],ascending=[True,False]).groupby('cate
m3
```

Out[200…

	category	name	revenue	quantity
5	Chicken	The Thai Chicken Pizza	43484.50	2374
0	Chicken	The Barbecue Chicken Pizza	42797.50	2434
1	Chicken	The California Chicken Pizza	41464.50	2374
7	Classic	The Classic Deluxe Pizza	38201.00	2454
9	Classic	The Hawaiian Pizza	32283.75	2423
12	Classic	The Pepperoni Pizza	30161.75	2418
21	Supreme	The Spicy Italian Pizza	34889.25	1927
16	Supreme	The Italian Supreme Pizza	33493.25	1885
19	Supreme	The Sicilian Pizza	30940.50	1938
24	Veggie	The Four Cheese Pizza	32301.60	1904
28	Veggie	The Mexicana Pizza	26780.75	1484
23	Veggie	The Five Cheese Pizza	26066.50	1409

In [204…

```
px.bar(m3,'category','revenue',color='name',barmode='group',hover_data='quantity',t
```

In [ ]: