Online Retail Analysis

Familiarizing with data

SQL | Saved to variable `df_1`

```sql
SELECT * FROM 'online_retail.csv'
LIMIT 10
```

	InvoiceNo object	StockCode object	Description object	Quantity int64	InvoiceDate datet...	UnitPrice float64	CustomerID float64	Country object
	536365 70% 536366 20% 536367 10%	85123A 10% 71053 10% 8 others 80%	WHITE HAN... 10% WHITE MET... 10% 8 others 80%	2 - 32	2010-12-01 08:26:...	1.69 - 7.65	13047.0 - 17850.0	United King... 100%
0	536365	85123A	WHITE HANGING HEART T-LIGHT...	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT...	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATE...	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE...	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
5	536365	22752	SET 7 BABUSHKA NESTING BOXES	2	2010-12-01 08:26:00	7.65	17850.0	United Kingdom
6	536365	21730	GLASS STAR FROSTED T-LIGH...	6	2010-12-01 08:26:00	4.25	17850.0	United Kingdom
7	536366	22633	HAND WARMER UNION JACK	6	2010-12-01 08:28:00	1.85	17850.0	United Kingdom
8	536366	22632	HAND WARMER RED POLKA DOT	6	2010-12-01 08:28:00	1.85	17850.0	United Kingdom
9	536367	84879	ASSORTED COLOUR BIRD...	32	2010-12-01 08:34:00	1.69	13047.0	United Kingdom

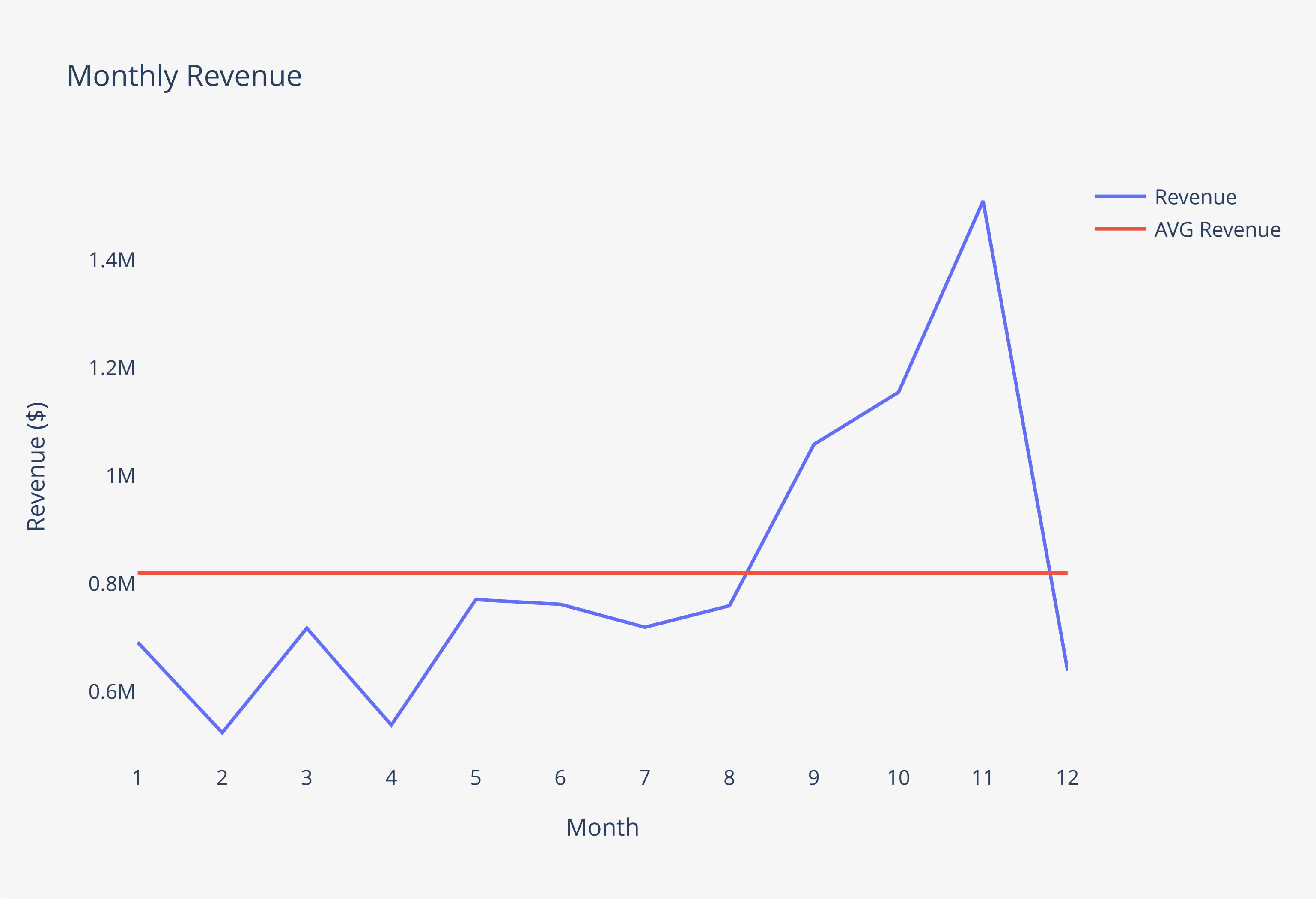
Monthly Revenue Analysis

How did the last (latest) year perform in terms of Revenue? Compare monthly revenue to yearly average

SQL | Saved to variable `revenue_df`

```sql
WITH year_lookup AS (
    SELECT MAX(EXTRACT('year' FROM InvoiceDate)) as max_year FROM 'online_retail.csv'
)
,base_table AS (
SELECT
    EXTRACT('year' FROM InvoiceDate) as invoice_year
    ,EXTRACT('month' FROM InvoiceDate) as invoice_month
    ,ROUND(SUM(Quantity * UnitPrice),2) as revenue
FROM 'online_retail.csv'
WHERE 1=1
AND invoice_year = (SELECT max_year FROM year_lookup)
AND UnitPrice >0
AND Quantity >0
GROUP BY 1,2
)
SELECT
invoice_year
,invoice_month
,ROUND(AVG(revenue) OVER (PARTITION BY invoice_year),2) as avg_revenue
,revenue
FROM base_table
```

```python
# Start with loading all necessary libraries
import numpy as np
import pandas as pd

# Viz libraries
import plotly.express as px
import plotly.graph_objects as go

colors = {'background':'#F6F6F6','text':'#333333','colorscale':'Purples'}

trace0 = go.Scatter(
    x=revenue_df['invoice_month']
    ,y=revenue_df['revenue']
    ,mode = 'lines'
    ,name = 'Revenue'
    ,hovertemplate='$%{y:.2s}'
    )

trace1 = go.Scatter(
    x=revenue_df['invoice_month']
    ,y=revenue_df['avg_revenue']
    ,mode = 'lines'
    ,name = 'AVG Revenue'
    ,hovertemplate='$%{y:.2s}'
    )

data = [trace0,trace1]

layout = go.Layout(
    title = 'Monthly Revenue'
    ,plot_bgcolor = colors['background']
    ,paper_bgcolor = colors['background']
    ,xaxis=dict(
        tickmode='array'
        ,tickvals=revenue_df['invoice_month']
        ,showgrid = False
        ,title = 'Month'
    )
    ,yaxis = dict(
        showgrid = False
        ,title = 'Revenue ($)'
    )
)

line_go = go.Figure(data,layout)
line_go.update_layout(
    hovermode='x unified'
    )
line_go.show()
```



Conclusion

Company achieved its highest revenue in November, generating $1.5M, which is almost double the yearly average of $820K. The biggest revenues tend to occur between September and November, indicating a seasonal trend.

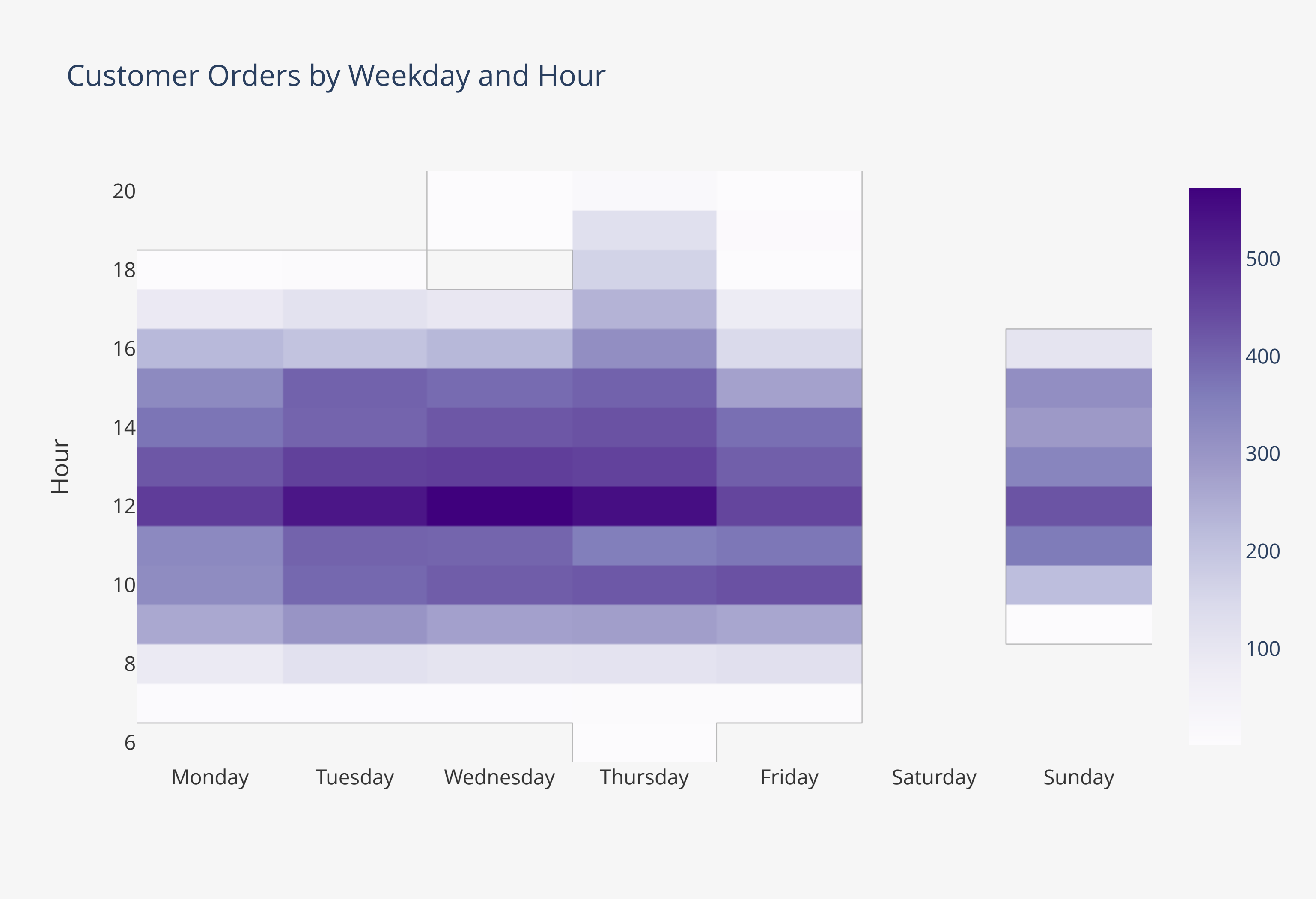
Orders Day/Time Analysis

When do customers order the most (Day & Time of the day)?

SQL Saved to variable `order_time_df`

```sql
WITH year_lookup AS (
    SELECT MAX(EXTRACT('year' FROM InvoiceDate)) as max_year FROM 'online_retail.csv'
)
,base_table AS (
SELECT
CASE EXTRACT('weekday' from InvoiceDate)
    WHEN 0 THEN 'Sunday'
    WHEN 1 THEN 'Monday'
    WHEN 2 THEN 'Tuesday'
    WHEN 3 THEN 'Wednesday'
    WHEN 4 THEN 'Thursday'
    WHEN 5 THEN 'Friday'
    WHEN 6 THEN 'Saturday'
END AS order_weekday
,EXTRACT('hour' from InvoiceDate) AS order_hour
,InvoiceNo
,Quantity
,Quantity * UnitPrice AS revenue
FROM 'online_retail.csv'
WHERE 1=1
AND EXTRACT('year' FROM InvoiceDate) = (SELECT max_year FROM year_lookup)
AND UnitPrice > 0
AND Quantity >0
)
SELECT
order_weekday
,order_hour
,COUNT(DISTINCT InvoiceNo) as orders
,SUM(Quantity) as units
,ROUND(SUM(revenue),2) as revenue
FROM base_table
GROUP by 1,2
```

```python
# Start with loading all necessary libraries
import numpy as np
import pandas as pd

# Viz libraries
import plotly.express as px
import plotly.graph_objects as go
colors = {'background':'#F6F6F6','text':'#333333','colorscale':'Purples'}

data = [
    go.Heatmap(
        colorscale = colors['colorscale']
        ,x=order_time_df['order_weekday']
        ,y=order_time_df['order_hour']
        ,z=order_time_df['orders'].values.tolist()
        ,hovertemplate="Day: %{x}<br>Hour: %{y:.2s}<br># of Orders: %{z} <extra></extra>"
    ),
]

layout = go.Layout(
    # template = 'plotly_dark'
    title='Customer Orders by Weekday and Hour'
    ,xaxis=dict(
        categoryorder='array'
        ,categoryarray=['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
        ,showgrid = False
        ,color = colors['text']
    )
    ,yaxis=dict(
        title = 'Hour'
        ,categoryorder='array'
        ,categoryarray=list(range(24))
        ,showgrid = False
        ,color = colors['text']
    )
    ,plot_bgcolor = colors['background']
    ,paper_bgcolor = colors['background']
)

hm_go = go.Figure(data, layout)
hm_go.show()
```



Conclusion

Most customer purchases occur from Tuesday to Thursday throughout the week. Additionally, the peak time for customer orders is between 12:00 PM to 2:00 PM, suggesting that customers tend to place their orders during lunchtime. This information can be useful for businesses to optimize their marketing strategies and adjust their staffing and inventory levels to accommodate peak sales times.