Electricity Consumption Analysis

In [3]:

```
# Importing all the necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [4]:

```
# Loading the CSV file into the notebook
df = pd.read_csv('powerconsumption.csv', usecols = ['Datetime', 'PowerConsumption_Zone1'
df
```

Out[4]:

	Datetime	PowerConsumption_Zone1	PowerConsumption_Zone2	PowerConsumption_Zone3
0	2017-01-01 00:00:00	34055.69620	16128.87538	20240.96386
1	2017-01-01 00:10:00	29814.68354	19375.07599	20131.08434
2	2017-01-01 00:20:00	29128.10127	19006.68693	19668.43373
3	2017-01-01 00:30:00	28228.86076	18361.09422	18899.27711
4	2017-01-01 00:40:00	27335.69620	17872.34043	18442.40964
...	...	...	...	...
52411	2017-12-30 23:10:00	31160.45627	26857.31820	14780.31212

	Datetime	PowerConsumption_Zone1	PowerConsumption_Zone2	PowerConsumption_Zone3
52412	2017-12-30 23:20:00	30430.41825	26124.57809	14428.81152
52413	2017-12-30 23:30:00	29590.87452	25277.69254	13806.48259
52414	2017-12-30 23:40:00	28958.17490	24692.23688	13512.60504
52415	2017-12-30 23:50:00	28349.80989	24055.23167	13345.49820

52416 rows × 4 columns

In [5]:

```
# Checking the columns for the dataframe
df.columns
```

Out[5]:

```
Index(['Datetime', 'PowerConsumption_Zone1', 'PowerConsumption_Zone2',
       'PowerConsumption_Zone3'],
      dtype='object')
```

In [6]:

```
# Checking the information of the dataframe
df.info
```

Out[6]:

```
<bound method DataFrame.info of                    Datetime  PowerConsumption_Zone1  Powe
rConsumption_Zone2  \
0      2017-01-01 00:00:00             34055.69620             16128.87538
1      2017-01-01 00:10:00             29814.68354             19375.07599
2      2017-01-01 00:20:00             29128.10127             19006.68693
3      2017-01-01 00:30:00             28228.86076             18361.09422
4      2017-01-01 00:40:00             27335.69620             17872.34043
...                    ...                     ...                     ...
52411  2017-12-30 23:10:00             31160.45627             26857.31820
52412  2017-12-30 23:20:00             30430.41825             26124.57809
52413  2017-12-30 23:30:00             29590.87452             25277.69254
52414  2017-12-30 23:40:00             28958.17490             24692.23688
52415  2017-12-30 23:50:00             28349.80989             24055.23167

       PowerConsumption_Zone3
0                 20240.96386
1                 20131.08434
2                 19668.43373
3                 18899.27711
4                 18442.40964
...                       ...
52411             14780.31212
52412             14428.81152
52413             13806.48259
52414             13512.60504
52415             13345.49820

[52416 rows x 4 columns]>
```

In [7]:

```
# Converting the Datetime column as Datetime datatype
df['Datetime'] = pd.to_datetime(df['Datetime'])
df['Datetime']
```

```
Out[7]:
0       2017-01-01 00:00:00
1       2017-01-01 00:10:00
2       2017-01-01 00:20:00
3       2017-01-01 00:30:00
4       2017-01-01 00:40:00
                ...
52411   2017-12-30 23:10:00
52412   2017-12-30 23:20:00
52413   2017-12-30 23:30:00
52414   2017-12-30 23:40:00
52415   2017-12-30 23:50:00
Name: Datetime, Length: 52416, dtype: datetime64[ns]
```

```
In [8]:
# Creating a Total Consumption column that will represent the sum of all three Power Con
df['Total_Consumption'] = df['PowerConsumption_Zone1'] + df['PowerConsumption_Zone2'] +
df['Total_Consumption']
```

```
Out[8]:
0        70425.53544
1        69320.84387
2        67803.22193
3        65489.23209
4        63650.44627
            ...
52411    72798.08659
52412    70983.80786
52413    68675.04965
52414    67163.01682
52415    65750.53976
Name: Total_Consumption, Length: 52416, dtype: float64
```

```
In [9]:
# Setting Datetime as index, and resampling the data
(df
 .set_index('Datetime')
 .resample('h')
 ['PowerConsumption_Zone1', 'PowerConsumption_Zone2', 'PowerConsumption_Zone3']
 .mean()
 .loc['2017-01']
)
```

Out[9]:

Datetime	PowerConsumption_Zone1	PowerConsumption_Zone2	PowerConsumption_Zone3
2017-01-01 00:00:00	29197.974683	18026.747720	19252.048193
2017-01-01 01:00:00	24657.215190	16078.419453	17042.891567
2017-01-01 02:00:00	22083.037973	14330.699088	15676.144578
2017-01-01 03:00:00	20811.139240	13219.452887	14883.855422

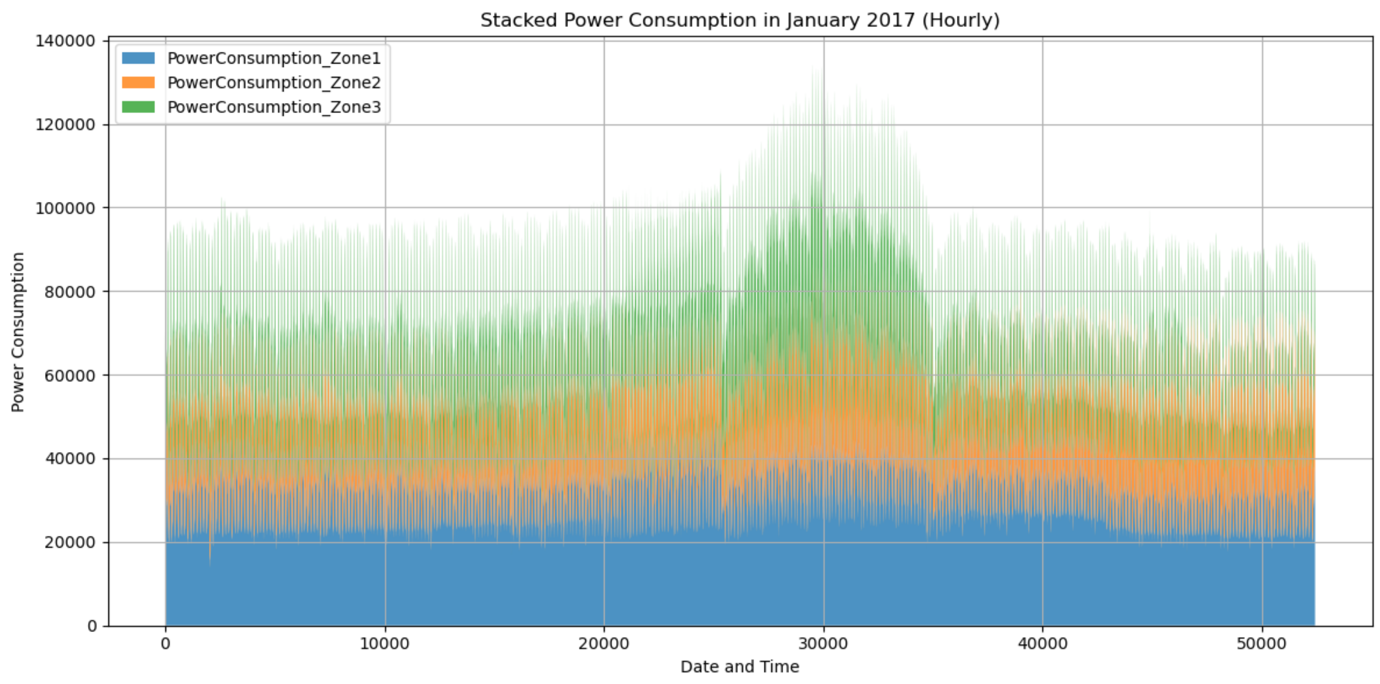
	PowerConsumption_Zone1	PowerConsumption_Zone2	PowerConsumption_Zone3
Datetime			
2017-01-01 04:00:00	20475.949367	12921.580547	14317.108433
...	...	...	...
2017-01-31 19:00:00	42843.544303	25438.297875	25731.084337
2017-01-31 20:00:00	43023.797470	25429.787233	26003.855422
2017-01-31 21:00:00	41560.506330	25259.574468	25527.710845
2017-01-31 22:00:00	38052.658228	23637.689968	23936.385542
2017-01-31 23:00:00	33158.481010	20456.534953	20732.530120

744 rows × 3 columns

In [10]:

```
# Visualizing the Power Consumption
plt.figure(figsize = (12, 6))
plt.stackplot(df.index,
              df['PowerConsumption_Zone1'],
              df['PowerConsumption_Zone2'],
              df['PowerConsumption_Zone3'],
              labels = ['PowerConsumption_Zone1', 'PowerConsumption_Zone2', 'PowerConsum
              alpha = 0.8)

plt.xlabel('Date and Time')
plt.ylabel('Power Consumption')
plt.title('Stacked Power Consumption in January 2017 (Hourly)')
plt.legend(loc = 'upper left')
plt.grid(True)
plt.tight_layout()
plt.show()
```



In [11]:

```
df['DayOfWeek'] = df['Datetime'].dt.day_name()
df['HourOfDay'] = df['Datetime'].dt.hour
```

In [12]:

```
# Creating a pivot table
pivot_table = pd.pivot_table(df, values = 'Total_Consumption', index = 'DayOfWeek', colu
pivot_table
```

Out[12]:

HourOfDay	0	1	2	3	4	5	
DayOfWeek							
Friday	68959.251459	62533.960861	59152.040758	56825.429786	55296.351259	52178.788896	511
Monday	67817.926234	61755.422717	58647.360270	56399.125350	54823.472531	51680.112788	508
Saturday	68991.576226	62628.416936	59321.825090	56745.339022	54816.778024	51275.831269	491
Sunday	69728.136464	63113.302575	59243.071267	56207.553656	53896.627602	49677.139543	460
Thursday	68293.632675	62187.544469	59063.328906	56958.133058	55465.004964	52336.590999	514
Tuesday	67935.356547	61868.544847	58801.200586	56642.907327	55088.146346	52165.725002	513
Wednesday	68239.842528	62022.724317	58798.761412	56626.703787	55114.497390	52157.866552	514

7 rows × 24 columns

In [13]:

```
# Reorder the rows to have the days of the week in order
days_order = ['Monday','Tuesday','Wednesday','Thursday','Friday','Saturday','Sunday']
pivot_table = pivot_table.reindex(days_order)
pivot_table
```

Out[13]:

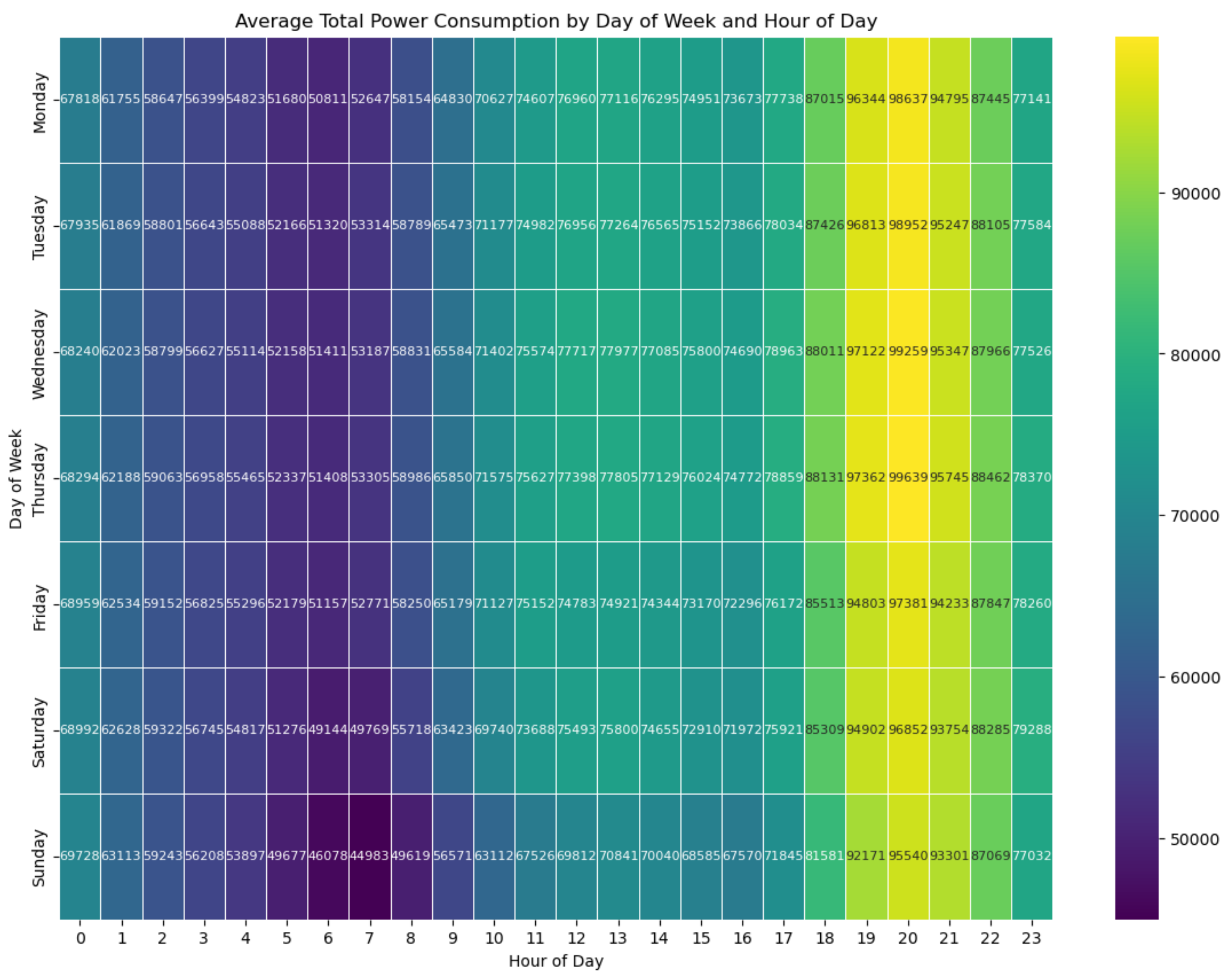
HourOfDay	0	1	2	3	4	5	
DayOfWeek							
Monday	67817.926234	61755.422717	58647.360270	56399.125350	54823.472531	51680.112788	508
Tuesday	67935.356547	61868.544847	58801.200586	56642.907327	55088.146346	52165.725002	513
Wednesday	68239.842528	62022.724317	58798.761412	56626.703787	55114.497390	52157.866552	514
Thursday	68293.632675	62187.544469	59063.328906	56958.133058	55465.004964	52336.590999	514
Friday	68959.251459	62533.960861	59152.040758	56825.429786	55296.351259	52178.788896	511
Saturday	68991.576226	62628.416936	59321.825090	56745.339022	54816.778024	51275.831269	491
Sunday	69728.136464	63113.302575	59243.071267	56207.553656	53896.627602	49677.139543	460

7 rows × 24 columns

In [14]:

```
# Create a heatmap with adjusted font size for annotations
plt.figure(figsize = (14, 10))
sns.heatmap(pivot_table, cmap = 'viridis', annot = True, fmt = ".0f", linewidths = .5, a
plt.title('Average Total Power Consumption by Day of Week and Hour of Day')
```

```python
plt.xlabel('Hour of Day')
plt.ylabel('Day of Week')
plt.show()
```



Day of Week	0	1	2	3	4	5	6	7	8	9	10	11
Monday	67818	61755	58647	56399	54823	51680	50811	52647	58154	64830	70627	74607
Tuesday	67935	61869	58801	56643	55088	52166	51320	53314	58789	65473	71177	74982
Wednesday	68240	62023	58799	56627	55114	52158	51411	53187	58831	65584	71402	75574
Thursday	68294	62188	59063	56958	55465	52337	51408	53305	58986	65850	71575	75627
Friday	68959	62534	59152	56825	55296	52179	51157	52771	58250	65179	71127	75152
Saturday	68992	62628	59322	56745	54817	51276	49144	49769	55718	63423	69740	73688
Sunday	69728	63113	59243	56208	53897	49677	46078	44983	49619	56571	63112	67526

Day of Week	12	13	14	15	16	17	18	19	20	21	22	23
Monday	76960	77116	76295	74951	73673	77738	87015	96344	98637	94795	87445	77141
Tuesday	76956	77264	76565	75152	73866	78034	87426	96813	98952	95247	88105	77584
Wednesday	77717	77977	77085	75800	74690	78963	88011	97122	99259	95347	87966	77526
Thursday	77398	77805	77129	76024	74772	78859	88131	97362	99639	95745	88462	78370
Friday	74783	74921	74344	73170	72296	76172	85513	94803	97381	94233	87847	78260
Saturday	75493	75800	74655	72910	71972	75921	85309	94902	96852	93754	88285	79288
Sunday	69812	70841	70040	68585	67570	71845	81581	92171	95540	93301	87069	77032

Conclusion

> Predictable Patterns: Power consumption exhibits clear daily and weekly patterns, with higher usage during weekdays' daytime and evening hours, and lower usage during nights and weekends.

> Time of Day Influence: The hour of the day significantly influences total consumption, though it's not the sole factor explaining all variations.

> Other Influencers: Other unanalyzed factors (like day of week specifics, weather, or seasonal changes) are also crucial drivers of consumption levels.