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', 'PowerConsumption_Zone2', 'PowerConsumption_Zone3'])
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
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  PowerConsumption_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 Consumption columns
df['Total_Consumption'] = df['PowerConsumption_Zone1'] + df['PowerConsumption_Zone2'] + df['PowerConsumption_Zone3']
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']
)
```

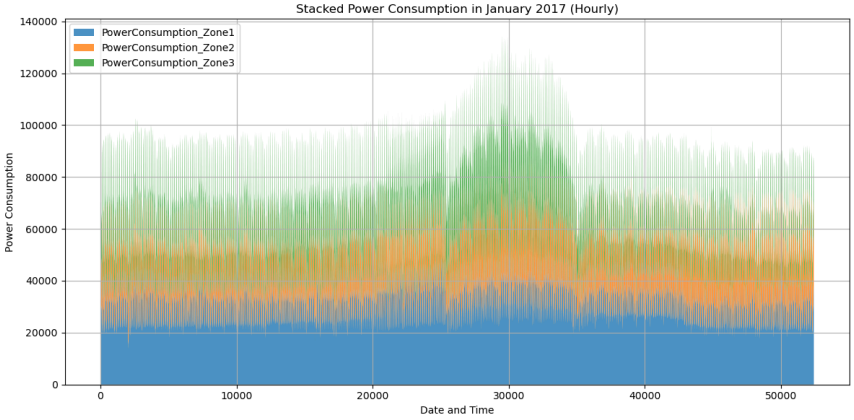
Out[9]:

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
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', 'PowerConsumption_Zone3'],
              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', columns = 'HourOfDay', aggfunc = "mean")
pivot_table
```

Out[12]:

DayOfWeek	HourOfDay 0	1	2	3	4	5	6	7	8	9	...	14	15	16	17	18	19	20	21	
Friday	68959.251459	62533.960861	59152.040758	56825.429786	55296.351259	52178.788896	51156.640309	52771.135698	58250.443622	65179.083324	...	74344.497887	73169.524043	72295.790052	76172.341277	85512.967870	94802.624915	97380.905756	94232.932202	8784
Monday	67817.926234	61755.422717	58647.360270	56399.125350	54823.472531	51680.112788	50810.531813	52647.477122	58153.995604	64829.891067	...	76295.125575	74951.160542	73672.590199	77737.724117	87015.319529	96344.104726	98636.766753	94795.229937	8744
Saturday	68991.576226	62628.416936	59321.825090	56745.339022	54816.778024	51275.831269	49144.134609	49769.161118	55718.168776	63423.248858	...	74654.603723	72910.123153	71972.130743	75921.019762	85309.097439	94902.067927	96852.132434	93754.497597	8828
Sunday	69728.136464	63113.302575	59243.071267	56207.553656	53896.627602	49677.139543	46078.167596	44983.099866	49619.256706	56571.380742	...	70039.555630	68585.463342	67570.378727	71844.793038	81580.676389	92171.105678	95540.060279	93301.028513	8706
Thursday	68293.632675	62187.544469	59063.328906	56958.133058	55465.004964	52336.590999	51408.387710	53305.035278	58985.960718	65850.016097	...	77129.099508	76024.015184	74771.523732	78859.415340	88130.713827	97361.509809	99639.276059	95745.056611	8846
Tuesday	67935.356547	61868.544847	58801.200586	56642.907327	55088.146346	52165.725002	51320.086009	53314.431143	58788.751316	65473.069281	...	76564.586378	75151.608847	73866.089284	78034.431306	87425.955170	96812.639776	98951.522986	95247.295786	8810
Wednesday	68239.842528	62022.724317	58798.761412	56626.703787	55114.497390	52157.866552	51411.322040	53186.989913	58830.511611	65583.730610	...	77084.718564	75799.737721	74690.144168	78963.032971	88010.877872	97121.892997	99259.106506	95347.105487	8796

7 rows × 24 columns

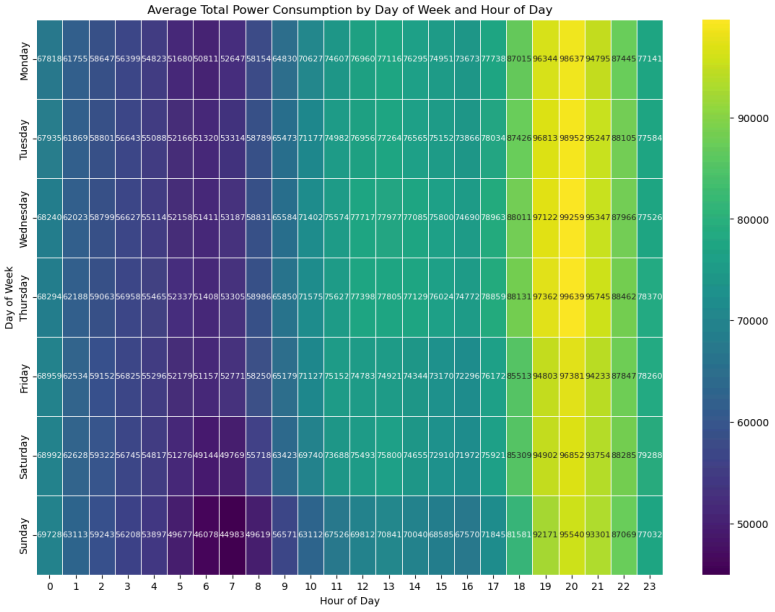
In [13]:
```
# Reorder the rows to have the days of the week in order
days_order = ['Monday','Tuesday','Wednesday','Thursday','Friday','Saturday','Sunday']
pivot_table = pivot_table.reindex(days_order)
pivot_table
```

Out[13]:

DayOfWeek	HourOfDay 0	1	2	3	4	5	6	7	8	9	...	14	15	16	17	18	19	20	21	
Monday	67817.926234	61755.422717	58647.360270	56399.125350	54823.472531	51680.112788	50810.531813	52647.477122	58153.995604	64829.891067	...	76295.125575	74951.160542	73672.590199	77737.724117	87015.319529	96344.104726	98636.766753	94795.229937	8744
Tuesday	67935.356547	61868.544847	58801.200586	56642.907327	55088.146346	52165.725002	51320.086009	53314.431143	58788.751316	65473.069281	...	76564.586378	75151.608847	73866.089284	78034.431306	87425.955170	96812.639776	98951.522986	95247.295786	8810
Wednesday	68239.842528	62022.724317	58798.761412	56626.703787	55114.497390	52157.866552	51411.322040	53186.989913	58830.511611	65583.730610	...	77084.718564	75799.737721	74690.144168	78963.032971	88010.877872	97121.892997	99259.106506	95347.105487	8796
Thursday	68293.632675	62187.544469	59063.328906	56958.133058	55465.004964	52336.590999	51408.387710	53305.035278	58985.960718	65850.016097	...	77129.099508	76024.015184	74771.523732	78859.415340	88130.713827	97361.509809	99639.276059	95745.056611	8846
Friday	68959.251459	62533.960861	59152.040758	56825.429786	55296.351259	52178.788896	51156.640309	52771.135698	58250.443622	65179.083324	...	74344.497887	73169.524043	72295.790052	76172.341277	85512.967870	94802.624915	97380.905756	94232.932202	8784
Saturday	68991.576226	62628.416936	59321.825090	56745.339022	54816.778024	51275.831269	49144.134609	49769.161118	55718.168776	63423.248858	...	74654.603723	72910.123153	71972.130743	75921.019762	85309.097439	94902.067927	96852.132434	93754.497597	8828
Sunday	69728.136464	63113.302575	59243.071267	56207.553656	53896.627602	49677.139543	46078.167596	44983.099866	49619.256706	56571.380742	...	70039.555630	68585.463342	67570.378727	71844.793038	81580.676389	92171.105678	95540.060279	93301.028513	8706

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, annot_kws = {"fontsize": 8})
plt.title('Average Total Power Consumption by Day of Week and Hour of Day')
plt.xlabel('Hour of Day')
plt.ylabel('Day of Week')
plt.show()
```



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.