```
In [2]:  import pandas as pd
         import numpy as np
```

```
In [4]:  import os
         import matplotlib.pyplot as plt
         import seaborn as sns
```

```
In [5]:  os.listdir('C:\\Users\\Ansh\\Downloads\\Datasets-20250505T055833Z-001\\Datasets')
```

```
Out[5]:  ['other-American_B01362.csv',
          'other-Carmel_B00256.csv',
          'other-Dial7_B00887.csv',
          'other-Diplo_B01196.csv',
          'other-Federal_02216.csv',
          'other-FHV-services_jan-aug-2015.csv',
          'other-Firstclass_B01536.csv',
          'other-Highclass_B01717.csv',
          'other-Lyft_B02510.csv',
          'other-Prestige_B01338.csv',
          'other-Skyline_B00111.csv',
          'Uber-Jan-Feb-FOIL.csv',
          'uber-raw-data-apr14.csv',
          'uber-raw-data-aug14.csv',
          'uber-raw-data-janjune-15.csv',
          'uber-raw-data-janjune-15_sample.csv',
          'uber-raw-data-jul14.csv',
          'uber-raw-data-jun14.csv',
          'uber-raw-data-may14.csv',
          'uber-raw-data-sep14.csv']
```

```
In [5]:  uber=pd.read_csv('C:\\Users\\Ansh\\Downloads\\Datasets-20250505T055833Z-001\\Datasets/uber-raw-data-janjune-15_
```

```
In [15]:  uber.shape
```

```
Out[15]:  (100000, 4)
```

```
In [17]:  uber.duplicated()
```

```
Out[17]:  0        False
          1        False
          2        False
          3        False
          4        False
                   ...
          99995    False
          99996    False
          99997    False
          99998    False
          99999    False
          Length: 100000, dtype: bool
```

```
In [19]:  uber.duplicated().sum()
```

```
Out[19]:  54
```

```
In [21]:  uber.drop_duplicates(inplace=True)
```

```
In [23]:  uber.duplicated().sum()
```

```
Out[23]:  0
```

1. Analysing which month has the max uber pickups

```
In [25]:  type(uber['Pickup_date'][0])
```

```
Out[25]:  str
```

```
In [27]:  uber['Pickup_date'] = pd.to_datetime(uber['Pickup_date'])
```

```
In [29]:  uber['Pickup_date'].dtype
```

```
Out[29]:  dtype('<M8[ns]')
```

```
In [31]:  uber.dtypes
```

```
Out[31]: Dispatching_base_num          object
         Pickup_date           datetime64[ns]
         Affiliated_base_num           object
         locationID                     int64
         dtype: object
```

```
In [33]: uber['Month']=uber['Pickup_date'].dt.month_name()
```

```
In [35]:  uber['Month']
```

```
Out[35]: 0            May
         1        January
         2          March
         3          April
         4          March
                   ...
         99995      April
         99996      March
         99997      March
         99998        May
         99999       June
         Name: Month, Length: 99946, dtype: object
```

```
In [37]: uber
```

Out[37]:

	Dispatching_base_num	Pickup_date	Affiliated_base_num	locationID	Month
0	B02617	2015-05-02 21:43:00	B02764	237	May
1	B02682	2015-01-20 19:52:59	B02682	231	January
2	B02617	2015-03-19 20:26:00	B02617	161	March
3	B02764	2015-04-10 17:38:00	B02764	107	April
4	B02764	2015-03-23 07:03:00	B00111	140	March
...	...	...	...	...	...
99995	B02764	2015-04-13 16:12:00	B02764	234	April
99996	B02764	2015-03-06 21:32:00	B02764	24	March
99997	B02598	2015-03-19 19:56:00	B02598	17	March
99998	B02682	2015-05-02 16:02:00	B02682	68	May
99999	B02764	2015-06-24 16:04:00	B02764	125	June

99946 rows × 5 columns

```
In [39]: uber['Month'].value_counts()
```
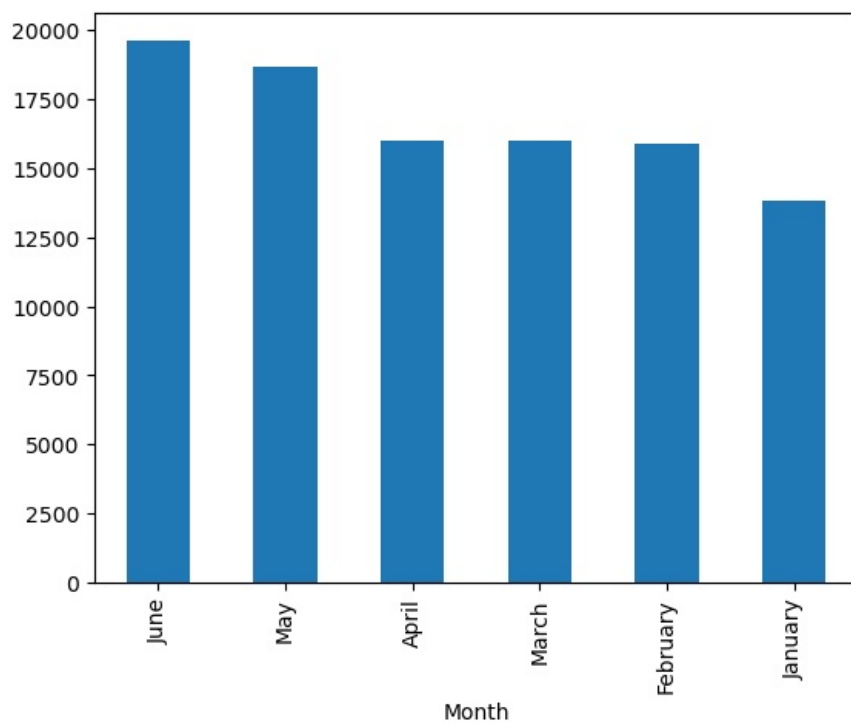
```
Out[39]: Month
         June        19620
         May         18660
         April       15982
         March       15969
         February    15896
         January     13819
         Name: count, dtype: int64
```

```
In [41]: uber['Month'].value_counts().plot(kind='bar')
```

```
Out[41]: <Axes: xlabel='Month'>
```

Inference = Uber has the maximum pickups in June

```
In [43]: uber['Weekday'] = uber['Pickup_date'].dt.day_name()
         uber['Day'] = uber['Pickup_date'].dt.day
         uber['Hour'] = uber['Pickup_date'].dt.hour
         uber['Minute'] = uber['Pickup_date'].dt.minute
```

```
In [45]: pivot=pd.crosstab(index=uber['Month'], columns=uber['Weekday'])
```

```
In [47]: pivot
```

Out[47]:

Weekday	Friday	Monday	Saturday	Sunday	Thursday	Tuesday	Wednesday
Month							
April	2365	1833	2508	2052	2823	1880	2521
February	2655	1970	2550	2183	2396	2129	2013
January	2508	1353	2745	1651	2378	1444	1740
June	2793	2848	3037	2485	2767	3187	2503
March	2465	2115	2522	2379	2093	2388	2007
May	3262	1865	3519	2944	2627	2115	2328

```
In [49]: pivot.plot(kind='bar')
```
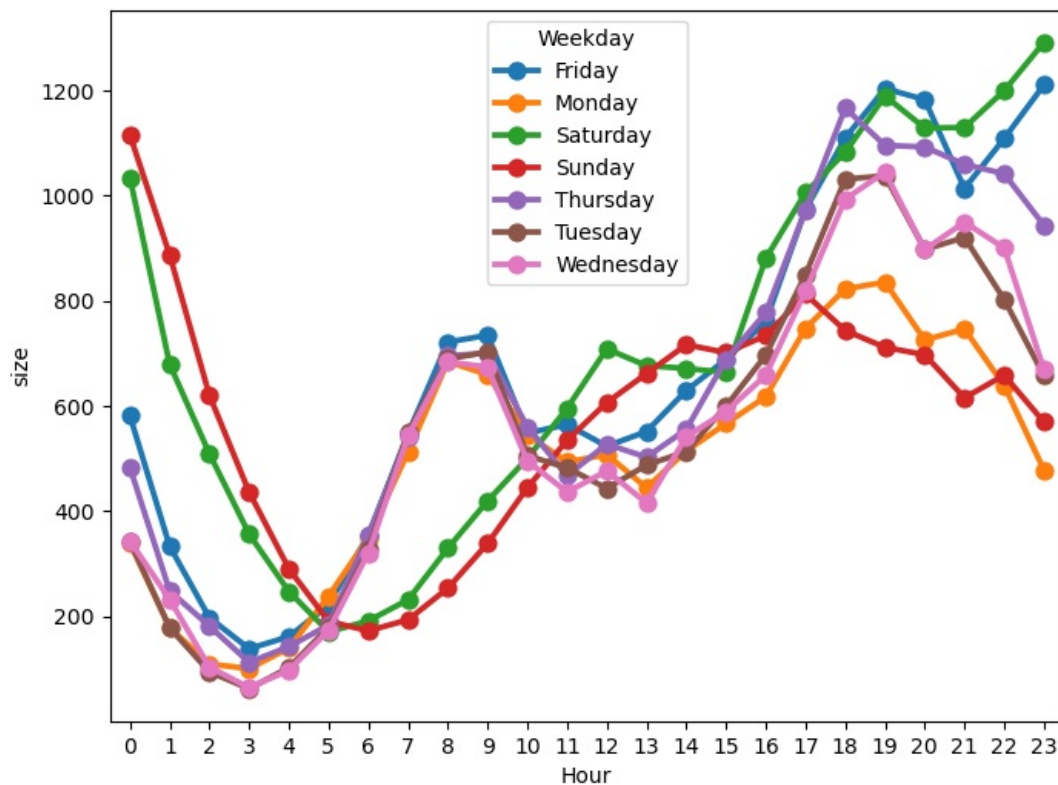
Out[49]: <Axes: xlabel='Month'>
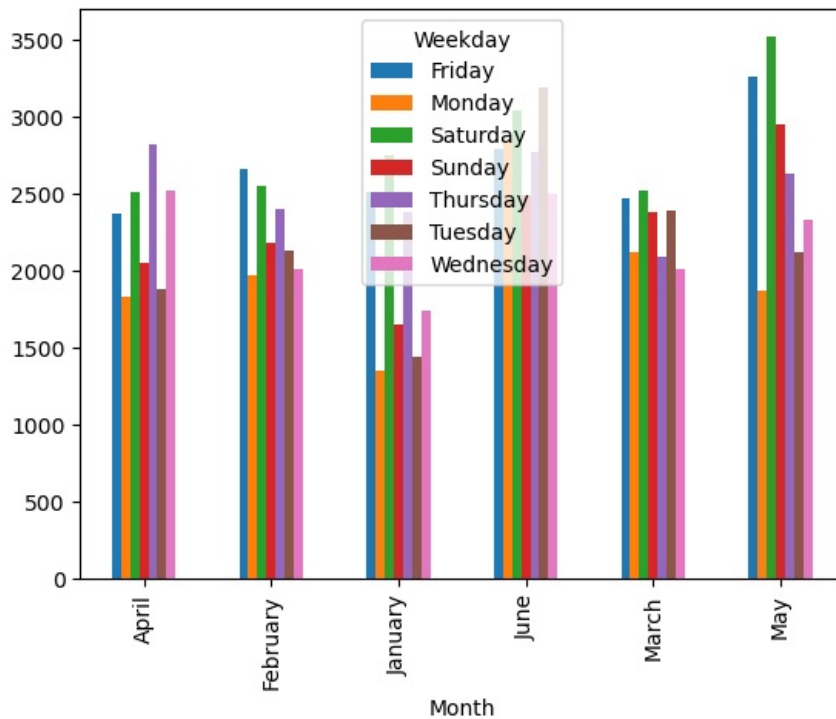
Inference = On Saturdays & Fridays, we are getting more uber pickups

2. Hourly Rush in New York City

```
In [57]: summary=uber.groupby(['Weekday', 'Hour'], as_index=False).size()
```

```
In [68]: plt.figure(figsize=(8,6))
         sns.pointplot(x='Hour', y='size', hue='Weekday', data=summary)
```

```
Out[68]: <Axes: xlabel='Hour', ylabel='size'>
```


3. Analysing Most Active Uber Base-Number

```
In [10]: os.listdir(r'C:\Users\Ansh\Downloads\Datasets-20250505T055833Z-001\Datasets')
```

```
Out[10]:  ['other-American_B01362.csv',
           'other-Carmel_B00256.csv',
           'other-Dial7_B00887.csv',
           'other-Diplo_B01196.csv',
           'other-Federal_02216.csv',
           'other-FHV-services_jan-aug-2015.csv',
           'other-Firstclass_B01536.csv',
           'other-Highclass_B01717.csv',
           'other-Lyft_B02510.csv',
           'other-Prestige_B01338.csv',
           'other-Skyline_B00111.csv',
           'Uber-Jan-Feb-FOIL.csv',
           'uber-raw-data-apr14.csv',
           'uber-raw-data-aug14.csv',
           'uber-raw-data-janjune-15.csv',
           'uber-raw-data-janjune-15_sample.csv',
           'uber-raw-data-jul14.csv',
           'uber-raw-data-jun14.csv',
           'uber-raw-data-may14.csv',
           'uber-raw-data-sep14.csv']
```

In [12]:
```
uber_foil=pd.read_csv(r'C:\Users\Ansh\Downloads\Datasets-20250505T055833Z-001\Datasets/Uber-Jan-Feb-FOIL.csv')
```

In [14]:
```
uber_foil
```

Out[14]:

	dispatching_base_number	date	active_vehicles	trips
0	B02512	1/1/2015	190	1132
1	B02765	1/1/2015	225	1765
2	B02764	1/1/2015	3427	29421
3	B02682	1/1/2015	945	7679
4	B02617	1/1/2015	1228	9537
...	...	...	...	...
349	B02764	2/28/2015	3952	39812
350	B02617	2/28/2015	1372	14022
351	B02682	2/28/2015	1386	14472
352	B02512	2/28/2015	230	1803
353	B02765	2/28/2015	747	7753

354 rows × 4 columns

In [16]:
```
!pip install chart_studio
!pip install plotly
```
```
Collecting chart_studio
  Downloading chart_studio-1.1.0-py3-none-any.whl.metadata (1.3 kB)
Requirement already satisfied: plotly in c:\users\ansh\anaconda3\lib\site-packages (from chart_studio) (5.24.1)
Requirement already satisfied: requests in c:\users\ansh\anaconda3\lib\site-packages (from chart_studio) (2.32.3
)
Collecting retrying>=1.3.3 (from chart_studio)
  Downloading retrying-1.3.4-py3-none-any.whl.metadata (6.9 kB)
Requirement already satisfied: six in c:\users\ansh\anaconda3\lib\site-packages (from chart_studio) (1.16.0)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\ansh\anaconda3\lib\site-packages (from plotly->chart_
studio) (8.2.3)
Requirement already satisfied: packaging in c:\users\ansh\anaconda3\lib\site-packages (from plotly->chart_studio
) (24.1)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\ansh\anaconda3\lib\site-packages (from reque
sts->chart_studio) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in c:\users\ansh\anaconda3\lib\site-packages (from requests->chart_s
tudio) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\ansh\anaconda3\lib\site-packages (from requests->c
hart_studio) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\ansh\anaconda3\lib\site-packages (from requests->c
hart_studio) (2025.1.31)
Downloading chart_studio-1.1.0-py3-none-any.whl (64 kB)
Downloading retrying-1.3.4-py3-none-any.whl (11 kB)
Installing collected packages: retrying, chart_studio
Successfully installed chart_studio-1.1.0 retrying-1.3.4
Requirement already satisfied: plotly in c:\users\ansh\anaconda3\lib\site-packages (5.24.1)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\ansh\anaconda3\lib\site-packages (from plotly) (8.2.3
)
Requirement already satisfied: packaging in c:\users\ansh\anaconda3\lib\site-packages (from plotly) (24.1)
```

In [18]:
```
import chart_studio.plotly as py
import plotly.graph_objs as go
import plotly.express as px
```

```
from plotly.offline import download_plotlyjs , init_notebook_mode , plot , iplot
```

In [20]:
```
init_notebook_mode(connected=True)
```

In [22]:
```
uber_foil.columns
```

Out[22]:
```
Index(['dispatching_base_number', 'date', 'active_vehicles', 'trips'], dtype='object')
```

In [28]:
```
px.box(x='dispatching_base_number', y='active_vehicles', data_frame=uber_foil)
```

4. Collecting Data for Analysis

In [10]:
```
files=os.listdir(r'C:\Users\Ansh\Downloads\Datasets-20250505T055833Z-001\Datasets')[-8:]
```

In [12]:
```
files.remove('uber-raw-data-janjune-15.csv')
```

In [14]:
```
files.remove('uber-raw-data-janjune-15_sample.csv')
```

In [16]:
```
files
```

Out[16]:
```
['uber-raw-data-apr14.csv',
 'uber-raw-data-aug14.csv',
 'uber-raw-data-jul14.csv',
 'uber-raw-data-jun14.csv',
 'uber-raw-data-may14.csv',
 'uber-raw-data-sep14.csv']
```

In [18]:
```
final=pd.DataFrame()
path=r'C:\Users\Ansh\Downloads\Datasets-20250505T055833Z-001\Datasets'
for file in files:
    current_file=pd.read_csv(path+'/'+file)
    final=pd.concat([current_file,final])
```

In [22]:
```
final.shape
```

Out[22]:
```
(4534327, 4)
```

In [24]:
```
final.duplicated().sum()
```

Out[24]:
```
82581
```

In [28]:
```
final.drop_duplicates(inplace=True)
```

In [30]:
```
final.shape
```

Out[30]:
```
(4451746, 4)
```

In [32]:
```
final.head()
```

Out[32]:

	Date/Time	Lat	Lon	Base
0	9/1/2014 0:01:00	40.2201	-74.0021	B02512
1	9/1/2014 0:01:00	40.7500	-74.0027	B02512
2	9/1/2014 0:03:00	40.7559	-73.9864	B02512
3	9/1/2014 0:06:00	40.7450	-73.9889	B02512
4	9/1/2014 0:11:00	40.8145	-73.9444	B02512

Spatial Analysis of Locations at which We are getting Rush in New York City

In [43]: `uber_rush=final.groupby(['Lat','Lon'], as_index=False).size()`

In [45]: `uber_rush`

Out[45]:

	Lat	Lon	size
0	39.6569	-74.2258	1
1	39.6686	-74.1607	1
2	39.7214	-74.2446	1
3	39.8416	-74.1512	1
4	39.9055	-74.0791	1
...	...	...	...
574553	41.3730	-72.9237	1
574554	41.3737	-73.7988	1
574555	41.5016	-72.8987	1
574556	41.5276	-72.7734	1
574557	42.1166	-72.0666	1

574558 rows × 3 columns

In [47]: `import folium`

In [49]: `map=folium.Map()`

In [51]: `map`

Out[51]:

In [55]: `from folium.plugins import HeatMap`

In [57]: `HeatMap(uber_rush).add_to(map)`

```
Out[57]:    <folium.plugins.heat_map.HeatMap at 0x1f282877440>
```

```
In [59]:    map
```

Out[59]:

Inference = We can see a number of hot spots here. Midtown Manhattan is clearly a huge bright spot

Pair Wise Analysis of Rush on Hour and Day

```
In [64]:    final.dtypes
```

```
Out[64]:    Date/Time    object
            Lat          float64
            Lon          float64
            Base         object
            dtype: object
```

```
In [66]:    final['Date/Time'] = pd.to_datetime(final['Date/Time'] , format="%m/%d/%Y %H:%M:%S")
```

```
In [68]:    final['Date/Time'].dtype
```

```
Out[68]:    dtype('<M8[ns]')
```

```
In [70]:    final.head()
```

Out[70]:

	Date/Time	Lat	Lon	Base
0	2014-09-01 00:01:00	40.2201	-74.0021	B02512
1	2014-09-01 00:01:00	40.7500	-74.0027	B02512
2	2014-09-01 00:03:00	40.7559	-73.9864	B02512
3	2014-09-01 00:06:00	40.7450	-73.9889	B02512
4	2014-09-01 00:11:00	40.8145	-73.9444	B02512

```
In [74]:    final['Day'] = final['Date/Time'].dt.day
            final['Hour'] = final['Date/Time'].dt.hour
```

```
In [76]:    pivot = final.groupby(['Day' , 'Hour']).size().unstack()
```

```
In [78]:    pivot
```

Out[78]:

Hour	0	1	2	3	4	5	6	7	8	9	...	14	15	16	17	18	19	20	21
Day																			
1	3178	1944	1256	1308	1429	2126	3664	5380	5292	4617	...	6933	7910	8633	9511	8604	8001	7315	7803
2	2435	1569	1087	1414	1876	2812	4920	6544	6310	4712	...	6904	8449	10109	11100	11123	9474	8759	8357
3	3354	2142	1407	1467	1550	2387	4241	5663	5386	4657	...	7226	8850	10314	10491	11239	9599	9026	8531
4	2897	1688	1199	1424	1696	2581	4592	6029	5704	4744	...	7158	8515	9492	10357	10259	9097	8358	8649
5	2733	1541	1030	1253	1617	2900	4814	6261	6469	5530	...	6955	8312	9609	10699	10170	9430	9354	9610
6	4537	2864	1864	1555	1551	2162	3642	4766	4942	4401	...	7235	8612	9444	9929	9263	8405	8117	8567
7	3645	2296	1507	1597	1763	2422	4102	5575	5376	4639	...	7276	8474	10393	11013	10573	9472	8691	8525
8	2830	1646	1123	1483	1889	3224	5431	7361	7357	5703	...	7240	8775	9851	10673	9687	8796	8604	8367
9	2657	1724	1222	1480	1871	3168	5802	7592	7519	5895	...	7877	9220	10270	11910	11449	9804	8909	8665
10	3296	2126	1464	1434	1591	2594	4664	6046	6158	5072	...	7612	9578	11045	11875	10934	9613	9687	9240
11	3036	1665	1095	1424	1842	2520	4954	6876	6871	5396	...	7503	8920	10125	10898	10361	9327	8824	8730
12	3227	2147	1393	1362	1757	2710	4576	6250	6231	5177	...	7743	9390	10734	11713	12216	10393	9965	10310
13	5408	3509	2262	1832	1705	2327	4196	5685	6060	5631	...	8200	9264	10534	11826	11450	9921	8705	8423
14	3748	2349	1605	1656	1756	2629	4257	5781	5520	4824	...	6963	8192	9511	10115	9553	9146	9182	8589
15	2497	1515	1087	1381	1862	2980	5050	6837	6729	5201	...	7633	8505	10285	11959	11728	11032	10509	9105
16	2547	1585	1119	1395	1818	2966	5558	7517	7495	5958	...	7597	9290	10804	11773	10855	10924	10142	10374
17	3155	2048	1500	1488	1897	2741	4562	6315	5882	4934	...	7472	8997	10323	11236	11089	9919	9935	9823
18	3390	2135	1332	1626	1892	2959	4688	6618	6451	5377	...	7534	9040	10274	10692	10338	9551	9310	9285
19	3217	2188	1604	1675	1810	2639	4733	6159	6014	5006	...	7374	8898	9893	10741	10429	9701	10051	10049
20	4475	3190	2100	1858	1618	2143	3584	4900	5083	4765	...	7462	8630	9448	10046	9272	8592	8614	8703
21	4294	3194	1972	1727	1926	2615	4185	5727	5529	4707	...	7064	8127	9483	9817	9291	8317	8107	8245
22	2787	1637	1175	1468	1934	3151	5204	6872	6850	5198	...	7337	9148	10574	10962	9884	8980	8772	8430
23	2546	1580	1136	1429	1957	3132	5204	6890	6436	5177	...	7575	9309	9980	10341	10823	11347	11447	10347
24	3200	2055	1438	1493	1798	2754	4484	6013	5913	5146	...	7083	8706	10366	10786	9772	9080	9213	8831
25	2405	1499	1072	1439	1943	2973	5356	7627	7078	5994	...	7298	8732	9922	10504	10673	9048	8751	9508
26	3810	3065	2046	1806	1730	2337	3776	5172	5071	4808	...	7269	8815	9885	10697	10867	10122	9820	10441
27	5196	3635	2352	2055	1723	2336	3539	4937	5053	4771	...	7519	8803	9793	9838	9228	8267	7908	8507
28	4123	2646	1843	1802	1883	2793	4290	5715	5671	5206	...	7341	8584	9671	9975	9132	8255	8309	7949
29	2678	1827	1409	1678	1948	3056	5213	6852	6695	5481	...	7630	9249	10105	11113	10411	9301	9270	9114
30	2401	1510	1112	1403	1841	3216	5757	7596	7611	6064	...	8396	10243	11554	12126	12561	11024	10836	10042
31	2174	1394	1087	919	773	997	1561	2169	2410	2525	...	4104	5099	5386	5308	5350	4898	4819	5064

31 rows × 24 columns

In [80]: `pivot.style.background_gradient()`

Hour Day	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	3178	1944	1256	1308	1429	2126	3664	5380	5292	4617	4607	4729	4930	5794	6933	7910	8633	9511	8604
2	2435	1569	1087	1414	1876	2812	4920	6544	6310	4712	4797	4975	5188	5695	6904	8449	10109	11100	11123
3	3354	2142	1407	1467	1550	2387	4241	5663	5386	4657	4788	5065	5384	6093	7226	8850	10314	10491	11239
4	2897	1688	1199	1424	1696	2581	4592	6029	5704	4744	4743	4975	5193	6175	7158	8515	9492	10357	10259
5	2733	1541	1030	1253	1617	2900	4814	6261	6469	5530	5141	5011	5047	5690	6955	8312	9609	10699	10170
6	4537	2864	1864	1555	1551	2162	3642	4766	4942	4401	4801	5174	5426	6258	7235	8612	9444	9929	9263
7	3645	2296	1507	1597	1763	2422	4102	5575	5376	4639	4905	5166	5364	6214	7276	8474	10393	11013	10573
8	2830	1646	1123	1483	1889	3224	5431	7361	7357	5703	5288	5350	5483	6318	7240	8775	9851	10673	9687
9	2657	1724	1222	1480	1871	3168	5802	7592	7519	5895	5406	5443	5496	6419	7877	9220	10270	11910	11449
10	3296	2126	1464	1434	1591	2594	4664	6046	6158	5072	4976	5415	5506	6527	7612	9578	11045	11875	10934
11	3036	1665	1095	1424	1842	2520	4954	6876	6871	5396	5215	5423	5513	6486	7503	8920	10125	10898	10361
12	3227	2147	1393	1362	1757	2710	4576	6250	6231	5177	5157	5319	5570	6448	7743	9390	10734	11713	12216
13	5408	3509	2262	1832	1705	2327	4196	5685	6060	5631	5442	5720	5914	6678	8200	9264	10534	11826	11450
14	3748	2349	1605	1656	1756	2629	4257	5781	5520	4824	4911	5118	5153	5747	6963	8192	9511	10115	9553
15	2497	1515	1087	1381	1862	2980	5050	6837	6729	5201	5347	5517	5503	6997	7633	8505	10285	11959	11728
16	2547	1585	1119	1395	1818	2966	5558	7517	7495	5958	5626	5480	5525	6198	7597	9290	10804	11773	10855
17	3155	2048	1500	1488	1897	2741	4562	6315	5882	4934	5004	5306	5634	6507	7472	8997	10323	11236	11089
18	3390	2135	1332	1626	1892	2959	4688	6618	6451	5377	5150	5487	5490	6383	7534	9040	10274	10692	10338
19	3217	2188	1604	1675	1810	2639	4733	6159	6014	5006	5092	5240	5590	6367	7374	8898	9893	10741	10429
20	4475	3190	2100	1858	1618	2143	3584	4900	5083	4765	5135	5650	5745	6656	7462	8630	9448	10046	9272
21	4294	3194	1972	1727	1926	2615	4185	5727	5529	4707	4911	5212	5465	6085	7064	8127	9483	9817	9291
22	2787	1637	1175	1468	1934	3151	5204	6872	6850	5198	5277	5352	5512	6342	7337	9148	10574	10962	9884
23	2546	1580	1136	1429	1957	3132	5204	6890	6436	5177	5066	5304	5504	6232	7575	9309	9980	10341	10823
24	3200	2055	1438	1493	1798	2754	4484	6013	5913	5146	4947	5311	5229	5974	7083	8706	10366	10786	9772
25	2405	1499	1072	1439	1943	2973	5356	7627	7078	5994	5432	5504	5694	6204	7298	8732	9922	10504	10673
26	3810	3065	2046	1806	1730	2337	3776	5172	5071	4808	5061	5179	5381	6166	7269	8815	9885	10697	10867
27	5196	3635	2352	2055	1723	2336	3539	4937	5053	4771	5198	5732	5839	6820	7519	8803	9793	9838	9228
28	4123	2646	1843	1802	1883	2793	4290	5715	5671	5206	5247	5500	5486	6120	7341	8584	9671	9975	9132
29	2678	1827	1409	1678	1948	3056	5213	6852	6695	5481	5234	5163	5220	6305	7630	9249	10105	11113	10411
30	2401	1510	1112	1403	1841	3216	5757	7596	7611	6064	5987	6090	6423	7249	8396	10243	11554	12126	12561
31	2174	1394	1087	919	773	997	1561	2169	2410	2525	2564	2777	2954	3280	4104	5099	5386	5308	5350

In [ ]: