

```
In [1]: # imports
import pandas as pd
import plotly.offline as pyo
import plotly.graph_objs as go
```

```
In [2]: df = pd.read_csv('C:/Users/eugen/OneDrive/Main_Env/udemy_dash_course/Data/2010YumaAZ.csv')
df
```

	LST_DATE	DAY	LST_TIME	T_HR_AVG
0	20100601	TUESDAY	0:00	25.2
1	20100601	TUESDAY	1:00	24.1
2	20100601	TUESDAY	2:00	24.4
3	20100601	TUESDAY	3:00	24.9
4	20100601	TUESDAY	4:00	22.8
...	...	...	...	...
163	20100607	MONDAY	19:00	39.4
164	20100607	MONDAY	20:00	38.5
165	20100607	MONDAY	21:00	37.0
166	20100607	MONDAY	22:00	34.7
167	20100607	MONDAY	23:00	32.6

168 rows × 4 columns

```
In [3]: days = [x for x in df['DAY'].unique()]
days
```

```
['TUESDAY', 'WEDNESDAY', 'THURSDAY', 'FRIDAY', 'SATURDAY', 'SUNDAY', 'MONDAY']
```

```
In [4]: #data
x = df['T_HR_AVG']
y = df['DAY']
```

```
df[['T_HR_AVG', 'DAY']][['DAY']=='TUESDAY']
```

0	25.2
1	24.1
2	24.4
3	24.9
4	22.8
...	...
163	39.4
164	38.5
165	37.0
166	34.7
167	32.6

Name: T\_HR\_AVG, Length: 168, dtype: float64

```
In [5]: # traces
data = []
for d in days:
    traces = go.Scatter(x=df['LST_TIME'],
                        y=df[df['DAY']==d]['T_HR_AVG'],
                        mode='lines',
                        name=d,)
    data.append(traces)

In [6]: # # data list
# data=[go.Scatter(x=df[['T_HR_AVG', 'DAY']][['DAY']==n], y=df['DAY']==n,
#                 mode='markers+lines',
#                 name='markers') for n in days] #needs to be a list in plotly

In [7]: # Layout design options - title, axis, etc
layout = go.Layout(title='Line Chart')

In [8]: # plotting
fig = go.Figure(data=data, layout=layout)
pyo.plot(fig)

'temp-plot.html'
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```