Date 05-07-2021

Line Charts Exercise

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

In [2]:
```
temperature_data = pd.read_csv("2010YumaAZ.csv" ,usecols=["DAY", "LST_TIME", "T_HR_AVG"])
temperature_data
```

Out[2]:

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

168 rows × 3 columns

In [3]:
```
temperature_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 168 entries, 0 to 167
Data columns (total 3 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   DAY       168 non-null    object 
 1   LST_TIME  168 non-null    object 
 2   T_HR_AVG  168 non-null    float64
dtypes: float64(1), object(2)
memory usage: 4.1+ KB
```

In [4]:
```
temperature_data.describe()
```

Out[4]:

	T_HR_AVG
count	168.000000
mean	30.402976
std	6.288974
min	17.900000
25%	25.125000
50%	30.900000
75%	35.525000
max	40.600000

In [5]:
```
days = np.array(['TUESDAY', 'WEDNESDAY', 'THURSDAY', 'FRIDAY', 'SATURDAY','SUNDAY', 'MONDAY'])
```

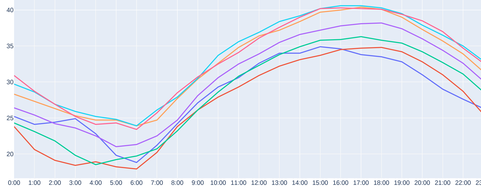
In [6]:
```
data = []
```

In [7]:
```
for day in days:
    trace_0 = go.Scatter(x=temperature_data["LST_TIME"],
                         y=temperature_data[temperature_data["DAY"]==day]['T_HR_AVG'],
                         mode='lines',
                         name=day)
    data.append(trace_0)
```

In [8]:
```
layout = go.Layout(title="Daily Temperature from June 1-7, 2010 in Yuma, Arizona",)
```

In [9]:
```
fig = go.Figure(data=data, layout=layout)
```

In [10]:
```
pyo.iplot(fig)
```

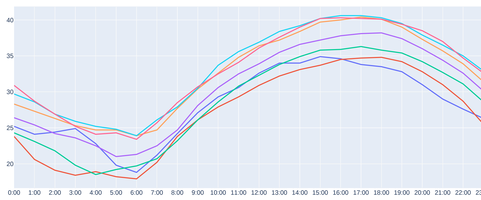


In [11]:
```
pyo.plot(fig, filename="tutorial_5 (Line Charts Exercise).html",
         image_width=1600,
         image_height=900,)
```

Out[11]:
```
'tutorial_5 (Line Charts Exercise).html'
```

Instructor First Solution is Down

In [12]:
```
######
# Objective: Using the file 2010YumaAZ.csv, develop a Line Chart
# that plots seven days worth of temperature data on one graph.
# You can use a for loop to assign each day to its own trace.
######

# Perform imports here:
import plotly.offline as pyo
import plotly.graph_objs as go
import pandas as pd

# Create a pandas DataFrame from 2010YumaAZ.csv
df = pd.read_csv('2010YumaAZ.csv')
days = ['TUESDAY','WEDNESDAY','THURSDAY','FRIDAY','SATURDAY','SUNDAY','MONDAY']

# Use a for loop to create the traces for the seven days
# There are many ways to do this! Could also do this with a
# list comprehension.

data = []

for day in days:
    trace = go.Scatter(x=df['LST_TIME'],
                       y=df[df['DAY']==day]['T_HR_AVG'],
                       mode='lines',
                       name=day)
    data.append(trace)

# Define the layout
layout = go.Layout(
    title='Daily temperatures from June 1-7, 2010 in Yuma, Arizona',
    hovermode='closest'
)

# Create a fig from data and layout, and plot the fig
fig = go.Figure(data=data, layout=layout)

pyo.iplot(fig)
```



Instructor Second Solution is Down

In [13]:
```
###################
## NOTE: ADVANCED SOLUTION THAT USES ONLY PURE DF CALLS
## THIS IS FOR MORE ADVANCED PANDAS USERS TO TAKE A LOOK AT! :)

######
# Objective: Using the file 2010YumaAZ.csv, develop a Line Chart
# that plots seven days worth of temperature data on one graph.
# You can use a for loop to assign each day to its own trace.
######

# Perform imports here:
import plotly.offline as pyo
import plotly.graph_objs as go
import pandas as pd

# Create a pandas DataFrame from 2010YumaAZ.csv
df = pd.read_csv('2010YumaAZ.csv')

# Define a data variable
data = [{
    'x': df['LST_TIME'],
    'y': df[df['DAY']==day]['T_HR_AVG'],
    'name': day
} for day in df['DAY'].unique()]

# Define the layout
layout = go.Layout(
    title='Daily temperatures from June 1-7, 2010 in Yuma, Arizona',
    hovermode='closest'
)

# Create a fig from data and layout, and plot the fig
fig = go.Figure(data=data, layout=layout)

pyo.iplot(fig)
```