Date 07-07-2021

Dash Layouts - Part One

In [1]:
```
import plotly.offline as pyo
import plotly.graph_objs as go
import plotly
import dash
import dash_core_components as dcc
import dash_html_components as html
import numpy as np

versions_of_modules_used = {dash.__name__ : dash.__version__,
                            dcc.__name__ : dcc.__version__,
                            plotly.__name__ : plotly.__version__,
                            html.__name__ : html.__version__,
                            np.__name__ : np.__version__}

for i, j in versions_of_modules_used.items():
        print(i,"=",j)
```

```
dash = 1.20.0
dash_core_components = 1.16.0
plotly = 5.1.0
dash_html_components = 1.1.3
numpy = 1.19.2
```

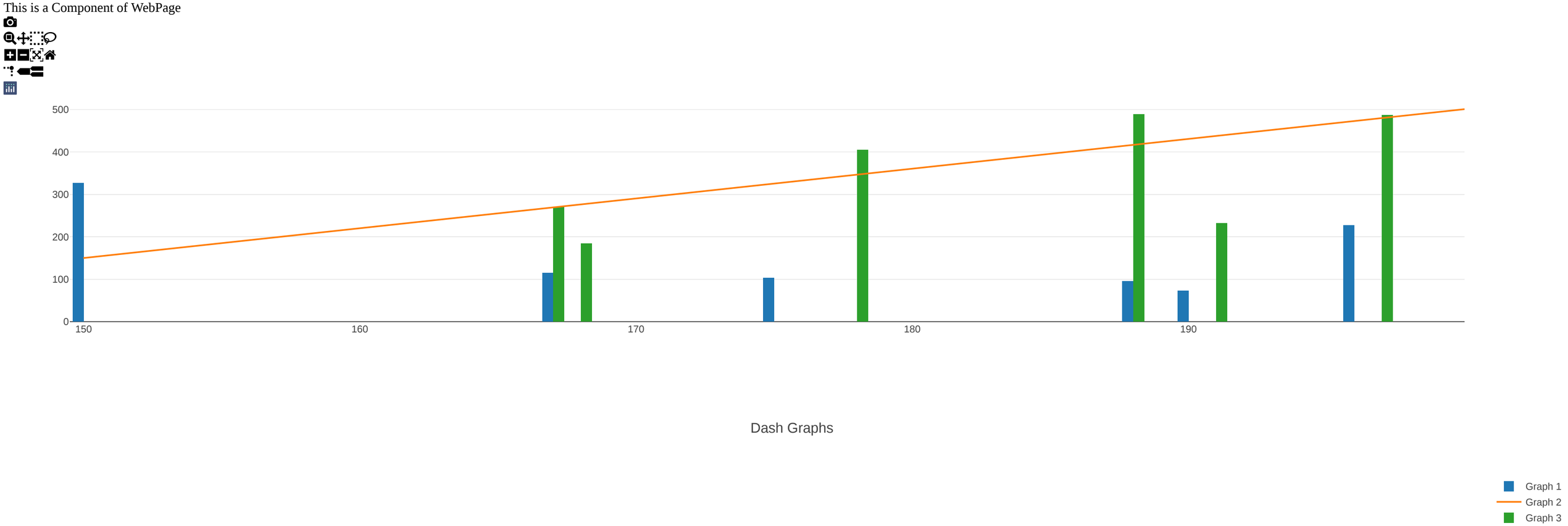
Module Name	Module Versions
dash	1.20.0
dash_core_components	1.16.0
plotly	5.1.0
dash_html_components	1.1.3
numpy	1.19.2

In [2]:
```
app = dash.Dash()
app.layout = html.Div(children = [html.H1("I\'m a Heading"),
                                  html.Div("This is a Component of WebPage"),
                                  dcc.Graph(id = "Example",
                                            figure = dict(data = [{'x': np.random.randint(150,200,6), 'y': np.random.randint(0,500,6), 'type': 'bar', 'name': 'Graph 1'},
                                                                  {'x': np.linspace(149.99,199.99,42), 'y': np.linspace(149.99,500.99,42), 'type': 'markers', 'name': 'Graph 2'},
                                                                  {'x': np.random.randint(150,200,6), 'y': np.random.randint(0,500,6), 'type': 'bar', 'name': 'Graph 3'}],
                                                          layout = {'title': 'Dash Graphs'}))])
app.run_server()
```

```
Dash is running on http://127.0.0.1:8050/

 * Serving Flask app "__main__" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:8050/ (Press CTRL+C to quit)
127.0.0.1 - - [07/Jul/2021 13:58:52] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [07/Jul/2021 13:58:53] "GET /_dash-layout HTTP/1.1" 200 -
127.0.0.1 - - [07/Jul/2021 13:58:53] "GET /_dash-dependencies HTTP/1.1" 200 -
```

I'm a Heading

This is a Component of WebPage



Date 07-07-2021

Dash Layouts - Part Two - Styling

In [1]:
```
import plotly.offline as pyo
import plotly.graph_objs as go
import plotly
import dash
import dash_core_components as dcc
import dash_html_components as html
import numpy as np
import pandas as pd

versions_of_modules_used = {dash.__name__ : dash.__version__,
                            dcc.__name__ : dcc.__version__,
                            plotly.__name__ : plotly.__version__,
                            html.__name__ : html.__version__,
                            np.__name__ : np.__version__,
                            pd.__name__ : pd.__version__}

for i, j in versions_of_modules_used.items():
        print(i,"=",j)
```

```
dash = 1.20.0
dash_core_components = 1.16.0
plotly = 5.1.0
dash_html_components = 1.1.3
numpy = 1.19.2
pandas = 1.1.3
```

Module Name	Module Versions
dash	1.20.0
dash_core_components	1.16.0
plotly	5.1.0
dash_html_components	1.1.3
numpy	1.19.2
pandas	1.1.3

In [2]:
```
winter_olympics_2018_data = pd.read_csv("2018WinterOlympics.csv", usecols = ["Total", "NOC"])
winter_olympics_2018_data
```

Out[2]:

	NOC	Total
0	Norway	39
1	Germany	31
2	Canada	29
3	United States	23
4	Netherlands	20
5	Sweden	14
6	Republic of Korea	17
7	Switzerland	15
8	France	15
9	Austria	14
10	Japan	13
11	Italy	10
12	OAR	17
13	Czech Republic	7
14	Belarus	3
15	China	9
16	Slovakia	3
17	Finland	6
18	Great Britain	5
19	Poland	2
20	Hungary	1
21	Ukraine	1
22	Australia	3
23	Slovenia	2
24	Belgium	1
25	Spain	2
26	New Zealand	2
27	Kazakhstan	1
28	Latvia	1
29	Liechtenstein	1

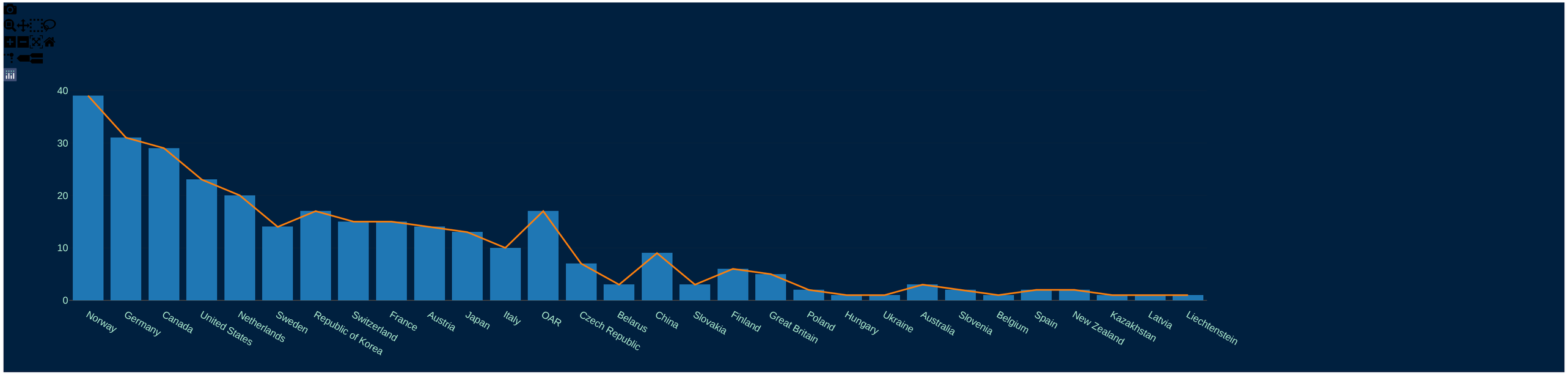
In [3]:
```
colors = dict(background = "#00203FFF", text = "#ADEFD1FF")
# colors["text"]
```

In [4]:
```
heading_style = {'font-size' : '50px',
'line-height' : '40px',
'margin' : '1em 0 .6em 0',
'font-weight' : 'normal',
'color' : 'white',
'font-family' : 'Hammersmith One',
'text-shadow' : '0 1px 0 rgba(0,0,0,0.4)',
'position' : 'relative',
'color' : '#6Cf',
'text-align' : 'center',}
```

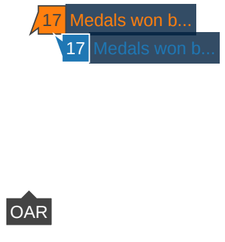
In [5]:
```
app = dash.Dash()
app.layout = html.Div(children = [html.H1("I\'m a Heading", style = heading_style),
                                  dcc.Graph(id = "Example",
                                            figure = dict(data = [{'x': winter_olympics_2018_data["NOC"], 'y': winter_olympics_2018_data["Total"], 'type': 'bar', 'name': 'Medals won by diffrent countries in 2018 Winter Olympics'},
                                                                  {'x': winter_olympics_2018_data["NOC"], 'y': winter_olympics_2018_data["Total"], 'type': 'markers', 'name': 'Medals won by diffrent countries in 2018 Winter Olympi
                                                          layout = {'title' : 'Dash Graphs',
                                                                    'plot_bgcolor' : colors['background'],
                                                                    'paper_bgcolor' : colors['background'],
                                                                    'font' : {'color' : colors['text']}})])])

app.run_server()
```

```
Dash is running on http://127.0.0.1:8050/

 * Serving Flask app "__main__" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:8050/ (Press CTRL+C to quit)
```

I'm a Heading



Dash Graphs



Date 07-07-2021

Converting Simple Plotly Plot to Dashboard with Dash

In [1]:
```
import plotly.offline as pyo
import plotly.graph_objs as go
from plotly import subplots
import plotly
import dash
import dash_core_components as dcc
import dash_html_components as html
import numpy as np
import pandas as pd

versions_of_modules_used = {dash.__name__ : dash.__version__,
                            dcc.__name__ : dcc.__version__,
                            plotly.__name__ : plotly.__version__,
                            html.__name__ : html.__version__,
                            np.__name__ : np.__version__,
                            pd.__name__ : pd.__version__}

for i, j in versions_of_modules_used.items():
        print(i,"=",j)
```

```
dash = 1.20.0
dash_core_components = 1.16.0
plotly = 5.1.0
dash_html_components = 1.1.3
numpy = 1.19.2
pandas = 1.1.3
```

Module Name	Module Versions
dash	1.20.0
dash_core_components	1.16.0
plotly	5.1.0
dash_html_components	1.1.3
numpy	1.19.2
pandas	1.1.3

In [2]:
```
mpg_data_csv = pd.read_csv("mpg.csv")
mpg_data_csv
```

Out[2]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	name
0	18.0	8	307.0	130	3504	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350.0	165	3693	11.5	70	1	buick skylark 320
2	18.0	8	318.0	150	3436	11.0	70	1	plymouth satellite
3	16.0	8	304.0	150	3433	12.0	70	1	amc rebel sst
4	17.0	8	302.0	140	3449	10.5	70	1	ford torino
...	...	...	...	...	...	...	...	...	...
393	27.0	4	140.0	86	2790	15.6	82	1	ford mustang gl
394	44.0	4	97.0	52	2130	24.6	82	2	vw pickup
395	32.0	4	135.0	84	2295	11.6	82	1	dodge rampage
396	28.0	4	120.0	79	2625	18.6	82	1	ford ranger
397	31.0	4	119.0	82	2720	19.4	82	1	chevy s-10

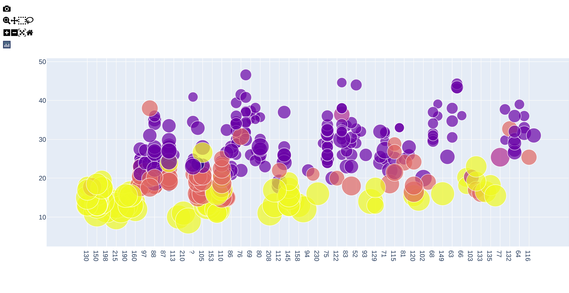
398 rows × 9 columns

In [3]:
```
data = go.Scatter(x = mpg_data_csv["horsepower"],
                  y = mpg_data_csv["mpg"],
                  text = mpg_data_csv["name"],
                  mode = "markers",
                  hovertemplate='HorsePower: %{x}<br>Miles Per Gallon: %{y}<br>Vechicle Name: %{text}',
                  name='A Dash<br>App',
                  marker = dict(size = mpg_data_csv["weight"]/100,
                                color = mpg_data_csv["cylinders"],
                                showscale = True),)
```

In [4]:
```
layout = go.Layout(title = "A Bubble Chart<br>in A Dash App",
                   xaxis = dict(title = 'Horsepower'),
                   yaxis = dict(title = 'Miles Per Gallon'),
                   hovermode='closest',
                   title_x = 0.5)
```

In [5]:
```
fig = go.Figure(data, layout)
```

In [6]:
```
pyo.iplot(fig)
```



In [7]:
```
temperature_data_1_of_santa_barbara_in_california_csv = pd.read_csv("2010SantaBarbaraCA.csv")
temperature_data_2_of_yuma_in_arizona_csv = pd.read_csv("2010YumaAZ.csv")
temperature_data_3_of_sitka_in_alaska_csv = pd.read_csv("2010SitkaAK.csv")
```

In [8]:
```
trace_1 = go.Heatmap(go.Heatmap(x = temperature_data_1_of_santa_barbara_in_california_csv["DAY"],
                     y = temperature_data_1_of_santa_barbara_in_california_csv["LST_TIME"],
                     z = temperature_data_1_of_santa_barbara_in_california_csv["T_HR_AVG"].values.tolist(),
                     zmin = 5,
                     zmax = 40,
                     colorbar=dict(title="Temperature"),
                     colorscale = "jet"))
```

In [9]:
```
trace_2 = go.Heatmap(go.Heatmap(x = temperature_data_2_of_yuma_in_arizona_csv["DAY"],
                     y = temperature_data_2_of_yuma_in_arizona_csv["LST_TIME"],
                     z = temperature_data_2_of_yuma_in_arizona_csv["T_HR_AVG"].values.tolist(),
                     zmin = 5,
                     zmax = 40,
                     colorbar=dict(title="Temperature"),
                     colorscale = "jet"))
```

In [10]:
```
trace_3 = go.Heatmap(go.Heatmap(x = temperature_data_3_of_sitka_in_alaska_csv["DAY"],
                     y = temperature_data_3_of_sitka_in_alaska_csv["LST_TIME"],
                     z = temperature_data_3_of_sitka_in_alaska_csv["T_HR_AVG"].values.tolist(),
                     zmin = 5,
                     zmax = 40,
                     colorbar=dict(title="Temperature"),
                     colorscale = "jet"))
```

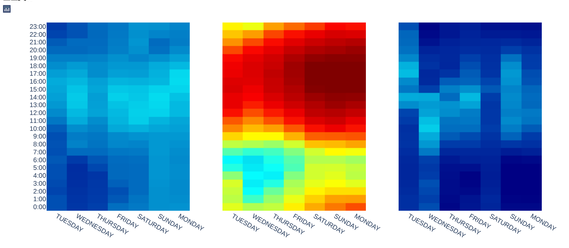
In [11]:
```
fig = subplots.make_subplots(1, 3,
                             subplot_titles = ["Santa Barbara in California", "Yuma in Arizona", "Sitka in Alaska"],
                             shared_yaxes = True)
```

In [12]:
```
fig['layout'].update(title = "Temperature of Diffrent Cities", title_x = 0.5)
```

Out[12]:
```
Layout({
    'annotations': [{'font': {'size': 16},
                     'showarrow': False,
                     'text': 'Santa Barbara in California',
                     'x': 0.14444444444444446,
                     'xanchor': 'center',
                     'xref': 'paper',
                     'y': 1.0,
                     'yanchor': 'bottom',
                     'yref': 'paper'},
                    {'font': {'size': 16},
                     'showarrow': False,
                     'text': 'Yuma in Arizona',
                     'x': 0.5,
                     'xanchor': 'center',
                     'xref': 'paper',
                     'y': 1.0,
                     'yanchor': 'bottom',
                     'yref': 'paper'},
                    {'font': {'size': 16},
                     'showarrow': False,
                     'text': 'Sitka in Alaska',
                     'x': 0.8555555555555556,
                     'xanchor': 'center',
                     'xref': 'paper',
                     'y': 1.0,
                     'yanchor': 'bottom',
                     'yref': 'paper'}],
    'template': '...',
    'title': {'text': 'Temperature of Diffrent Cities', 'x': 0.5},
    'xaxis': {'anchor': 'y', 'domain': [0.0, 0.2888888888888889]},
    'xaxis2': {'anchor': 'y2', 'domain': [0.35555555555555557, 0.6444444444444445]},
    'xaxis3': {'anchor': 'y3', 'domain': [0.7111111111111111, 1.0]},
    'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0]},
    'yaxis2': {'anchor': 'x2', 'domain': [0.0, 1.0], 'matches': 'y', 'showticklabels': False},
    'yaxis3': {'anchor': 'x3', 'domain': [0.0, 1.0], 'matches': 'y', 'showticklabels': False}
})
```

In [13]:
```
fig.append_trace(trace_1, 1, 1)
fig.append_trace(trace_2, 1, 2)
fig.append_trace(trace_3, 1, 3)
```

In [14]:
```
fig.show()
```

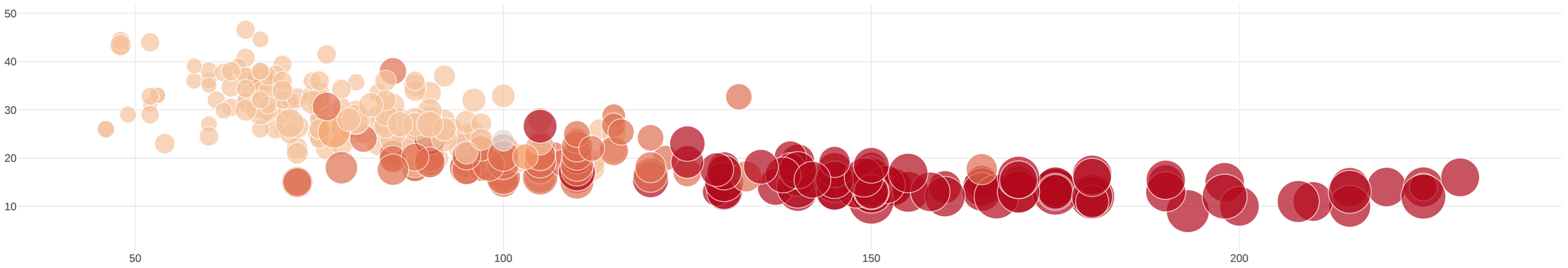


In [15]:
```
heading_style = {'font-size' : '50px',
'line-height' : '40px',
'margin' : '1em 0 .6em 0',
'font-weight' : 'normal',
'color' : 'white',
'font-family' : 'Hammersmith One',
'text-shadow' : '0 1px 0 rgba(0,0,0,0.4)',
'position' : 'relative',
'color' : '#6cf',
'text-align' : 'center',}
```

In [16]:
```
app = dash.Dash()
app.layout = html.Div(children = [html.H1("The Below is a Scatter Plot", style = heading_style),
                                  dcc.Graph(id = "ScatterPlot",
                                            figure = {'data' : [data],
                                                      'layout' : layout}),
                                  html.H1("The Below is a HeatMap", style = heading_style),
                                  dcc.Graph(id = "HeatMap",
                                            figure = fig)])
app.run_server()
```

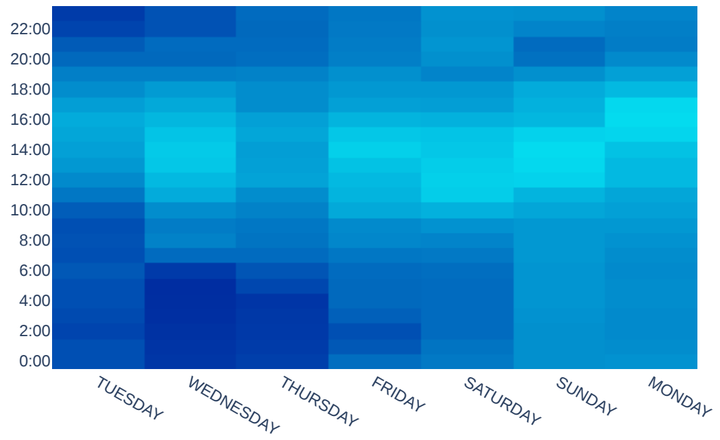
```
Dash is running on http://127.0.0.1:8050/

 * Serving Flask app "__main__" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:8050/ (Press CTRL+C to quit)
127.0.0.1 - - [07/Jul/2021 16:14:01] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [07/Jul/2021 16:14:01] "GET /_dash-dependencies HTTP/1.1" 200 -
127.0.0.1 - - [07/Jul/2021 16:14:01] "GET /_dash-layout HTTP/1.1" 200 -
```

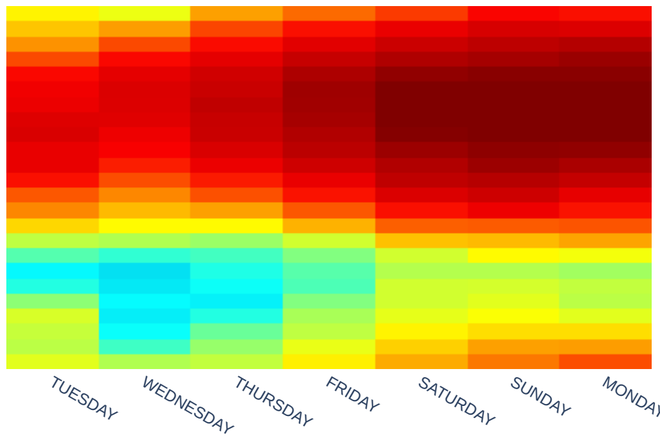
The Below is a Scatter Plot

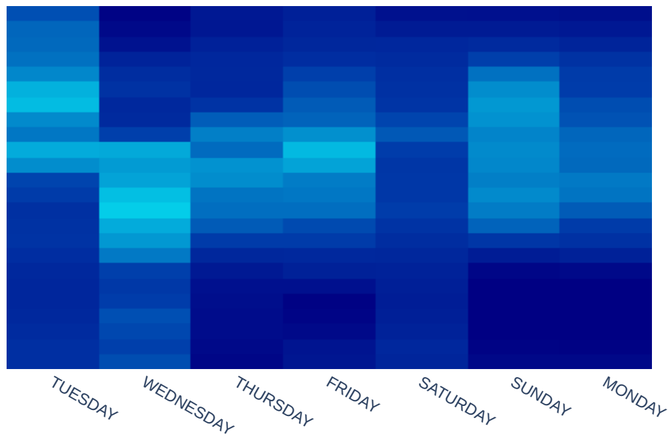


A Bubble Chart
in A Dash App

The Below is a HeatMap







Temperature of Diffrent Cities

Santa Barbara in California

Yuma in Arizona

Sitka in Alaska