

Data Analysis

Ram - Battery Power - Price Range

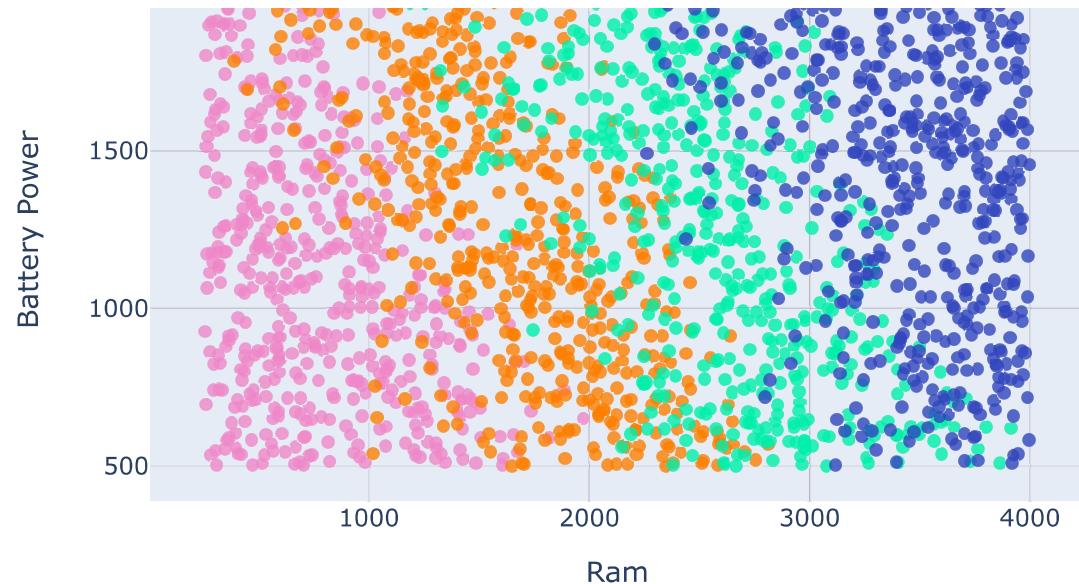
In [13]:

```

1 df_pr_0 = df[df.price_range == 0]
2 df_pr_1 = df[df.price_range == 1]
3 df_pr_2 = df[df.price_range == 2]
4 df_pr_3 = df[df.price_range == 3]
5
6 trace1 = go.Scatter(
7     y = df_pr_0.battery_power,
8     x = df_pr_0.ram,
9     mode = "markers",
10    name = "Price Range: 0",
11    marker = dict(color = 'rgba(240, 136, 200, 0.8)'),
12    text= df_pr_0.price_range)
13
14 trace2 = go.Scatter(
15     y = df_pr_1.battery_power,
16     x = df_pr_1.ram,
17     mode = "markers",
18     name = "Price Range: 1",
19     marker = dict(color = 'rgba(255, 128, 2, 0.8)'),
20     text= df_pr_1.price_range)
21
22 trace3 = go.Scatter(
23     y = df_pr_2.battery_power,
24     x = df_pr_2.ram,
25     mode = "markers",
26     name = "Price Range: 2",
27     marker = dict(color = 'rgba(0, 240, 170, 0.8)'),
28     text= df_pr_2.price_range)
29
30 trace4 = go.Scatter(
31     y = df_pr_3.battery_power,
32     x = df_pr_3.ram,
33     mode = "markers",
34     name = "Price Range: 3",
35     marker = dict(color = 'rgba(50, 70, 190, 0.8)'),
36     text= df_pr_3.price_range)
37
38
39 data = [trace1, trace2, trace3, trace4]
40
41 layout = dict(title = 'Ram - Battery Power - Price Range',
42                 xaxis= dict(title= 'Ram',
43                             ticklen= 5,zeroline= False),
44                 yaxis= dict(title= 'Battery Power',
45                             ticklen= 5,zeroline= False),
46                 autosize=False,
47                 width=700,
48                 height=450, )
49 fig = dict(data = data, layout = layout)
50
51 iplot(fig)

```

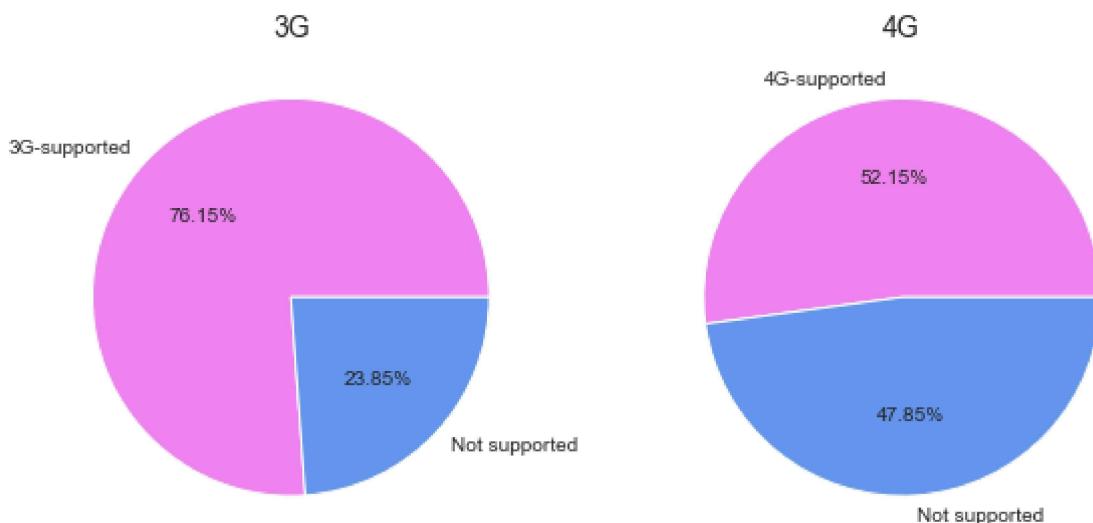
Ram - Battery Power - Price Range



3G - 4G

In [14]:

```
1 values_3g = df["three_g"].value_counts()
2 values_4g = df["four_g"].value_counts()
3
4 labels4g = ["4G-supported", 'Not supported']
5 labels3g = ["3G-supported", 'Not supported']
6
7 colors = ["violet", "cornflowerblue"]
8
9 fig = plt.figure(figsize=(10,10))
10 ax1 = plt.subplot2grid((2,2),(0,0))
11 plt.pie(x= values_3g, autopct="% .2f%%",
12         labels=labels3g, pctdistance=0.6,
13         colors = colors)
14 plt.title('3G', size = 14)
15
16 ax1 = plt.subplot2grid((2,2), (0, 1))
17 plt.pie(x =values_4g,autopct="% .2f%%",
18         labels=labels4g, pctdistance=0.6,
19         colors = colors)
20 plt.title('4G', size = 14)
21 plt.show()
```

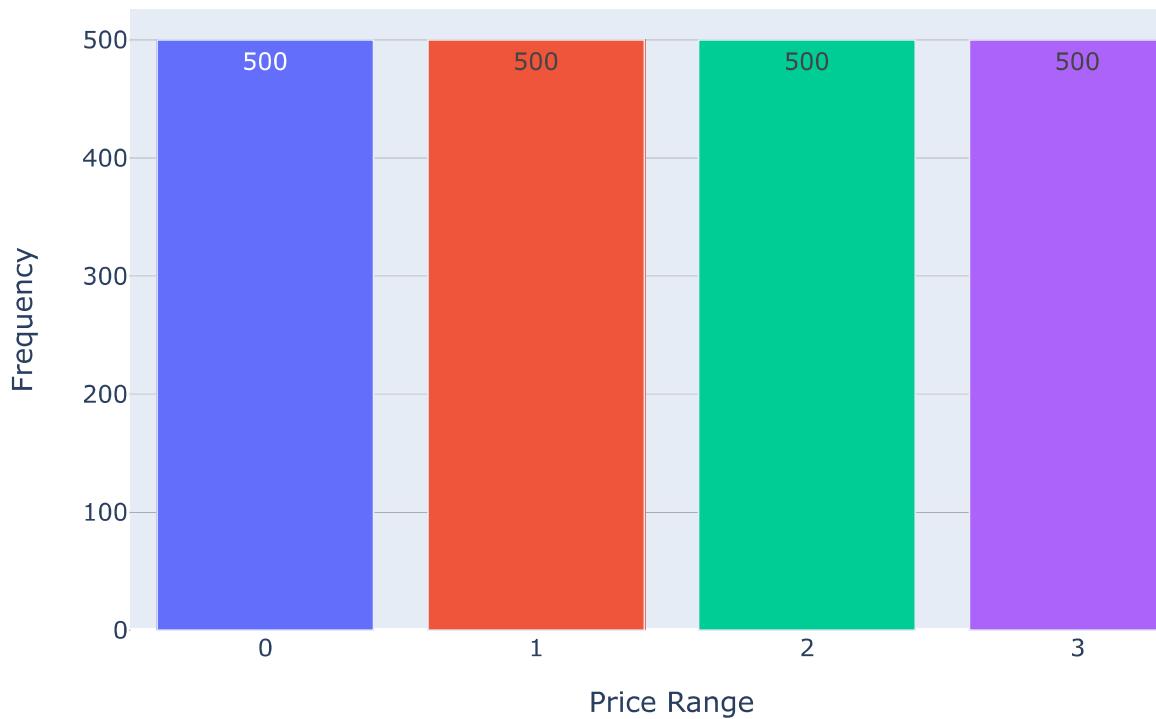


Price Range

In [15]:

```
1 price_range = df["price_range"].value_counts()
2
3 fig = px.bar(price_range, y = 'price_range', text = 'price_range',
4
5             color = ["Price Range: 0","Price Range: 1",
6             "Price Range: 2","Price Range: 3"])
7
8
9 fig.update_layout(title = "Price Range", showlegend = False,
10                  xaxis = dict(zeroline = False, tickmode = 'linear', tick0 = 0, dtick =
11                  autosize=False,
12                  width=700,
13                  height=450,
14                  xaxis_title_text = 'Price Range',
15                  yaxis_title_text = 'Frequency')
16
17 fig
```

Price Range

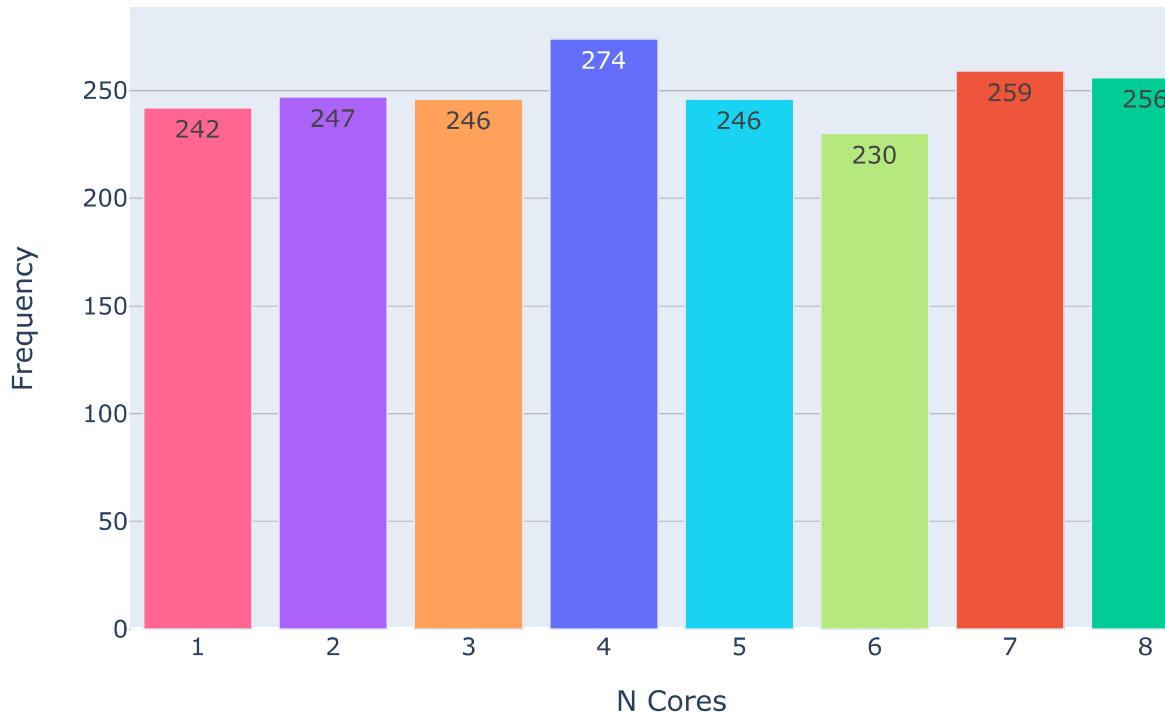


N Cores

In [16]:

```
1 n_cores = df["n_cores"].value_counts()
2
3 fig = px.bar(n_cores, y = 'n_cores', text = 'n_cores',
4
5                 color = ["4","7","8","2","3","5","1","6"])
6
7 fig.update_layout(title = "N Cores", showlegend = False,
8                     xaxis = dict(zeroLine = False, tickmode = 'linear', tick0 = 0, dtick =
9                     autosize=False,
10                    width=700,
11                    height=450,
12                    xaxis_title_text = 'N Cores',
13                    yaxis_title_text = 'Frequency')
14
15 fig
```

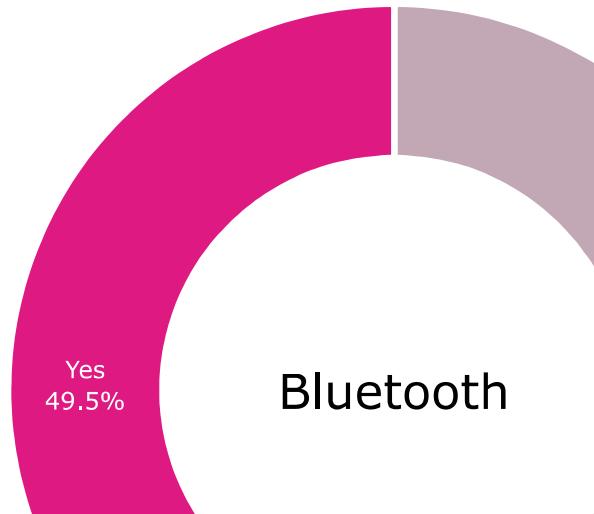
N Cores



Bluetooth

In [17]:

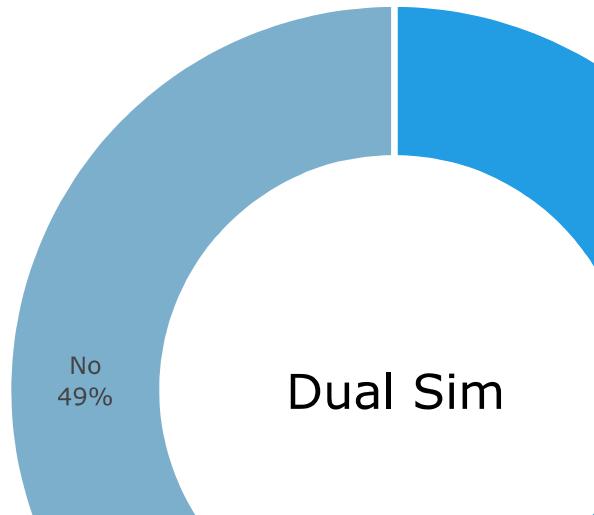
```
1 fig = px.pie(df['blue'].value_counts().reset_index(), values = 'blue',
2                 names = ['No', 'Yes'])
3 fig.update_traces(textposition = 'inside',
4                     textinfo = 'percent + label',
5                     hole = 0.6,
6                     marker = dict(colors = ['#C2A7B5', '#DE1A82'],
7                                   line = dict(color = 'white', width = 3)))
8
9 fig.update_layout(annotations = [dict(text = 'Bluetooth',
10                                         x = 0.5, y = 0.5,
11                                         font_size = 24, showarrow = False,
12                                         font_family = 'Verdana',
13                                         font_color = 'black')],  
14                                         showlegend = False)
15
16 fig.show()
```



Dual Sim

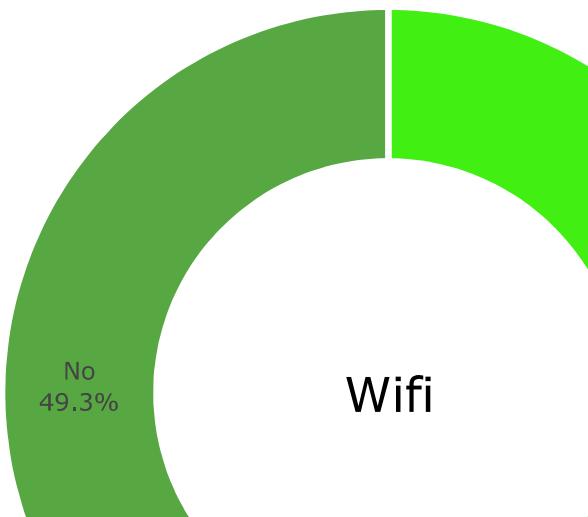
In [18]:

```
1 fig = px.pie(df['dual_sim'].value_counts().reset_index(), values = 'dual_sim',
2                 names = ['Yes', 'No'])
3 fig.update_traces(textposition = 'inside',
4                     textinfo = 'percent + label',
5                     hole = 0.6,
6                     marker = dict(colors = ['#229CE2', '#7CAFCC'],
7                                   line = dict(color = 'white', width = 3)))
8
9 fig.update_layout(annotations = [dict(text = 'Dual Sim',
10                                         x = 0.5, y = 0.5,
11                                         font_size = 24, showarrow = False,
12                                         font_family = 'Verdana',
13                                         font_color = 'black')],  
14                                         showlegend = False)
15
16 fig.show()
```

**Wifi**

In [19]:

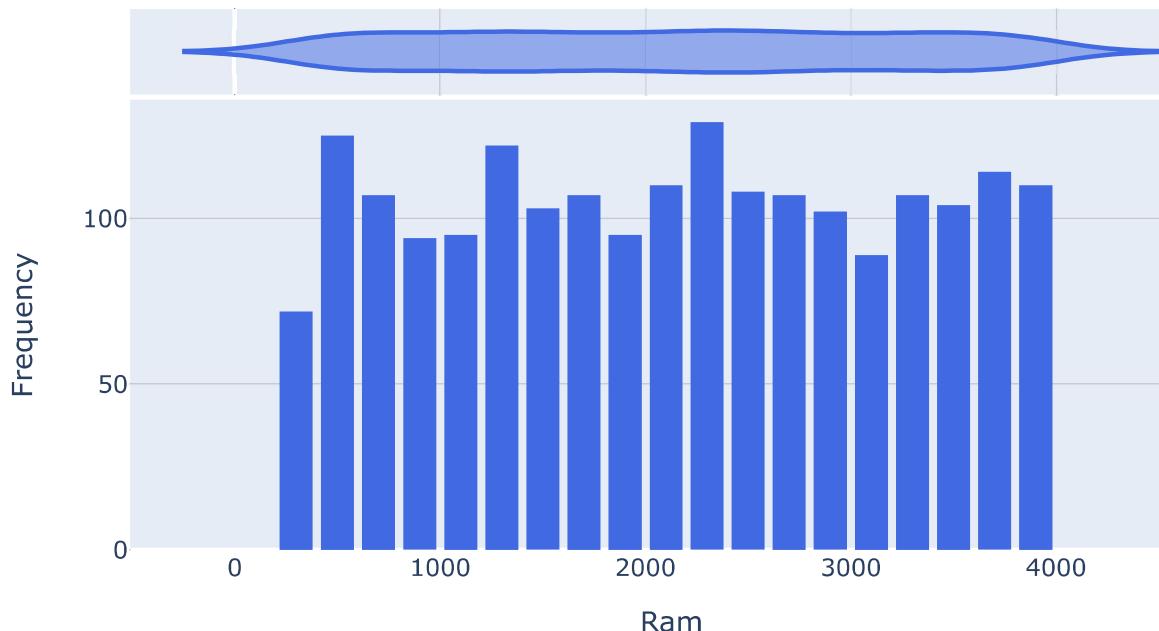
```
1 fig = px.pie(df['wifi'].value_counts().reset_index(), values = 'wifi',
2                 names = ['Yes', 'No'])
3 fig.update_traces(textposition = 'inside',
4                     textinfo = 'percent + label',
5                     hole = 0.6,
6                     marker = dict(colors = ['#41EF13', '#57A742'],
7                                   line = dict(color = 'white', width = 3)))
8
9 fig.update_layout(annotations = [dict(text = 'Wifi',
10                                         x = 0.5, y = 0.5,
11                                         font_size = 24, showarrow = False,
12                                         font_family = 'Verdana',
13                                         font_color = 'black')],  
14                                         showlegend = False)
15
16 fig.show()
17
```

**RAM**

In [20]:

```
1 fig = px.histogram(df,x = 'ram',
2                     title = 'Ram',
3                     marginal = 'violin',
4                     color_discrete_sequence = ['royalblue'])
5 fig.update_layout(
6     xaxis_title_text = 'Ram',
7     yaxis_title_text = 'Frequency',
8     bargap = 0.2, showlegend = False,
9     autosize=False,
10    width=700,
11    height=450)
12
13 iplot(fig)
```

Ram

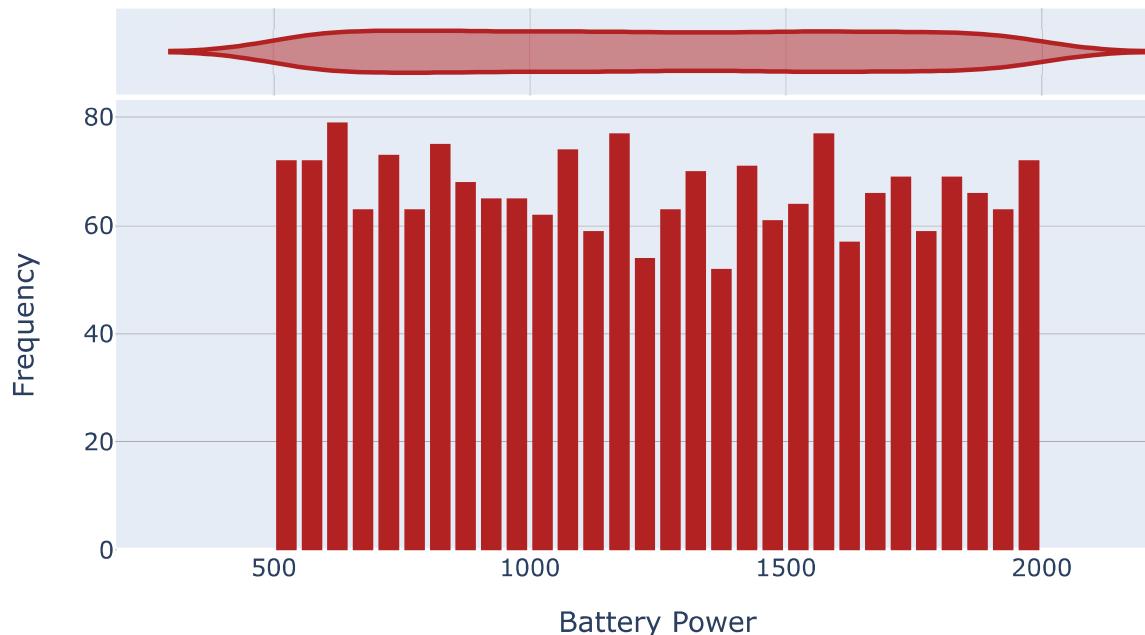


Battery Power

In [21]:

```
1 fig = px.histogram(df,x = 'battery_power',
2                     title = 'Battery Power',
3                     marginal = 'violin',
4                     color_discrete_sequence = ['firebrick'])
5 fig.update_layout(
6     xaxis_title_text = 'Battery Power',
7     yaxis_title_text = 'Frequency',
8     bargap = 0.2, showlegend = False,
9     autosize=False,
10    width=700,
11    height=450)
12
13 iplot(fig)
14
```

Battery Power



Ram - Price Range

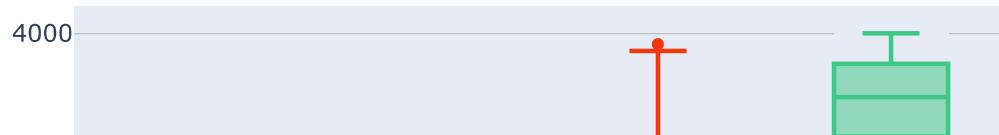
In [22]:

```

1 df_pr_0 = df[df.price_range == 0]
2 df_pr_1 = df[df.price_range == 1]
3 df_pr_2 = df[df.price_range == 2]
4 df_pr_3 = df[df.price_range == 3]
5
6 trace0 = go.Box(
7     y=df_pr_0.ram,
8     name = 'Price Range: 0',
9     marker = dict(
10         color = 'rgb(223, 240, 0)',
11     )
12 )
13 trace1 = go.Box(
14     y=df_pr_1.ram,
15     name = 'Price Range: 1',
16     marker = dict(
17         color = 'rgb(10, 110, 220)',
18     )
19 )
20 trace2 = go.Box(
21     y=df_pr_2.ram,
22     name = 'Price Range: 2',
23     marker = dict(
24         color = 'rgb(242, 54, 14)',
25     )
26 )
27 trace3 = go.Box(
28     y=df_pr_3.ram,
29     name = 'Price Range: 3',
30     marker = dict(
31         color = 'rgb(60, 200, 135)',
32     )
33 )
34
35 data = [trace0, trace1, trace2, trace3]
36
37 layout = dict(title = 'Ram - Price Range',
38                 xaxis= dict(title= 'Price Range',
39                             ticklen= 5,zeroline= False),
40                 yaxis= dict(title= 'Ram',
41                             ticklen= 5,zeroline= False),
42                 autosize=False,
43                 width=700,
44                 height=450)
45
46 fig = dict(data = data, layout = layout)
47
48 iplot(fig)

```

Ram - Price Range





Battery Power - Price Range

In [23]:

```

1 df_pr_0 = df[df.price_range == 0]
2 df_pr_1 = df[df.price_range == 1]
3 df_pr_2 = df[df.price_range == 2]
4 df_pr_3 = df[df.price_range == 3]
5
6 trace0 = go.Box(
7     y=df_pr_0.battery_power,
8     name = 'Price Range: 0',
9     marker = dict(
10         color = 'rgb(223, 240, 0)',
11     )
12 )
13 trace1 = go.Box(
14     y=df_pr_1.battery_power,
15     name = 'Price Range: 1',
16     marker = dict(
17         color = 'rgb(10, 110, 220)',
18     )
19 )
20 trace2 = go.Box(
21     y=df_pr_2.battery_power,
22     name = 'Price Range: 2',
23     marker = dict(
24         color = 'rgb(242, 54, 14)',
25     )
26 )
27 trace3 = go.Box(
28     y=df_pr_3.battery_power,
29     name = 'Price Range: 3',
30     marker = dict(
31         color = 'rgb(60, 200, 135)',
32     )
33 )
34
35 data = [trace0, trace1, trace2, trace3]
36
37 layout = dict(title = 'Battery Power - Price Range',
38                 xaxis= dict(title= 'Price Range',
39                             ticklen= 5,zeroline= False),
40                 yaxis= dict(title= 'Battery Power',
41                             ticklen= 5,zeroline= False),
42                 autosize=False,
43                 width=700,
44                 height=450)
45
46 fig = dict(data = data, layout = layout)
47
48 iplot(fig)

```

Battery Power - Price Range



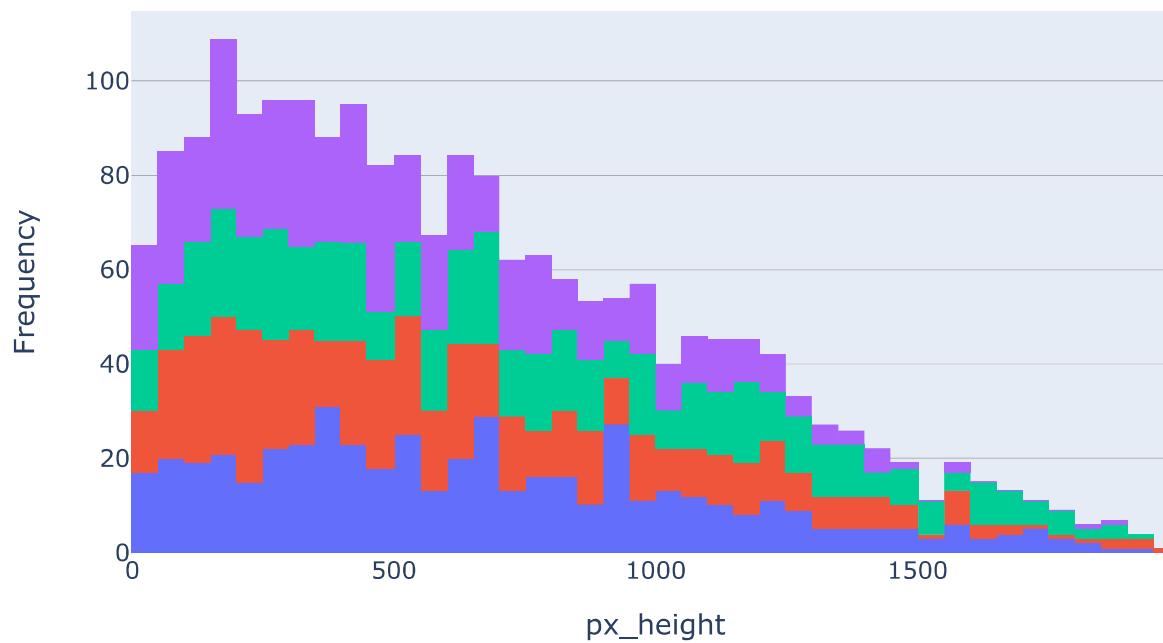


Phone Height - Price Range

In [24]:

```
1 fig = px.histogram(df, x = 'px_height',
2                     color = 'price_range',
3                     title = "Phone Height - Price Range")
4 fig.update_layout(
5     xaxis_title_text = 'px_height',
6     yaxis_title_text = 'Frequency', autosize=False,
7     width=700,
8     height=450)
9 fig
```

Phone Height - Price Range

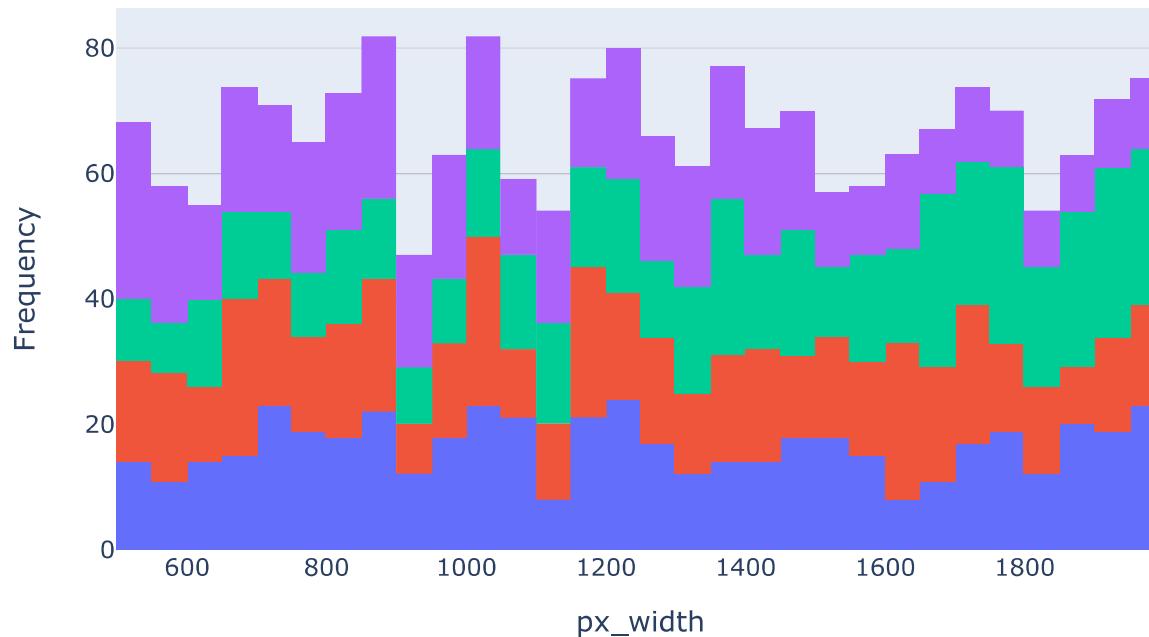


Phone Width - Price Range

In [25]:

```
1 fig = px.histogram(df, x = 'px_width',
2                     color = 'price_range',
3                     title = "Phone Width - Price Range")
4 fig.update_layout(
5     xaxis_title_text = 'px_width',
6     yaxis_title_text = 'Frequency', autosize=False,
7     width=700,
8     height=450)
9 fig
```

Phone Width - Price Range



Correlation Map