CampusX

In [2]:
```
import numpy as np
import pandas as pd
import plotly.offline as pyo
import plotly.graph_objs as go
```

In [3]:
```
match = pd.read_csv('matches.csv')
delivery = pd.read_csv('deliveries.csv')

ipl = delivery.merge(match, left_on = 'match_id', right_on = 'id')
ipl.head(2)
```

Out[3]:

	match_id	inning	batting_team	bowling_team	over	ball	batsman	non_striker	bowler	is_super_ov
0	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	1	DA Warner	S Dhawan	TS Mills	
1	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	2	DA Warner	S Dhawan	TS Mills	

2 rows × 39 columns

In [4]:
```
top50 = ipl.groupby('batsman')['batsman_runs'].sum().sort_values(ascending = Fa
new_ipl = ipl[ipl['batsman'].isin(top50)]
```

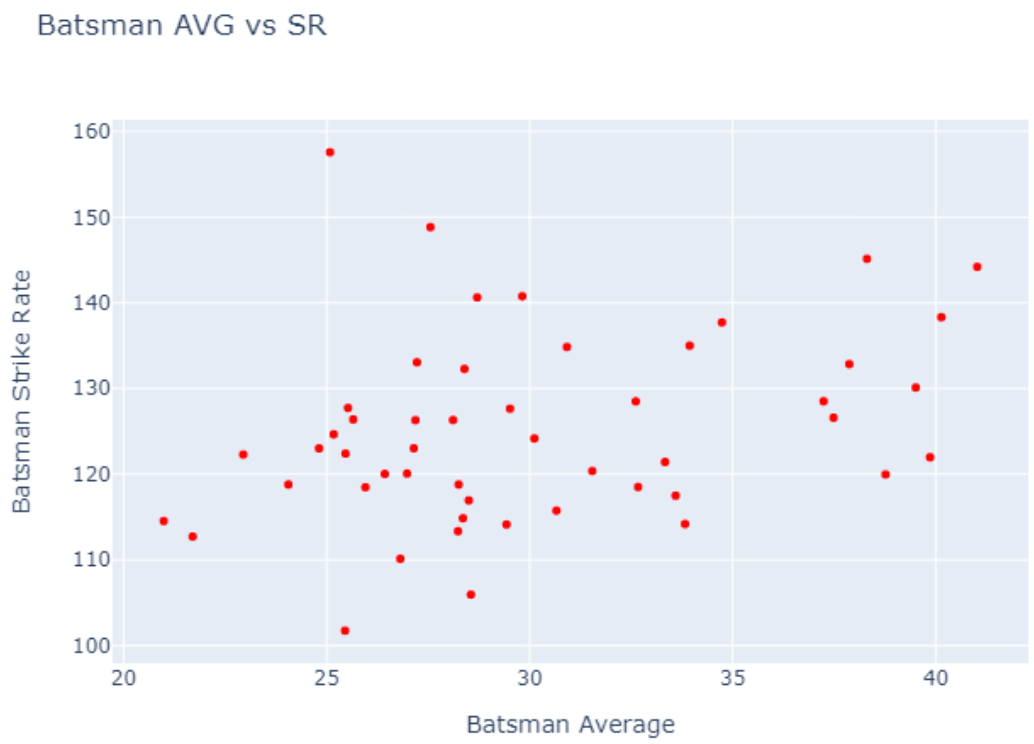
In [5]:
```
runs = new_ipl.groupby('batsman')['batsman_runs'].sum()
balls = new_ipl.groupby('batsman')['batsman_runs'].count()

sr = (runs/balls)*100
sr = sr.reset_index()
```

In [6]:
```
out = ipl[ipl['player_dismissed'].isin(top50)]
nouts = out['player_dismissed'].value_counts()

avg = runs/nouts
```

```
avg = avg.reset_index()
avg.rename(columns = {'index': 'batsman', 0: 'avg'}, inplace = True)
avg = avg.merge(sr, on = 'batsman')
```

Scatter Plot

In [36]:
```
trace = go.Scatter(x = avg['avg'], y = avg['batsman_runs'],
                   mode = 'markers', text = avg['batsman'],
                   marker = {'color': 'red', 'size': 5})
data = [trace]
layout = go.Layout(title = 'Batsman AVG vs SR',
                   xaxis = {'title': 'Batsman Average'},
                   yaxis = {'title': 'Batsman Strike Rate'})
fig = go.Figure(data = data, layout = layout)
pyo.plot(fig, filename = 'ipl_AVG_vs_SR.html')
fig.write_image('scatter_plot.png')
```

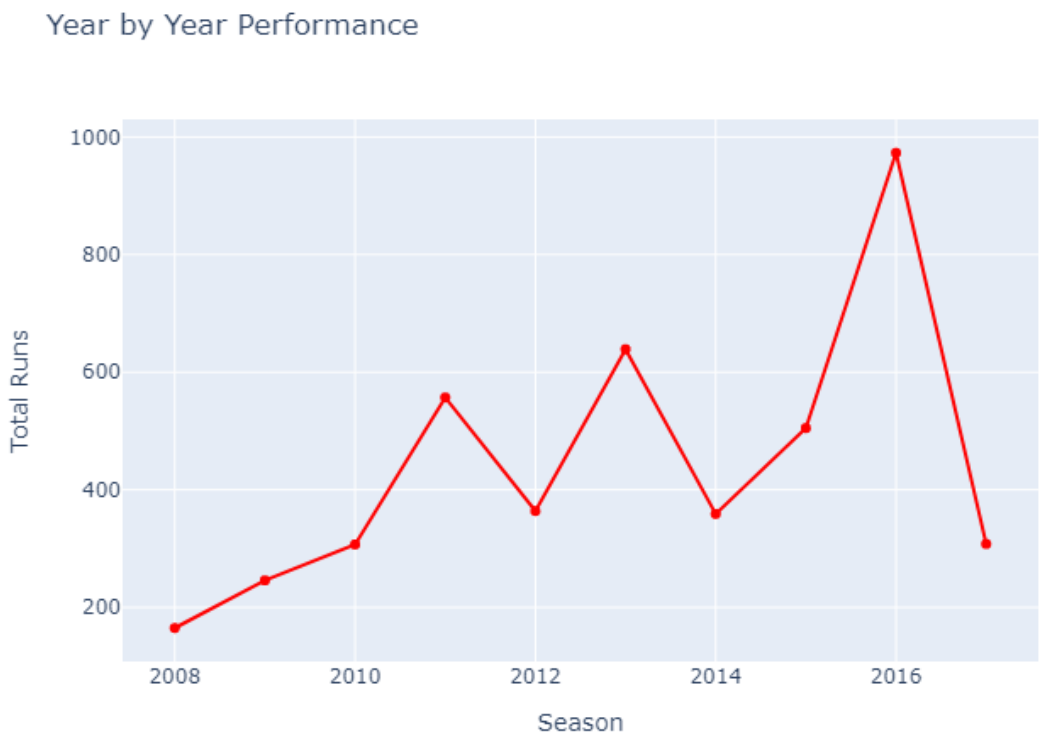


Line Chart

In [37]:
```
single = ipl[ipl['batsman']=='V Kohli']
performance = single.groupby('season')['batsman_runs'].sum().reset_index()
```

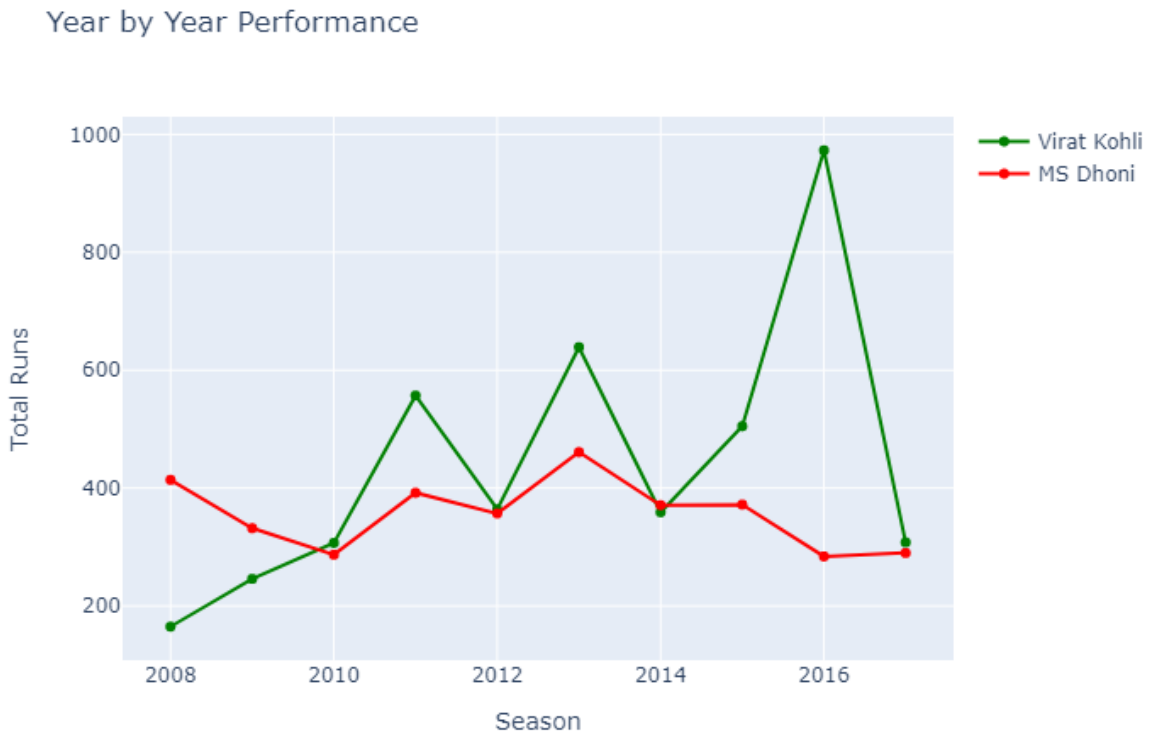
In [38]:
```
trace = go.Scatter(x = performance['season'], y = performance['batsman_runs'],
                   mode = 'lines + markers',
                   marker = {'color': 'red'})
data = [trace]
layout = go.Layout(title = 'Year by Year Performance',
                     xaxis = {'title': 'Season'},
                     yaxis = {'title': 'Total Runs'})
fig = go.Figure(data = data, layout = layout)
```

```
pyo.plot(fig, filename = 'ipl_Years_Performance.html')
fig.write_image('line_chart.png')
```



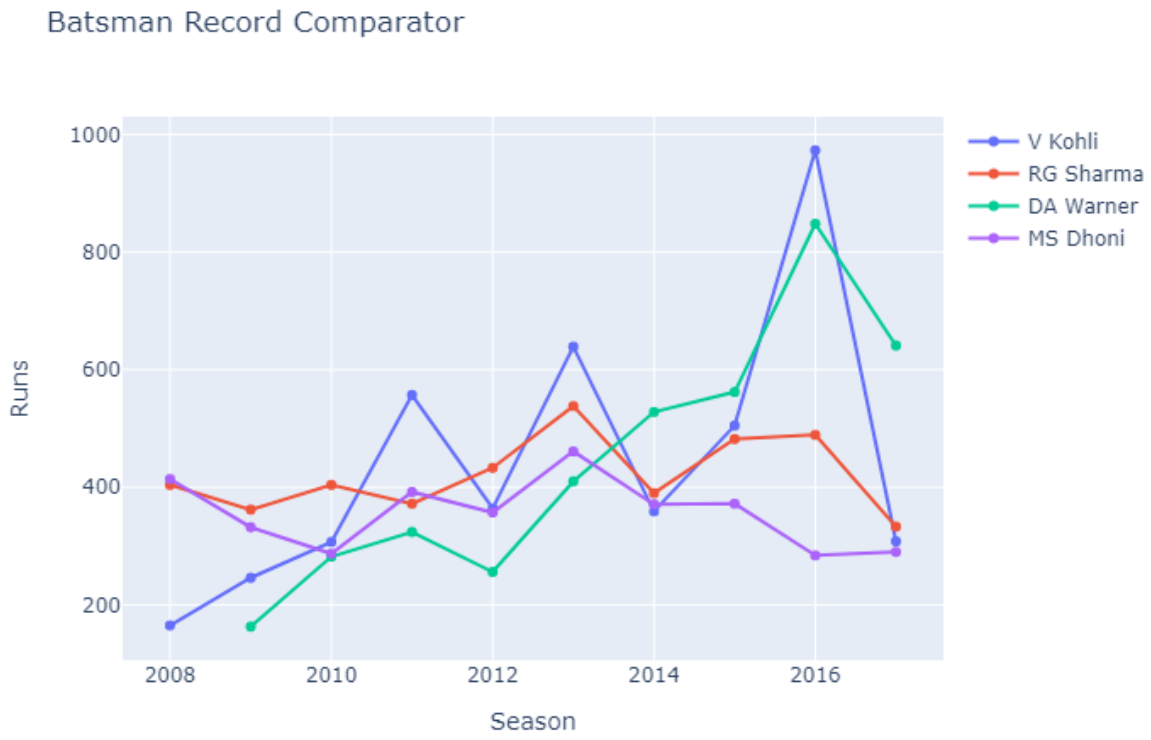
In [39]:

```
single = ipl[ipl['batsman']=='V Kohli']
performance = single.groupby('season')['batsman_runs'].sum().reset_index()

single_1 = ipl[ipl['batsman']=='MS Dhoni']
performance_1 = single_1.groupby('season')['batsman_runs'].sum().reset_index()

trace = go.Scatter(x = performance['season'], y = performance['batsman_runs'],
                   mode = 'lines + markers',
                   marker = {'color': 'green'}, name = 'Virat Kohli')

trace_1 = go.Scatter(x = performance_1['season'], y = performance_1['batsman_ru
                   mode = 'lines + markers',
                   marker = {'color': 'red'}, name = 'MS Dhoni')

data = [trace, trace_1]

layout = go.Layout(title = 'Year by Year Performance',
                     xaxis = {'title': 'Season'},
                     yaxis = {'title': 'Total Runs'})

fig = go.Figure(data = data, layout = layout)
pyo.plot(fig, filename = 'ipl_VK_vs_MD.html')
fig.write_image('line_chart_1.png')
```



In [43]:
```
def batsman_comp(*name):
    data = []
    for i in name:
        single = ipl[ipl['batsman']== i]
        performance = single.groupby('season')['batsman_runs'].sum().reset_inde

        trace = go.Scatter(x = performance['season'], y = performance['batsman_
                           mode = 'lines + markers', name = i)

        data.append(trace)

    layout = go.Layout(title = 'Batsman Record Comparator',
                       xaxis = {'title': 'Season'},
                       yaxis = {'title': 'Runs'})

    fig = go.Figure(data = data, layout = layout)
    pyo.plot(fig, filename = 'ipl_Year_by_Year.html')
    fig.write_image('line_chart_2.png')
```

In [44]:
```
batsman_comp('V Kohli', 'RG Sharma', 'DA Warner', 'MS Dhoni')
```

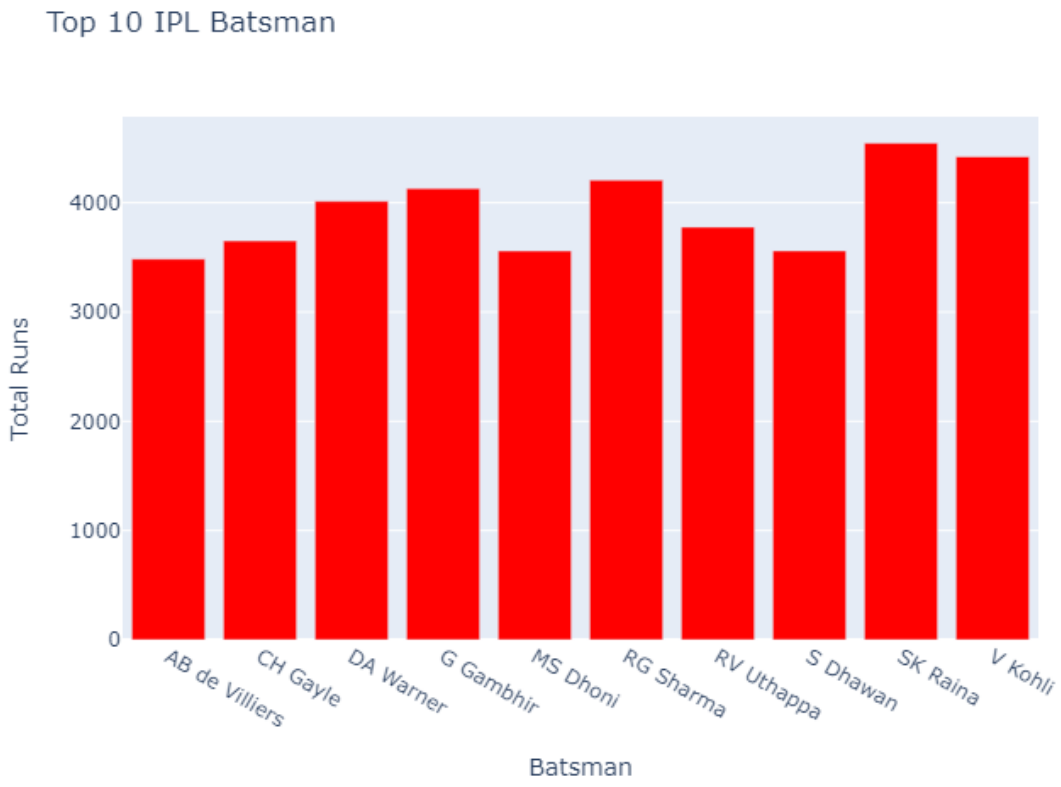


Bar Plot

In [45]:
```
top10 = ipl.groupby('batsman')['batsman_runs'].sum().sort_values(ascending = Fa
top10_df = ipl[ipl['batsman'].isin(top10)]
```

In [46]:
```
top10_score = top10_df.groupby('batsman')['batsman_runs'].sum().reset_index()
```

In [47]:
```
trace = go.Bar(x = top10_score['batsman'], y = top10_score['batsman_runs'], mar
data = [trace]
layout = go.Layout(title = 'Top 10 IPL Batsman',
                   xaxis = {'title': 'Batsman'},
                   yaxis = {'title': 'Total Runs'})

fig = go.Figure(data = data, layout = layout)
pyo.plot(fig, filename = 'Top10 IPL Batsman.html')
fig.write_image('bar_plot.png')
```

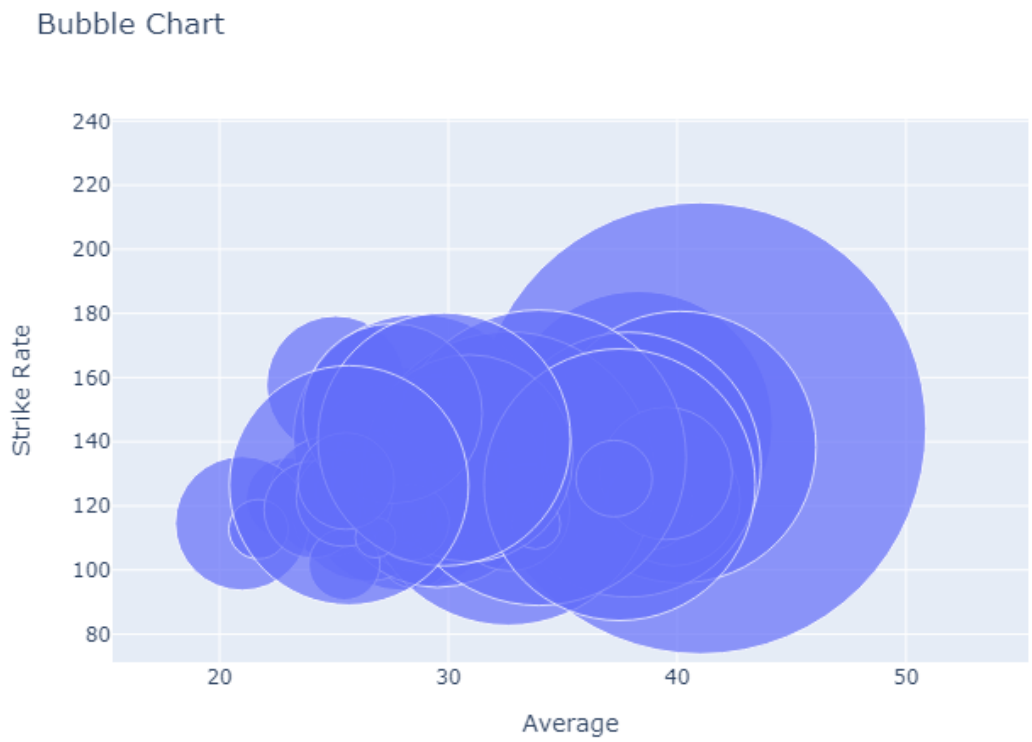


In [48]:
```
iw = top10_df.groupby(['batsman', 'inning'])['batsman_runs'].sum().reset_index(
mask = iw['inning']== 1
mask_2 = iw['inning']== 2
one = iw[mask]
two = iw[mask_2]


one.rename(columns = {'batsman_runs':'1st Innings'}, inplace = True)
two.rename(columns = {'batsman_runs':'2nd Innings'}, inplace = True)

final = one.merge(two, on = 'batsman')[['batsman', '1st Innings', '2nd Innings
```

In [49]:
```
trace_1 = go.Bar(x = final['batsman'], y = final['1st Innings'], name = '1st in
                 marker = {'color': 'red'})
trace_2 = go.Bar(x = final['batsman'], y = final['2nd Innings'], name = '2nd in
                 marker = {'color': 'green'})

data = [trace_1, trace_2]
layout = go.Layout(title = 'Inning wise Scores',
                   xaxis = {'title': 'Batsman'},
                   yaxis = {'title': 'Runs'})

fig = go.Figure(data = data, layout = layout)
pyo.plot(fig, filename = 'ipl_Year wise Scores')
fig.write_image('bar_plot_1.png')
```

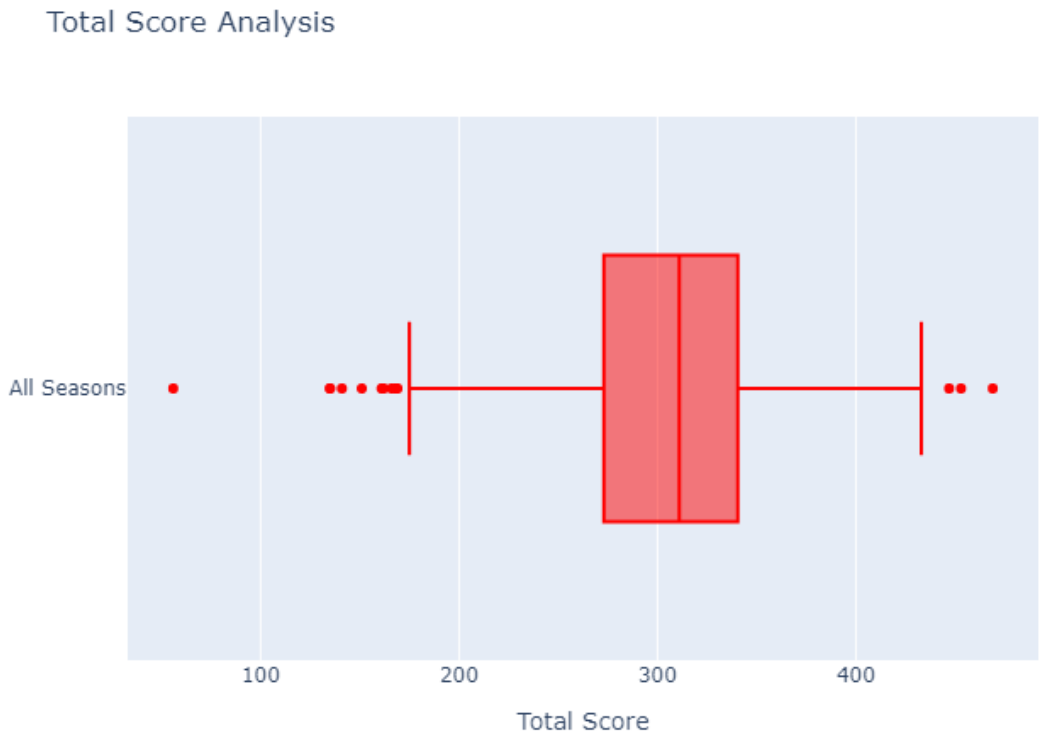
Bubble Plot

In [50]:
```
new_ipl = new_ipl[new_ipl['batsman_runs']== 6]
six = new_ipl.groupby('batsman')['batsman_runs'].count().reset_index()

x = avg.merge(six, on = 'batsman')
x

trace = go.Scatter(x = x['avg'], y = x['batsman_runs_x'], mode = 'markers',
                   marker = {'size': x['batsman_runs_y']}) #name = x['batsman']

data = [trace]
layout = go.Layout(title = 'Bubble Chart',
                   xaxis = {'title': 'Average'},
                   yaxis = {'title': 'Strike Rate'})

fig = go.Figure(data = data, layout = layout)
pyo.plot(fig, filename = 'ipl_Average_Strike Rate.html')
fig.write_image('bubble_plot.png')
```



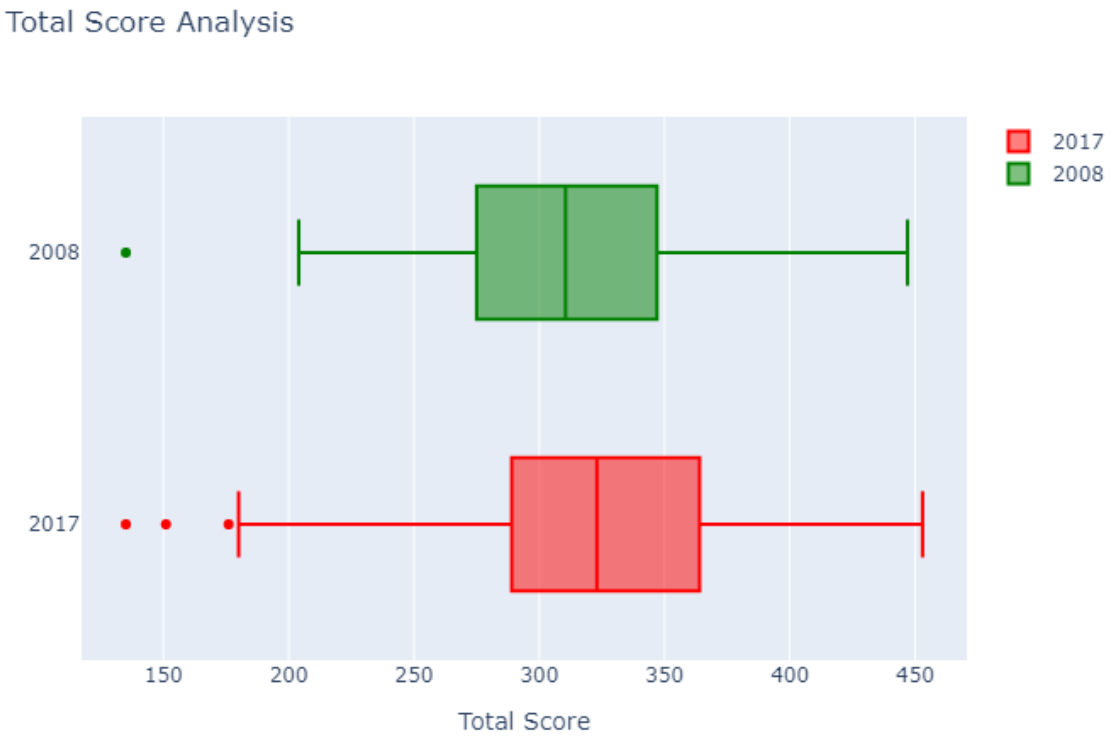
Box Plot

In [51]:
```
match_agg = delivery.groupby(['match_id'])['total_runs'].sum().reset_index()
season_wise = match_agg.merge(match, left_on = 'match_id', right_on = 'id')[['r
# season_wise

trace = go.Box(x = season_wise['total_runs'], name = 'All Seasons', marker = {
data = [trace]
layout = go.Layout(title = 'Total Score Analysis',
                   xaxis = {'title': 'Total Score'})

fig = go.Figure(data = data, layout = layout)
pyo.plot(fig, filename = 'ipl_Total Score Analysis.html')
fig.write_image('box_plot.png')
```



In [52]:

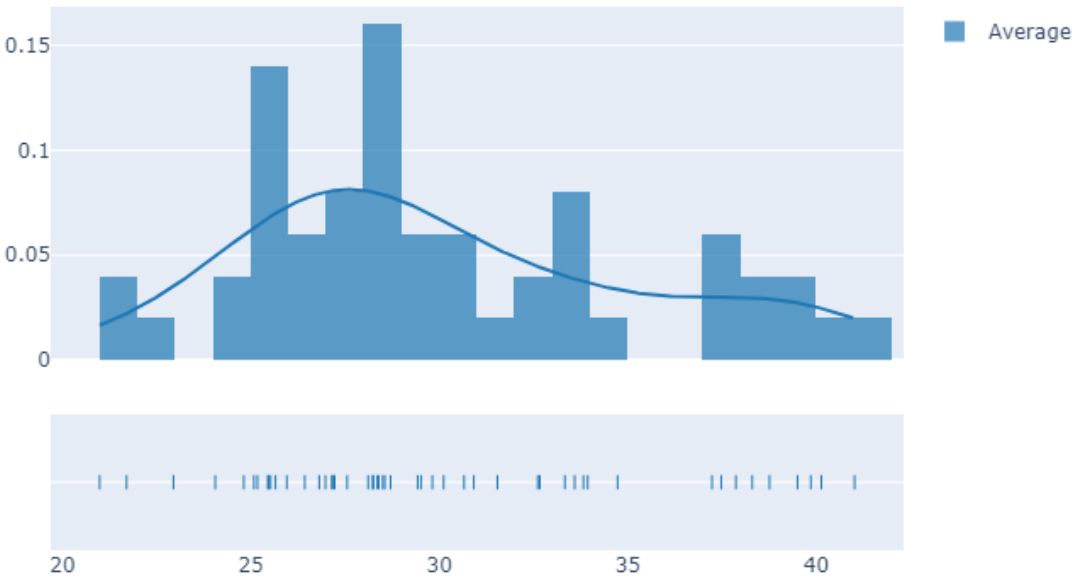
```
match_agg = delivery.groupby(['match_id'])['total_runs'].sum().reset_index()
season_wise = match_agg.merge(match, left_on = 'match_id', right_on = 'id')[['r
# season_wise

trace_1 = go.Box(x = season_wise[season_wise['season']== 2017]['total_runs'], r
trace_2 = go.Box(x = season_wise[season_wise['season']== 2008]['total_runs'], r

data = [trace_1, trace_2]
layout = go.Layout(title = 'Total Score Analysis',
                   xaxis = {'title': 'Total Score'})

fig = go.Figure(data = data, layout = layout)
pyo.plot(fig, filename = 'ipl_Total Score Analysis.html')
fig.write_image('box_plot_1.png')
```



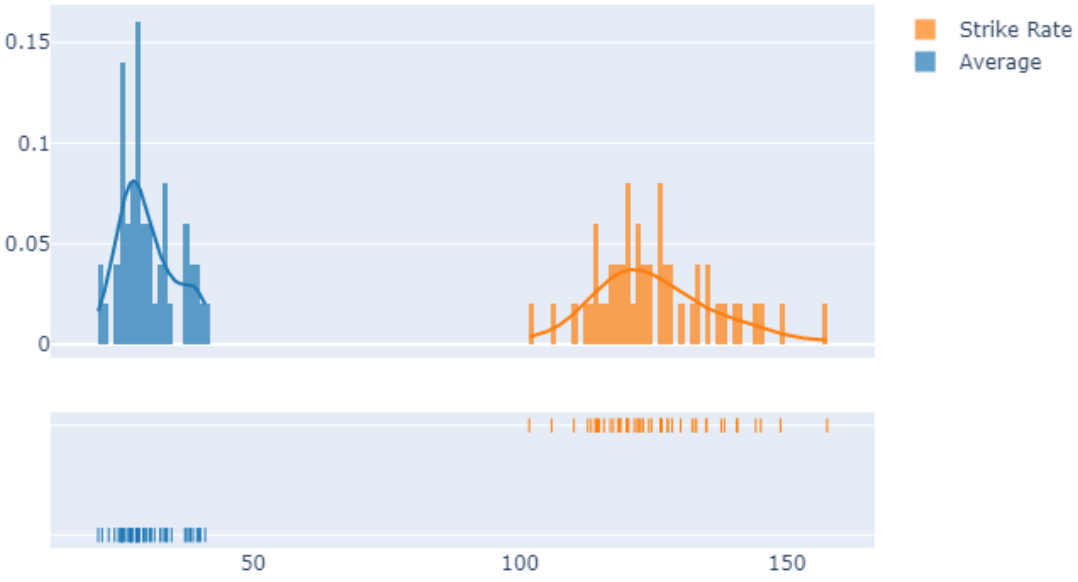
Dist Plot

In [53]:
```
import plotly.figure_factory as ff
```

In [55]:
```
hist_data = [avg['avg']]
group_labels = ['Average']

fig = ff.create_distplot(hist_data, group_labels)
pyo.plot(fig, filename = 'Average.html')
fig.write_image('dist_plot.png')
```



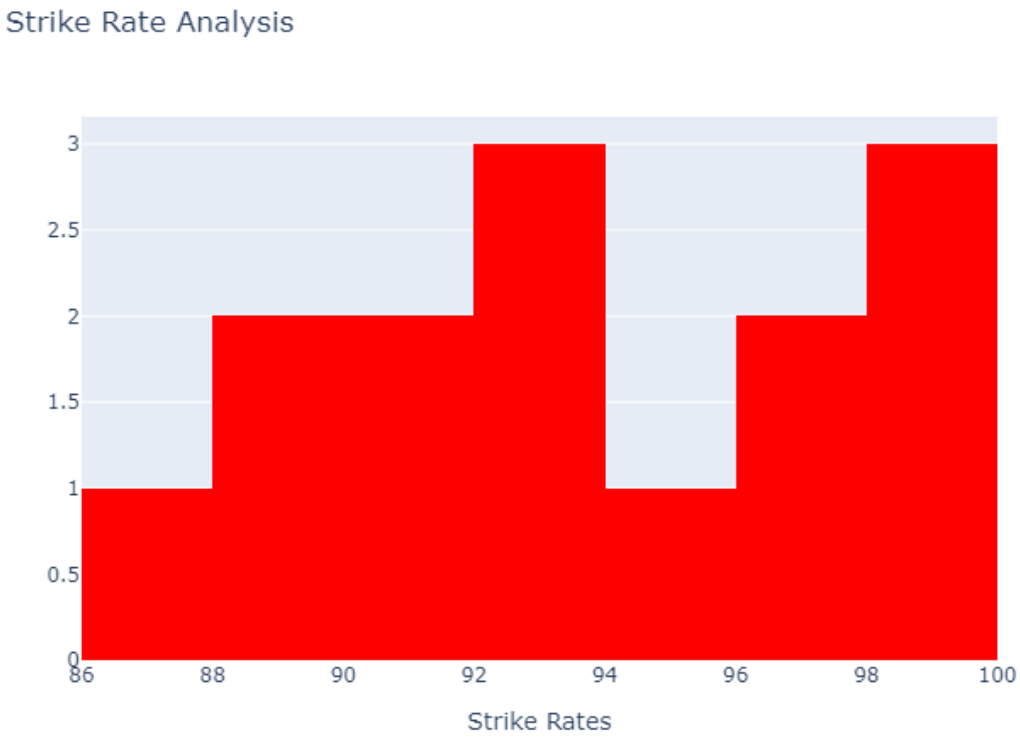
In [57]:
```
hist_data = [avg['avg'], avg['batsman_runs']]
group_labels = ['Average', 'Strike Rate']

fig = ff.create_distplot(hist_data, group_labels) #bin_size = [10, 20]
pyo.plot(fig, filename = 'Average.html')
fig.write_image('dist_plot_1.png')
```



Histogram

In [58]:
```
x = delivery.groupby('batsman')['batsman_runs'].count()>150
x = x[x].index.tolist()

new = delivery[delivery['batsman'].isin(x)]
runs = new.groupby('batsman')['batsman_runs'].sum()
balls = new.groupby('batsman')['batsman_runs'].count()

sr = (runs/balls)*100
sr = sr.reset_index()
```

In [59]:
```
trace = go.Histogram(x = sr['batsman_runs'], marker = {'color': 'red'}, xbins =
data = [trace]
layout = go.Layout(title = 'Strike Rate Analysis',
                   xaxis = {'title': 'Strike Rates'})

fig = go.Figure(data = data, layout = layout)
pyo.plot(fig, filename = 'ipl_Strike Rate Analysis.html')
fig.write_image('histogram.png')
```

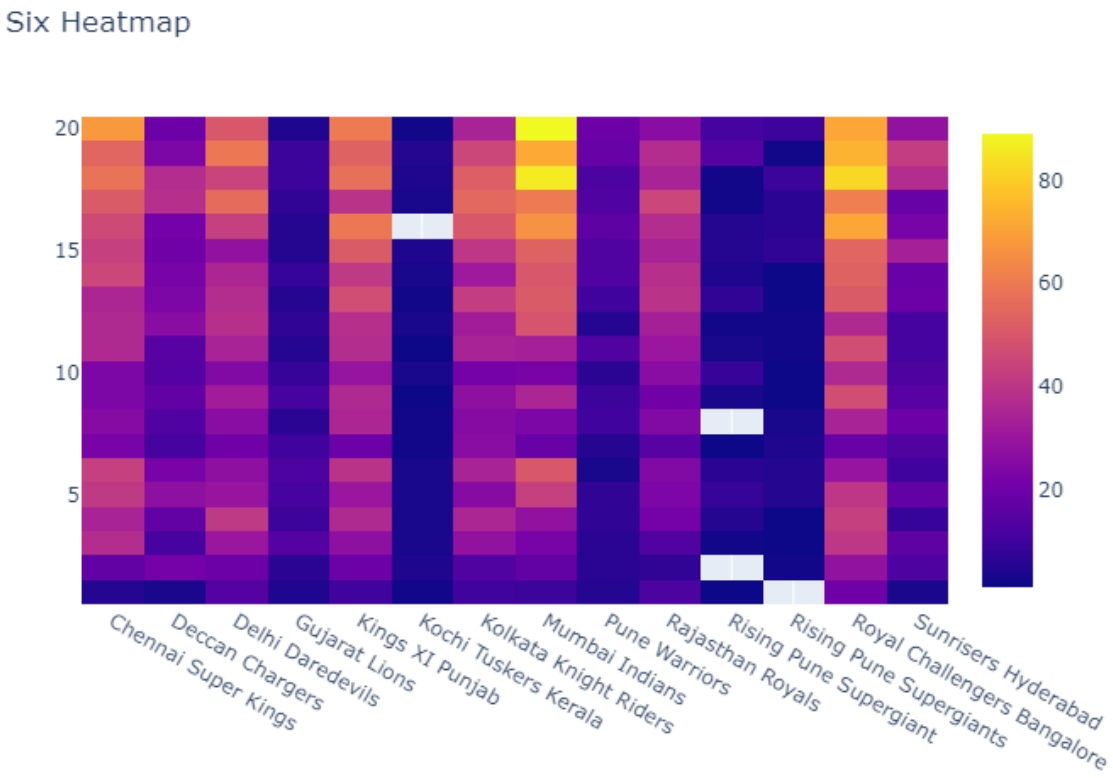


Heatmap

In [60]:
```
from plotly import tools
```

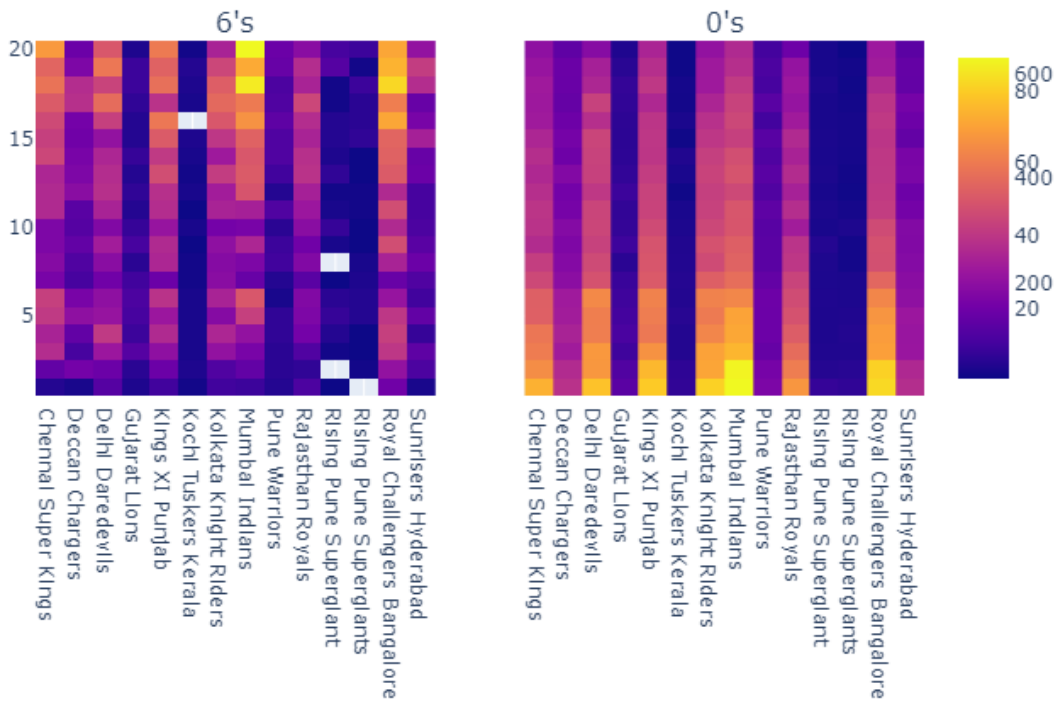
In [61]:
```
six = delivery[delivery['batsman_runs']== 6]
six = six.groupby(['batting_team', 'over'])['batsman_runs'].count().reset_index
```

In [62]:
```
trace = go.Heatmap(x = six['batting_team'], y = six['over'], z = six['batsman_r
data = [trace]
layout = go.Layout(title = 'Six Heatmap')

fig = go.Figure(data = data, layout = layout)
pyo.plot(fig, filename = 'ipl_Sex Heatmap.html')
fig.write_image('heatmap.png')
```



In [63]:
```
dots  = delivery[delivery['batsman_runs']== 0]
dots = dots.groupby(['batting_team', 'over'])['batsman_runs'].count().reset_in

trace_1 = go.Heatmap(x = six['batting_team'], y = six['over'],
                     z = six['batsman_runs'].values.tolist())
trace_2 = go.Heatmap(x = dots['batting_team'], y = dots['over'],
                     z = dots['batsman_runs'].values.tolist())

fig = tools.make_subplots(rows = 1, cols = 2, subplot_titles = ["6's", "0's"],

fig.append_trace(trace_1, 1, 1)
fig.append_trace(trace_2, 1, 2)
pyo.plot(fig, filename = 'ipl_S6 & Z0.html')
fig.write_image('heatmap_1.png')
```

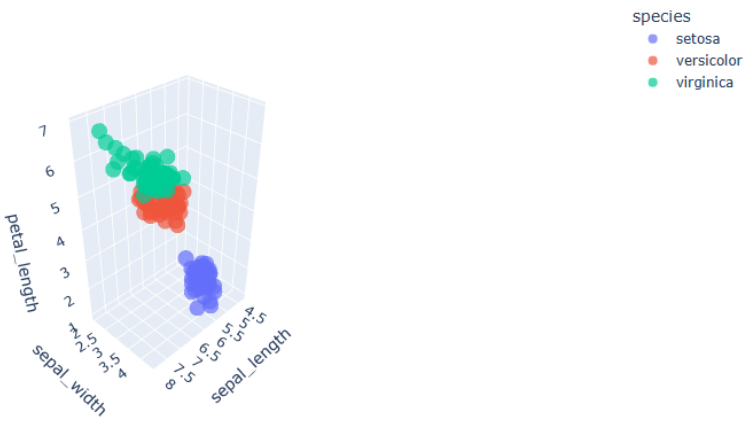
```
C:\Users\prasad jadhav\AppData\Local\Programs\Python\Python310\lib\site-packag
es\plotly\tools.py:460: DeprecationWarning:

plotly.tools.make_subplots is deprecated, please use plotly.subplots.make_subp
lots instead
```



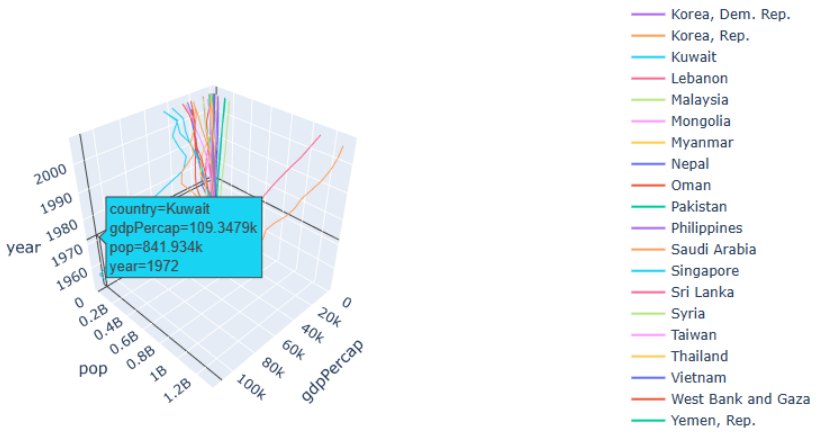
In [64]:
```
import numpy as np
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
```

In [65]:
```
iris_df = px.data.iris()
```

In [ ]:
```
fig = px.scatter_3d(iris_df, x = 'sepal_length', y = 'sepal_width', z = 'petal_

fig.show()
fig.write_image('3d.png')
```

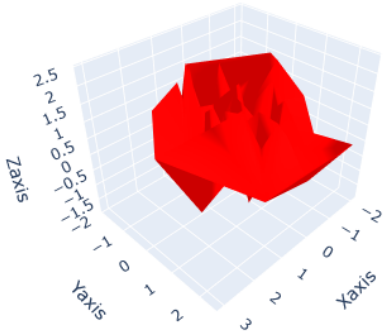


In [ ]:
```
gapminder_df = px.data.gapminder().query("continent == 'Asia'")

fig = px.line_3d(gapminder_df, x = 'gdpPercap', y = 'pop', z = 'year', color =
fig.show()
```



_3DAxes

In [ ]:
```
n = 100

fig = go.Figure(data = go.Mesh3d(x = np.random.randn(n),
                                 y = np.random.randn(n),
                                 z = np.random.randn(n), color = 'red'))
fig.update_layout(scene = dict(xaxis_title = 'Xaxis',
                               yaxis_title = 'Yaxis',
                               zaxis_title = 'Zaxis')) #xaxis = dict(backgroun
fig.show()
```



Thank You

CONNECT WITH ME :

PRASADMJADHAV2