Interactive Data Visualization – Part 1

Main Code

```
plot_ly(data = dataframe, x = ~name_of_variable1,
                  y = ~name_of_variable2,
          color = ~name_variable3,
          symbol = ~name_variable3,
          type = "", mode = "",
          colors = pallete_vector)
```

Type	Mode	Type of Plot	Add on functions
type = “scatter”	mode = “markers”	Scatter Plot	add_trace() layout()
	mode = "lines"	Line Graph	
	mode = "lines+markers"	Line Graph with dots	
type = "histogram"		Histogram	add_histogram() layout()
type = "bar"		Bar Graph	add_trace() layout() Inside layout: barmode = "group" in layout barmode = "stack" in layout
type = "box"		Box Plot	add_trace

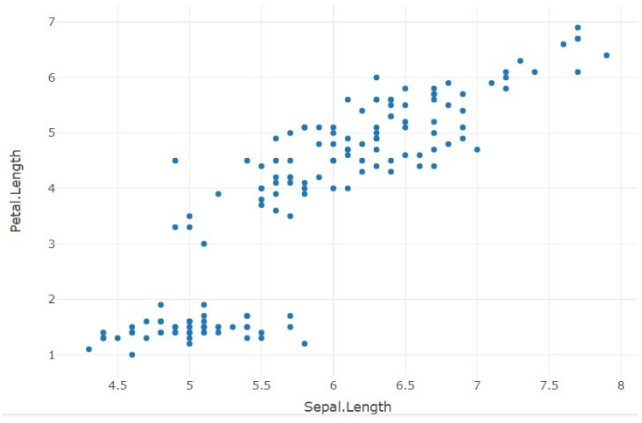
Dataset: Iris

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa
11	5.4	3.7	1.5	0.2	setosa

Scatter Plots with Plotly

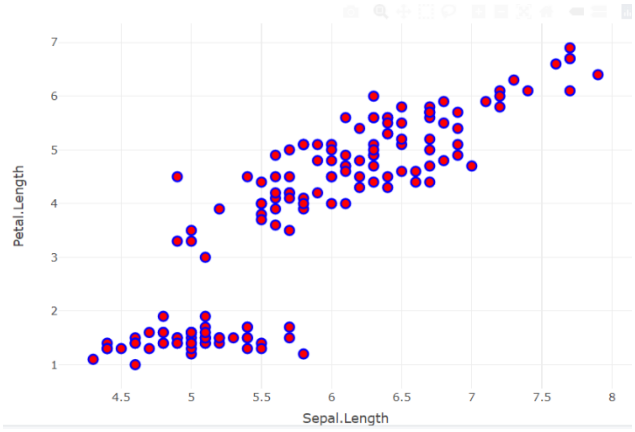
Simple Scatter Plot

```
fig1 <- plot_ly(data = iris, x = ~Sepal.Length, y = ~Petal.Length,
                type = "scatter", mode = "markers")
fig1
```



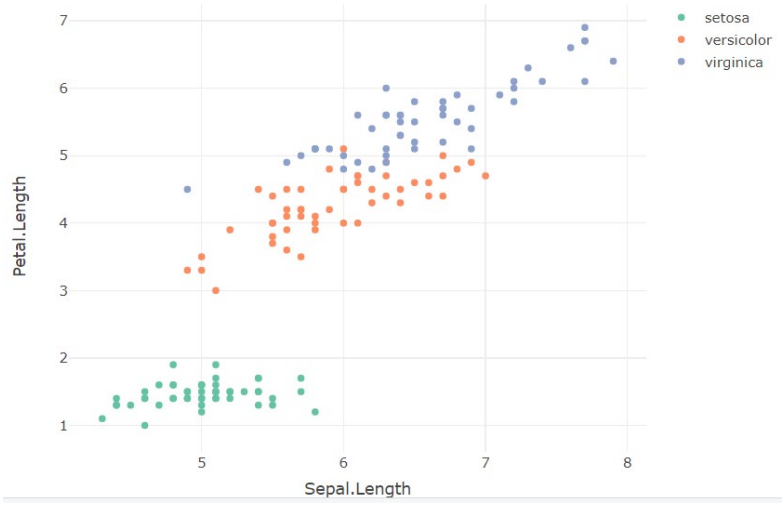
Stylish Dots

```
fig2 <- plot_ly(data = iris, x = ~Sepal.Length, y = ~Petal.Length,
                type = "scatter", mode = "markers",
                marker = list(size = 10,
                              color = "red",
                              line = list(color = "blue",
                                          width = 2)))
fig2
```



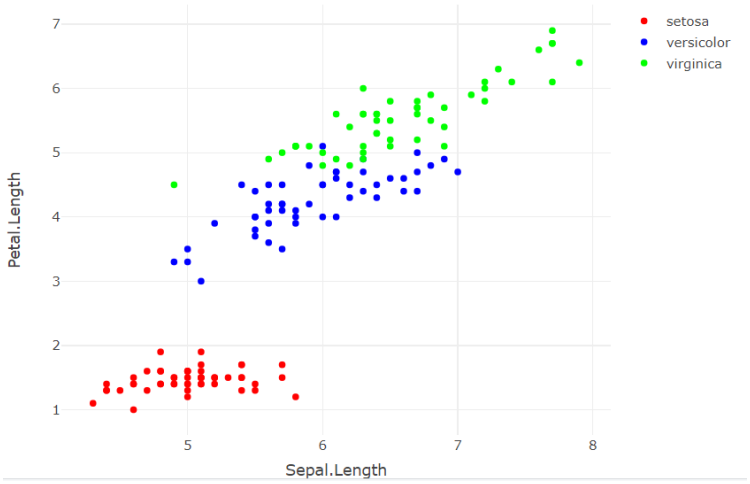
Three Variables

```
fig3 <- plot_ly(data = iris, x = ~Sepal.Length, y = ~Petal.Length,
                color = ~Species,
                type = "scatter", mode = "markers")
fig3
```



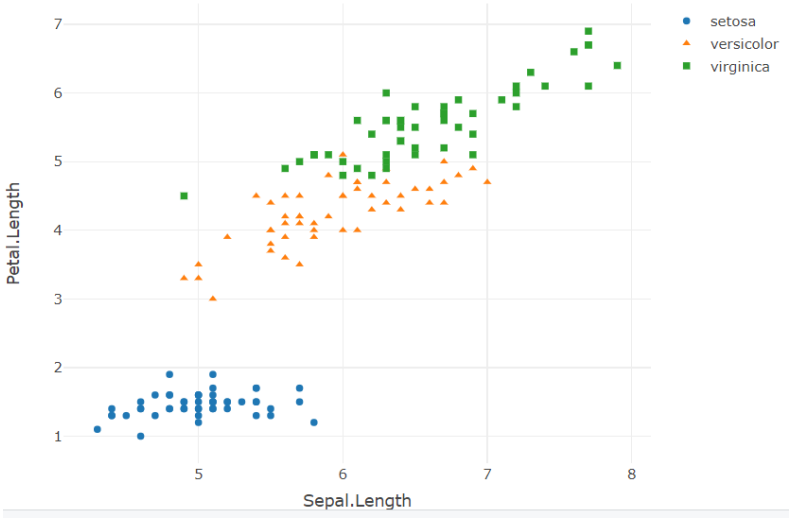
Manually Change the Colors for the Third Variable

```
fig4 <- plot_ly(data = iris, x = ~Sepal.Length, y = ~Petal.Length,
                color = ~Species,
                type = "scatter", mode = "markers",
                colors = c("red", "blue", "green"))
fig4
```



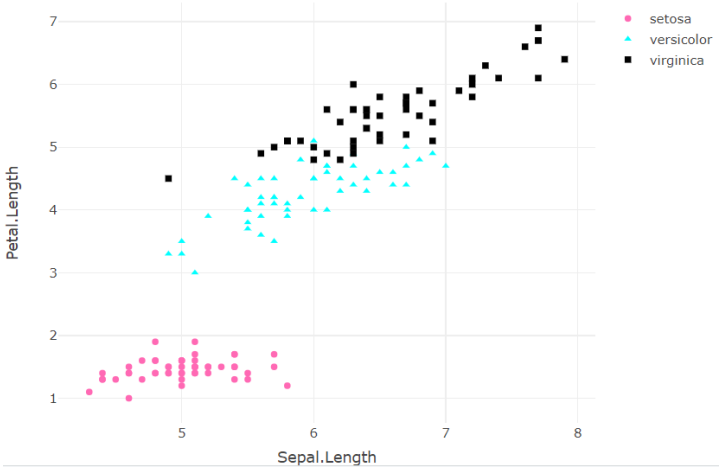
Symbols

```
fig5 <- plot_ly(data = iris, x = ~Sepal.Length, y = ~Petal.Length,
                symbol = ~Species,
                type = 'scatter', mode = 'markers')
fig5
```



Changing the colors of the symbols.

```
fig6 <- plot_ly(data = iris, x = ~Sepal.Length, y = ~Petal.Length,
                symbol = ~Species,
                color = ~Species,
                type = 'scatter', mode = 'markers',
                colors = c("hotpink", "cyan", "black"))
fig6
```



Dataset: dummy_data_1

	x	col_1	col_2	col_3	col_4
1	1	5.181458	-0.160951045	-0.951807493	-6.273066
2	2	5.768219	0.479718521	-0.610394550	-5.378716
3	3	6.520355	-0.846518233	-0.575741088	-4.148412
4	4	5.488036	0.355705894	2.464083378	-6.305281
5	5	5.088665	0.512848807	-0.499907128	-5.396206
6	6	4.392657	0.401019388	1.399282445	-4.572251
7	7	4.336057	-0.125980801	-0.746640188	-6.134401
8	8	4.651283	-0.245551624	0.423829716	-6.329959
9	9	6.017906	-0.160596048	0.821952838	-4.140733
10	10	4.213647	0.132291868	1.528079629	-4.230242

Layout (titles, background...etc)

Figure without layout

```
fig_7 <- plot_ly(data = dummy_data_1, x = ~col_2, y = ~col_3,
                         type = "scatter", mode = "markers")
fig_7
```

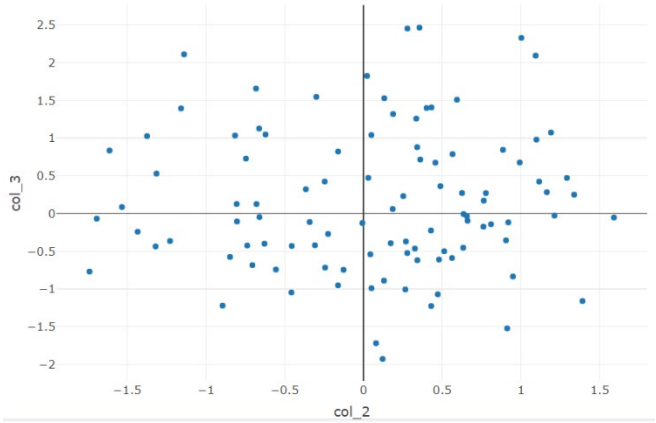


Figure with layout

```
fig8 <- plot_ly(data = dummy_data_1, x = ~col_2, y = ~col_3,
                type = "scatter", mode = "markers") %>%
  layout(title = "This is the Title of the Plot",
         legend = list(title=list(text="Legend Title")),
         plot_bgcolor="yellow",
         xaxis = list(
           title = "This is my x axis",
           zerolinecolor = "blue",
           zerolinewidth = 10,
           gridcolor = "red"),
         yaxis = list(
           title = "This is my y axis",
           zerolinecolor = "pink",
           zerolinewidth = 10,
           gridcolor = "black"))
fig8
```

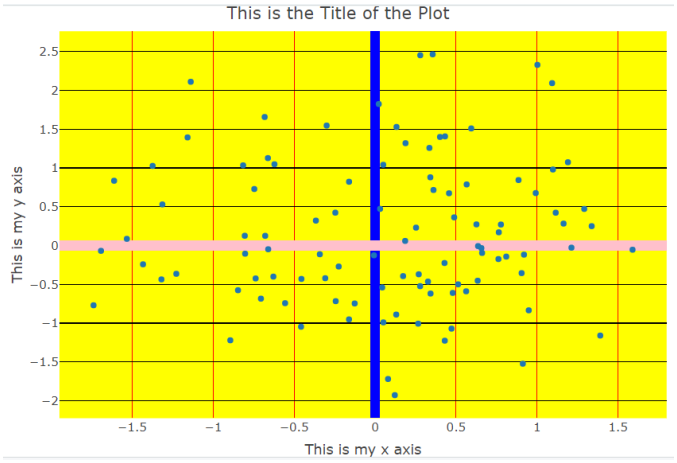
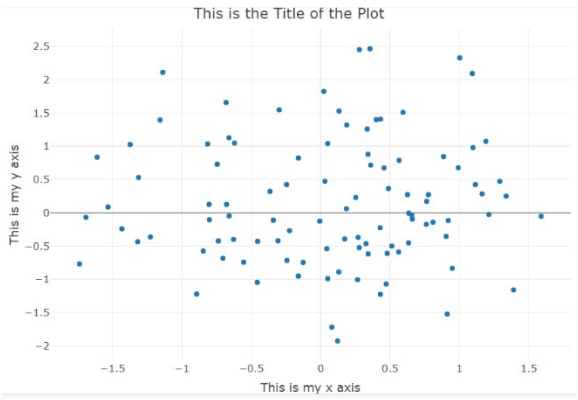


Figure with layout (zeroline)

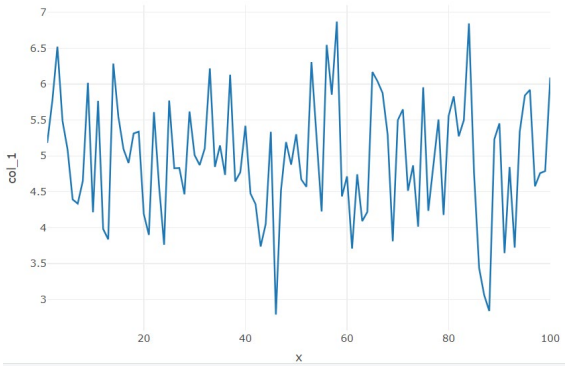
```
fig9 <- plot_ly(data = dummy_data_1, x = ~col_2, y = ~col_3,
                type = "scatter", mode = "markers") %>%
  layout(title = "This is the Title of the Plot",
         xaxis = list(
           title = "This is my x axis",
           zeroline = FALSE),
         yaxis = list(
           title = "This is my y axis"))
fig9
```



Line Plots

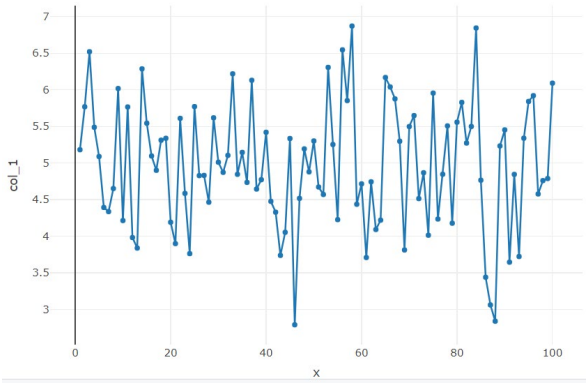
Simple Line Plot

```
fig10 <- plot_ly(data = dummy_data_1, x = ~x, y = ~col_1,
                 type = "scatter", mode = "lines")
fig10
```



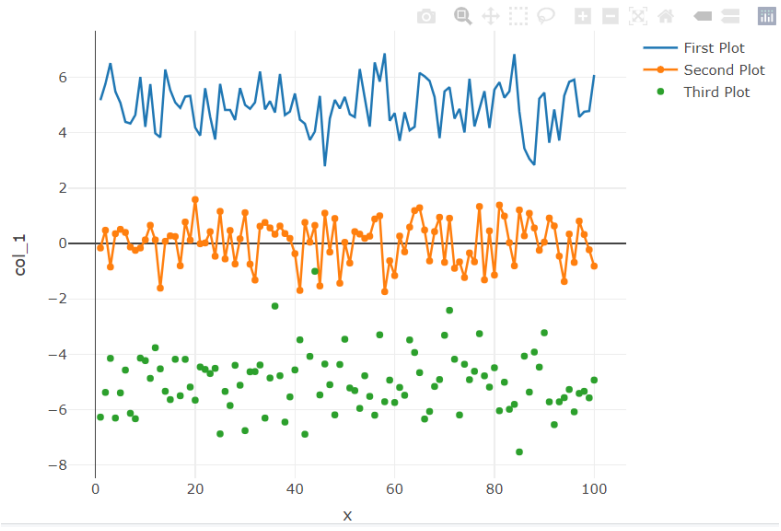
Line Plot with Markers

```
fig11 <- plot_ly(data = dummy_data_1, x = ~x, y = ~col_1,
                 type = "scatter", mode = "lines+markers")
fig11
```



Combining Multiple Plots

```
fig12 <- plot_ly(data = dummy_data_1, x = ~x, y = ~col_1,
                 name = "First Plot",
                 type = "scatter", mode = "lines") %>%
  add_trace(y = ~col_2, name = "Second Plot",
            type = "scatter", mode = "lines+markers") %>%
  add_trace(y = ~col_4, name = "Third Plot",
            type = "scatter", mode = "markers")
fig12
```



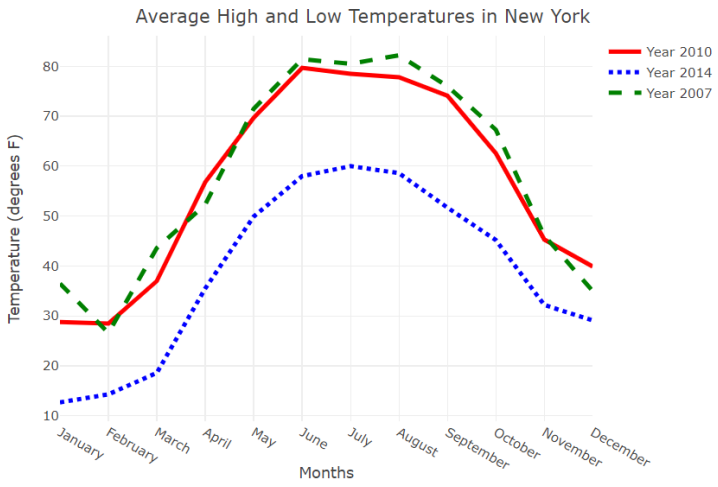
Dataset: dummy_data_2

	month	year_2007	year_2010	year_2014
1	January	36.5	28.8	12.7
2	February	26.6	28.5	14.3
3	March	43.6	37.0	18.6
4	April	52.3	56.8	35.5
5	May	71.5	69.7	49.9
6	June	81.4	79.7	58.0
7	July	80.5	78.5	60.0
8	August	82.2	77.8	58.6
9	September	76.0	74.1	51.7
10	October	67.3	62.6	45.2
11	November	46.1	45.3	32.2
12	December	35.0	39.9	29.1

Style- Multiple Line Plot

```
fig13 <- plot_ly(dummy_data_2, x = ~month, y = ~year_2010, name = "Year 2010",
                 type = "scatter", mode = "lines",
                 line = list(color = "red", width = 4))%>%
  add_trace(y = ~year_2014, name = "Year 2014",
            type = "scatter", mode = "lines",
            line = list(color = "blue", width = 4, dash = "dot")) %>%
  add_trace(y = ~year_2007, name = "Year 2007",
            type = "scatter", mode = "lines",
            line = list(color = "green", width = 4, dash = "dash")) %>%
  layout(title = "Average High and Low Temperatures in New York",
         xaxis = list(title = "Months"),
         yaxis = list (title = "Temperature (degrees F)"))
```

```
fig13
```



Combining ggplot2 with Plotly

You can enhance static ggplot2 plots with interactivity by using ggplotly(). The ggplotly() function is used to convert the ggplot plot to a Plotly interactive plot.

```
fig14 <- dummy_data_1 %>%
  ggplot(aes(x = col_1, y = col_2))+
  geom_point()
fig14

ggplotly(fig14)
```

Note: See this plot on R