

## Interactive Data Visualization - Part 2

### 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

## Histograms

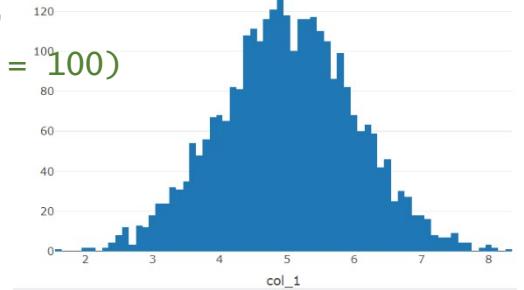
Dataframe: data\_1

	x	col_1	col_2	col_3	col_4
1	1	4.452418	3.6876016	0.052225403	-4.756294
2	2	2.955263	2.8928292	1.227396980	-3.771439
3	3	5.779916	3.6968157	2.321996332	-3.720551
4	4	4.724015	3.6440147	1.123346227	-3.463984
5	5	4.289810	2.4371923	0.136189320	-5.512061
6	6	3.477498	4.7305132	-0.492429713	-4.876484
7	7	4.626856	3.1279934	-0.239440013	-4.561630
8	8	2.612686	4.4751611	0.612236758	-4.645969
9	9	3.982926	1.9682513	-0.531182509	-5.214656
10	10	3.691051	2.6501433	0.786965740	-4.992421

### Simple Histogram

```
fig1 <- plot_ly(data = data_1, x = ~col_1,  
                  type = "histogram", nbinsx = 100)
```

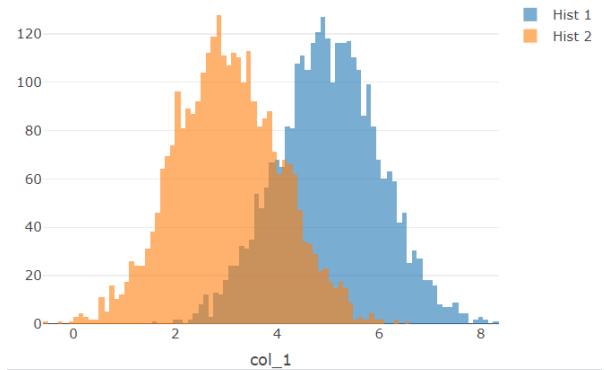
```
fig1
```



### Overlaid Histogram (alpha controls the level of transparency)

```
fig2 <- plot_ly(data = data_1, x = ~col_1,  
                  type = "histogram", alpha = 0.6, name = "Hist 1") %>%  
  add_histogram(x = ~col_2, name = "Hist 2") %>%  
  layout(barmode = "overlay")
```

```
fig2
```



## Bar Graphs

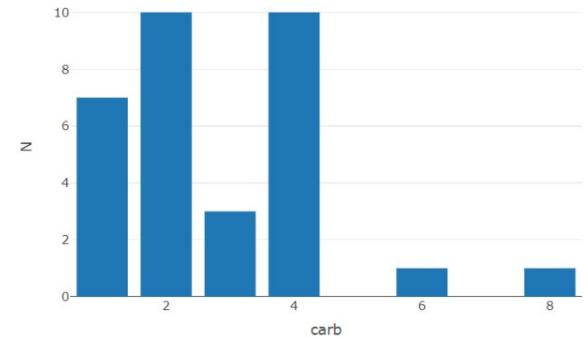
Dataframe: data\_2

	carb	N
1	1	7
2	2	10
3	3	3
4	4	10
5	6	1
6	8	1

### Simple Bar Graph

```
fig3 <- plot_ly(data = data_2,  
                  x = ~carb, y = ~N,  
                  type = "bar")
```

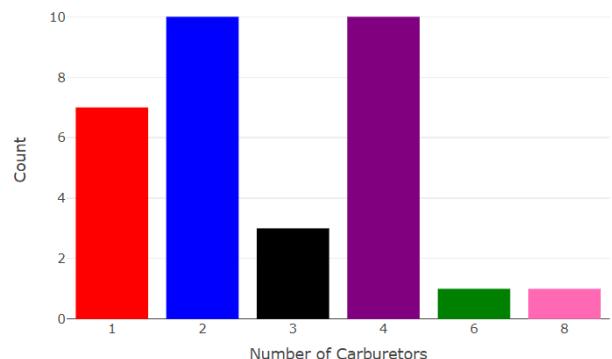
fig3



### Changing colors in a Bar Graph

```
fig4 <- plot_ly(data = data_2,  
                  x = ~factor(carb), y = ~N,  
                  type = "bar",  
                  marker = list(color = c("red", "blue", "black",  
                                         "purple", "green", "hotpink")) ) %>%  
  layout(xaxis = list(title = "Number of Carburetors"),  
         yaxis = list(title = "Count"))
```

fig4



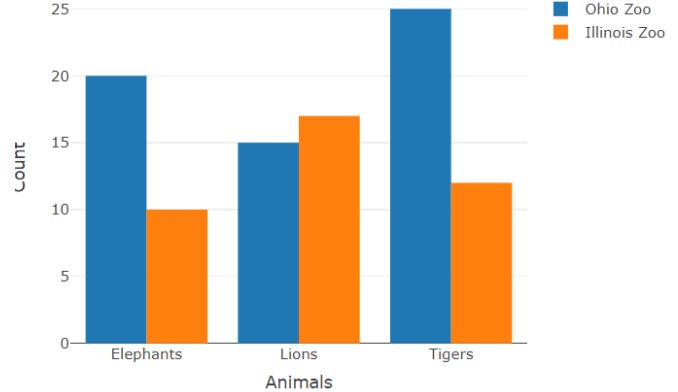
### Dataframe: data\_3

	Animals	OH_Zoo	IL_Zoo
1	Elephants	20	10
2	Lions	15	17
3	Tigers	25	12

### Grouped Bar Graph

```
fig5 <- plot_ly(data = data_3, x = ~Animals, y = ~OH_Zoo,
                  type = "bar", name = "Ohio zoo") %>%
  add_trace(y = ~IL_Zoo, name = "Illinois zoo") %>%
  layout(yaxis = list(title = "Count"),
         barmode = "group")
```

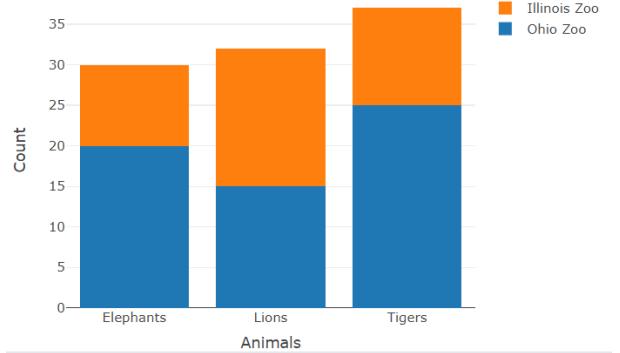
fig5



### Stacked Bar Graph

```
fig6 <- plot_ly(data = data_3, x = ~Animals, y = ~OH_Zoo,
                  type = "bar", name = "Ohio zoo") %>%
  add_trace(y = ~IL_Zoo, name = "Illinois zoo") %>%
  layout(yaxis = list(title = "Count"),
         barmode = "stack")
```

fig6



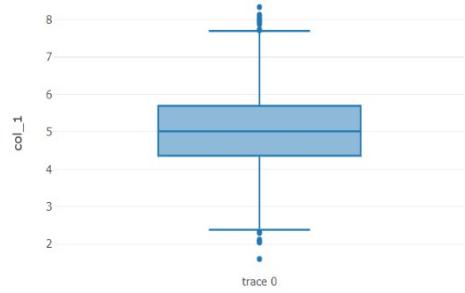
## Boxplots

Dataframe: data\_1

	x	col_1	col_2	col_3	col_4
1	1	4.452418	3.6876016	0.052225403	-4.756294
2	2	2.955263	2.8928292	1.227396980	-3.771439
3	3	5.779916	3.6968157	2.321996332	-3.720551
4	4	4.724015	3.6440147	1.123346227	-3.463984
5	5	4.289810	2.4371923	0.136189320	-5.512061
6	6	3.477498	4.7305132	-0.492429713	-4.876484
7	7	4.626856	3.1279934	-0.239440013	-4.561630
8	8	2.612686	4.4751611	0.612236758	-4.645969
9	9	3.982926	1.9682513	-0.531182509	-5.214656
10	10	3.691051	2.6501433	0.786965740	-4.992421

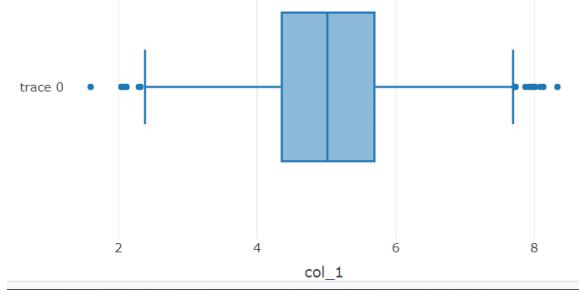
### Vertical Boxplot

```
fig7 <- plot_ly(data = data_1, y = ~col_1, type = "box")
fig7
```



### Horizontal Boxplot

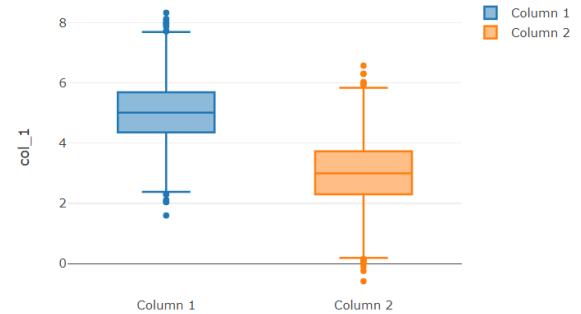
```
fig8 <- plot_ly(data = data_1, x = ~col_1, type = "box")
fig8
```



### **Multiple Boxplots**

```
fig9 <- plot_ly(data = data_1, y = ~col_1,
                  type = "box", name = "Column 1") %>%
  add_trace(y = ~col_2, name = "Column 2")
```

fig9



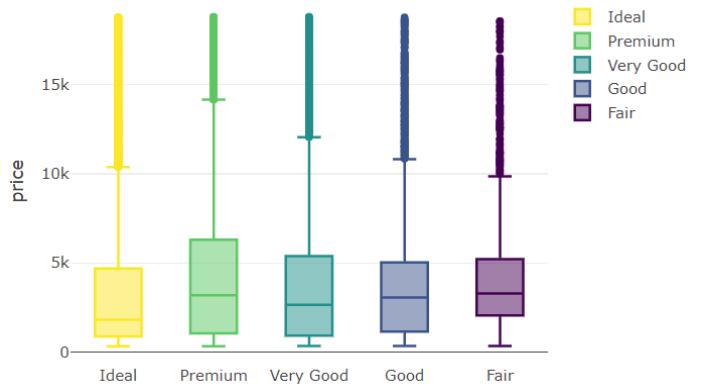
### **Multiple Boxplots by Categorical Variable**

**Dataframe: diamonds**

	carat	cut	color	clarity	depth	table	price	x	y	z
1	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
2	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
3	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
4	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
5	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75
6	0.24	Very Good	J	VVS2	62.8	57.0	336	3.94	3.96	2.48
7	0.24	Very Good	I	VVS1	62.3	57.0	336	3.95	3.98	2.47
8	0.26	Very Good	H	SI1	61.9	55.0	337	4.07	4.11	2.53
9	0.22	Fair	E	VS2	65.1	61.0	337	3.87	3.78	2.49
10	0.23	Very Good	H	VS1	59.4	61.0	338	4.00	4.05	2.39

```
fig10 <- plot_ly(data = diamonds,
                  y = ~price, color = ~cut, type = "box")
```

fig10



## Interactive Tables with Datatable

Dataframe: Beatles

	year	sex	name	n	prop
1	1880	M	John	9655	0.08154561
2	1880	M	George	5126	0.04329392
3	1880	M	Paul	301	0.00254223
4	1881	M	John	8769	0.08098299
5	1881	M	George	4664	0.04307272
6	1881	M	Paul	291	0.00268743

```
data("babynames")
Beatles <- babynames %>%
  filter(name %in% c("John", "Paul", "George", "Ringo") & sex == "M")
```

In this section, an interactive table is generated using the DT package. With the function `datatable()` we will turn a dataframe into an interactive table.

```
interactive_table <- datatable(Beatles, options = list(pageLength = 10))
interactive_table
```

Show 10 entries

Search:

year	sex	name	n	prop
1	1880	M	John	9655
2	1880	M	George	5126
3	1880	M	Paul	301
4	1881	M	John	8769

Showing 1 to 10 of 442 entries

Previous 1 2 3 4 5 ... 45 Next

## Dygraph

This code generates an interactive time series plot showing the popularity of Beatles names over time. You can adjust the date range and explore the data dynamically.

Dygraphs is a powerful tool for visualizing time-dependent information.

- We use the Beatles dataset.
- We select the relevant columns, such as year, name, and prop (popularity).
- We use pivot\_wider to reformat the data.
- dygraph(main = "Popularity of Beatles names over time") initializes a Dygraph plot and sets the title.
- dyRangeSelector(dateWindow = c("1900", "1980")) provides a range selector to focus on specific years.

```
fig11 <- Beatles %>%
  select(year, name, prop) %>%
  pivot_wider(names_from = name, values_from = prop) %>%
  dygraph(main = "Popularity of Beatles names over time") %>%
  dyRangeSelector(datewindow = c("1900", "1980"))
```

fig11

