

In []:

12 Lolipop_Grafiği:

Tanım:

Lolipop-Stem plot temelde Bar plot türevidir. Bar plot da olduğu gibi bir çizgi ve veri değerini gösteren bir daireden ibarettir.Sayısal ve kategorik iki değişken arasındaki ilişkiyi gösterir. Her bir kategorinin değeri numerik daire ile gösterilir. Daireyi kategoriye bir çizgi bağlar.

In [16]:

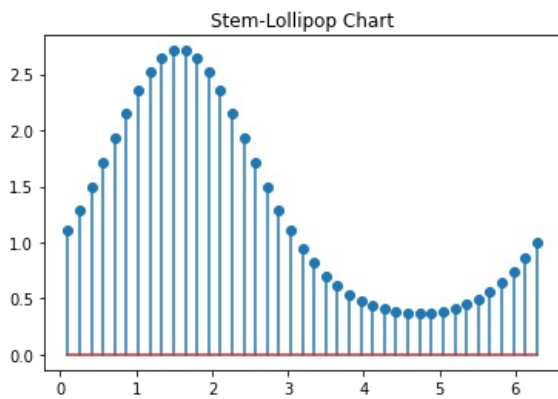
```
import matplotlib.pyplot as plt
import numpy as np
```

```
x = np.linspace(0.1, 2 * np.pi, 41)
y = np.exp(np.sin(x))
```

```
plt.title("Stem-Lollipop Chart")
plt.stem(x, y)
plt.show()
```

<ipython-input-16-6a42b74cf6c2>:8: UserWarning: In Matplotlib 3.3 individual lines on a stem plot will be added as a LineCollection instead of individual lines. This significantly improves the performance of a stem plot. To remove this warning and switch to the new behaviour, set the "use_line_collection" keyword argument to True.

```
plt.stem(x, y)
```



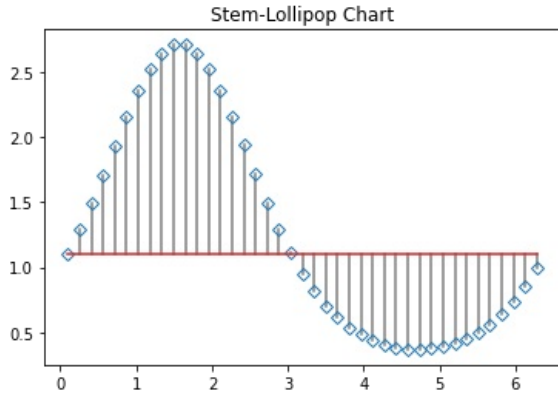
In []:

In [17]:

```
markerline, stemlines, baseline = plt.stem(
    x, y, linefmt='grey', markerfmt='D', bottom=1.1)
markerline.set_markerfacecolor('none')
plt.title("Stem-Lollipop Chart")
plt.show()
```

<ipython-input-17-29528c113026>:1: UserWarning: In Matplotlib 3.3 individual lines on a stem plot will be added as a LineCollection instead of individual lines. This significantly improves the performance of a stem plot. To remove this warning and switch to the new behaviour, set the "use_line_collection" keyword argument to True.

```
markerline, stemlines, baseline = plt.stem(
```



Ne için Kullanılır:

Lollipop-Stem plot temelde Bar plot türevidir sadece bazı durumlarda daha ilgi çekici bir grafikdir. Bar plot da olduğu gibi bir çizgi ve veri değerini gösteren bir daireden ibarettir. Sayısal ve kategorik iki değişken arasındaki ilişkiyi gösterir. Her bir kategorinin değeri numerik daire ile gösterilir. Daireyi kategoriye çizgi bağlar.

Lollipop grafiği özellikle değerlerin vurgulanması açısından bar plottan biraz daha verimlidir.

In [18]:

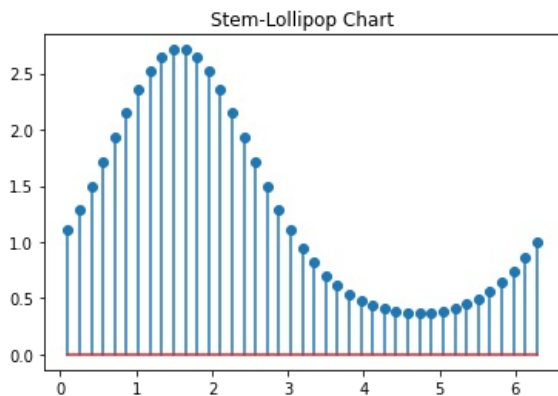
```
import matplotlib.pyplot as plt
import numpy as np
```

```
x = np.linspace(0.1, 2 * np.pi, 41)
y = np.exp(np.sin(x))
```

```
plt.title("Stem-Lollipop Chart")
plt.stem(x, y)
plt.show()
```

<ipython-input-18-6a42b74cf6c2>:8: UserWarning: In Matplotlib 3.3 individual lines on a stem plot will be added as a LineCollection instead of individual lines. This significantly improves the performance of a stem plot. To remove this warning and switch to the new behaviour, set the "use_line_collection" keyword argument to True.

```
plt.stem(x, y)
```

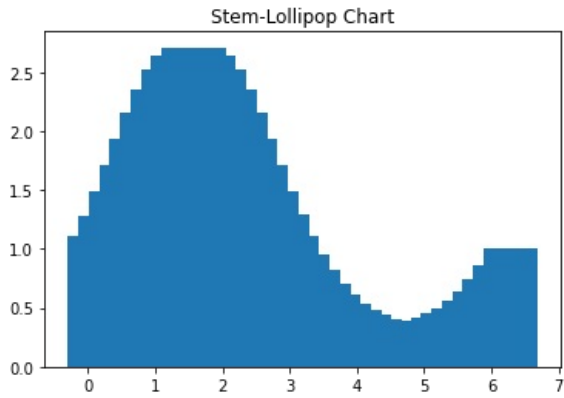


In [19]:

```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(0.1, 2 * np.pi, 41)
y = np.exp(np.sin(x))

plt.title("Stem-Lollipop Chart")
plt.bar(x, y)
plt.show()
```



Görüldüğü üzere Stem- Lollipop grafiği bar grafiğinden görsel açıdan pek çok yerde daha kullanışlıdır.

Çeşitleri:

In []:

Yatay ve Dikey çeşitleri mevcuttur. Tabi ki yatay lolipop grafiği Bar grafiğinde olduğu gibi uzun isimlendirmelerde avantajlıdır.

Ayrıca Cleveland dotplot-Dumbbell dot plot iki ayrı verinin kıyaslanması için oldukça elverişli bir Lolipop grafiğidir.

Ayrıca bu grafik türünde de bar plot da olduğu gibi gruplanmış Lolipop mevcuttur.

In [20]:

```
https://www.data-to-viz.com/graph/lollipop.html
https://datavizproject.com/data-type/dumbbell-plot/
```

File "<ipython-input-20-6bddc7a7b8a7>", line 1
https://www.data-to-viz.com/graph/lollipop.html
^

SyntaxError: invalid syntax

In [21]:

```
# libraries
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Create a dataframe
df = pd.DataFrame({'group':list(map(chr, range(65, 85))), 'values':np.random.uniform(size=20) })

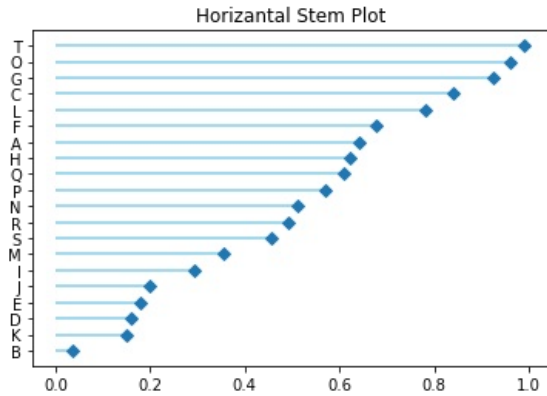
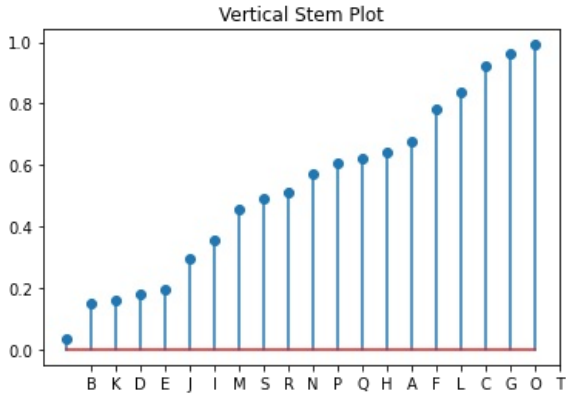
# Reorder it based on the values:
ordered_df = df.sort_values(by='values')
my_range=range(1,len(df.index)+1)

# Make the plot
plt.title("Vertical Stem Plot")
plt.stem(ordered_df['values'])
plt.xticks( my_range, ordered_df['group'])
plt.show()

# Horizontal version
plt.title("Horizontal Stem Plot")
plt.hlines(y=my_range, xmin=0, xmax=ordered_df['values'], color='skyblue')
plt.plot(ordered_df['values'], my_range, "D")
plt.yticks(my_range, ordered_df['group'])
plt.show()
```

<ipython-input-21-333f229130f5>:15: UserWarning: In Matplotlib 3.3 individual lines on a stem plot will be added as a LineCollection instead of individual lines. This significantly improves the performance of a stem plot. To remove this warning and switch to the new behaviour, set the "use_line_collection" keyword argument to True.

```
plt.stem(ordered_df['values'])
```



Kullanım Hataları:

Bar çubuklarının değerleri arasında bir sırlama olmaması verinin görselleştirilmesini zorlaştıracığından verileri öncelikle sıralamak daha uygun olacaktır.

Kategorik değişkeninizin seviyelerinin belirgin bir sırası yoksa, çubukları değerlerine göre sıralayın

Herhangi bir nedenle çubuklarınızın sıralanmamış kalması gerekiyorsa, bunun yerine bir barplot kullanmak muhtemelen daha iyidir. Zira Lolipop grafiği okumak daha zor olacaktır.

Her bir grupta çok fazla değer olması durumunda box plot veya violin kullanmak daha mantıklıdır.

Yatay sürümü bar grafiğinde olduğu gibi etiketlerin okunmasını kolaylaştırır.

In [22]:

```
https://www.data-to-viz.com/caveat/order_data.html
```

```
File "<ipython-input-22-ac5ddd6aa821>", line 1
  https://www.data-to-viz.com/caveat/order_data.html
    ^
```

SyntaxError: invalid syntax

In [23]:

```
# Libraries
library(tidyverse)
library(hrbrthemes)
library(kableExtra)
options(knitr.table.format = "html")

# Load dataset from github
data <- read.table("https://raw.githubusercontent.com/holtzy/data_to_viz/master/Example_dataset/7_OneCatOneNum.csv", header=TRUE, sep=",")

# Plot
data %>%
  filter(!is.na(Value)) %>%
  ggplot( aes(x=Country, y=Value) ) +
    geom_segment( aes(x=Country, xend=Country, y=0, yend=Value), color="grey") +
    geom_point(size=3, color="#69b3a2") +
    coord_flip() +
    theme_ipsum() +
    theme(
      panel.grid.minor.y = element_blank(),
      panel.grid.major.y = element_blank(),
      legend.position="none"
    ) +
    xlab("")
```

```
File "<ipython-input-23-ea2fe26b1a41>", line 11
  data %>%
    ^
```

SyntaxError: invalid syntax

In []:

In []:

In []:

In []:

In []: