

In []:

13-Bullet Plot:

Tanım:

Bullet Grafikleri bir Bar grafiği çeşidi olup, termometre gösteriminden ilham alınarak tasarlanmıştır. Kullanımdaki amaç bir hedef ve işlem aralığı belirlenip verilen değerin bu iki açıdan incelenmesi amacıyla tasarlanmıştır.

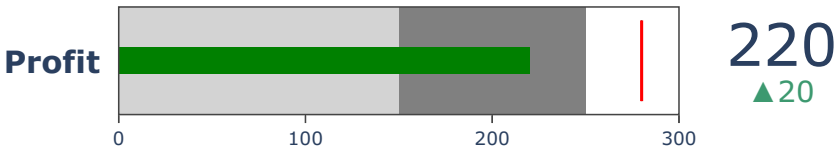
Ne için Kullanılır:

Özellikle teknik verilerin kesin bir değerinin olmadığı belli aralıklarda çalıştığı düşünülerek kötü - iyi- çok yüksek ve alarm gibi veriler için kullanımı veya kötü-iyi - çok iyi ve prim gibi veriler için kullanılabilir.

In [1]:

```
import plotly.graph_objects as go

fig = go.Figure(go.Indicator(
    mode = "number+gauge+delta", value = 220,
    domain = {'x': [0.1, 1], 'y': [0, 1]},
    title = {'text' : "<b>Profit</b>"},
    delta = {'reference': 200},
    gauge = {
        'shape': "bullet",
        'axis': {'range': [None, 300]},
        'threshold': {
            'line': {'color': "red", 'width': 2},
            'thickness': 0.75,
            'value': 280},
        'steps': [
            {'range': [0, 150], 'color': "lightgray"},
            {'range': [150, 250], 'color': "gray"}]))
fig.update_layout(height = 250)
fig.show()
```



Yukarıdaki grafikte görüldüğü gibi iyi-kötü performansın belirlendiği gri tonlarında belli aralıklar ve kırmızı çizgi ile belirlenmiş bir hedef söz konusudur.

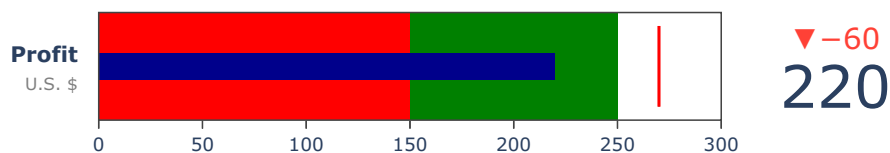
Çeşitleri:

Bullet grafiğinin yatay, dikey, tekli ve çoklu gibi çeşitleri olmakla birlikte temelde tüm bunlar aynı işlemi yapan aynı okuma özellikleri olan grafiklerdir. Ayrıca sadece gri tonları değil istediğini aralıklar için istediğiniz renleri de kullanıp daha basit grafikler de oluşturabilirsiniz.

In [2]:

```
import plotly.graph_objects as go

fig = go.Figure(go.Indicator(
    mode = "number+gauge+delta", value = 220,
    domain = {'x': [0, 1], 'y': [0, 1]},
    delta = {'reference': 280, 'position': "top"},
    title = {'text': "<b>Profit</b><br><span style='color: gray; font-size:0.8em'>U.S. $</span>", 'font': {"size":
14}},
    gauge = {
        'shape': "bullet",
        'axis': {'range': [None, 300]},
        'threshold': {
            'line': {'color': "red", 'width': 2},
            'thickness': 0.75, 'value': 270},
        'bgcolor': "white",
        'steps': [
            {'range': [0, 150], 'color': "red"},
            {'range': [150, 250], 'color': "green"}],
        'bar': {'color': "darkblue"}}))
fig.update_layout(height = 250)
fig.show()
```



In []:

In [3]:

```
import plotly.graph_objects as go

fig = go.Figure()

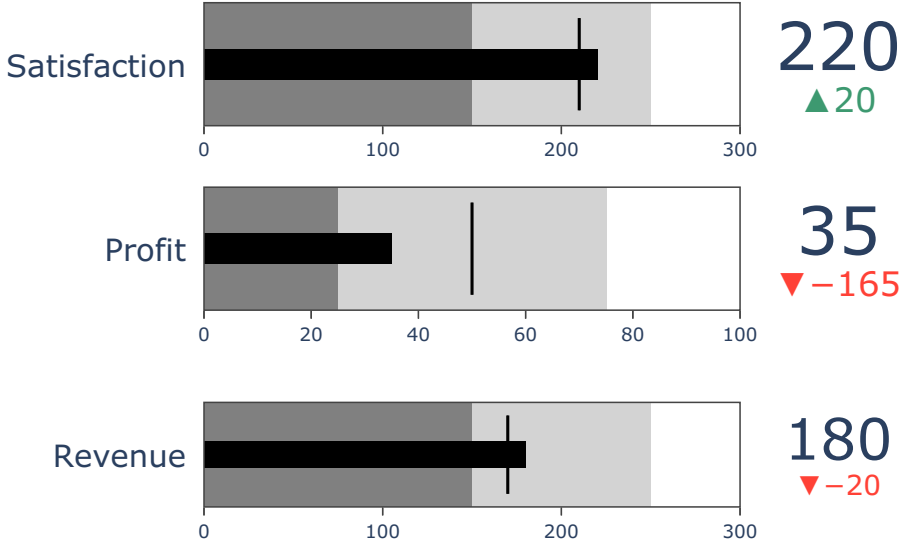
fig.add_trace(go.Indicator(
    mode = "number+gauge+delta", value = 180,
    delta = {'reference': 200},
    domain = {'x': [0.25, 1], 'y': [0.08, 0.25]},
    title = {'text': "Revenue"},
    gauge = {
        'shape': "bullet",
        'axis': {'range': [None, 300]},
        'threshold': {
            'line': {'color': "black", 'width': 2},
            'thickness': 0.75,
            'value': 170},
        'steps': [
            {'range': [0, 150], 'color': "gray"},
            {'range': [150, 250], 'color': "lightgray"}],
        'bar': {'color': "black"}}))

fig.add_trace(go.Indicator(
    mode = "number+gauge+delta", value = 35,
    delta = {'reference': 200},
    domain = {'x': [0.25, 1], 'y': [0.4, 0.6]},
    title = {'text': "Profit"},
    gauge = {
        'shape': "bullet",
        'axis': {'range': [None, 100]},
        'threshold': {
            'line': {'color': "black", 'width': 2},
            'thickness': 0.75,
            'value': 50},
        'steps': [
            {'range': [0, 25], 'color': "gray"},
            {'range': [25, 75], 'color': "lightgray"}],
        'bar': {'color': "black"}}))

fig.add_trace(go.Indicator(
    mode = "number+gauge+delta", value = 220,
    delta = {'reference': 200},
    domain = {'x': [0.25, 1], 'y': [0.7, 0.9]},
    title = {'text': "Satisfaction"},
    gauge = {
        'shape': "bullet",
        'axis': {'range': [None, 300]},
        'threshold': {
            'line': {'color': "black", 'width': 2},
            'thickness': 0.75,
            'value': 210},
        'steps': [
            {'range': [0, 150], 'color': "gray"},
            {'range': [150, 250], 'color': "lightgray"}],
        'bar': {'color': "black"}}))

fig.update_layout(height = 400 , margin = {'t':0, 'b':0, 'l':0})

fig.show()
```



Yukarıdaki gibi bir grafik birden fazla datayı karşılaştırmak için uygundur, ancak bir bir cihazın birden fazla açıdan verisi kıyaslanırken en alta bu verilerin toplam değerlendirme içerisindeki ağırlıkları göz önünde bulundurularak özet bir bullet grafiği oluşturulması daha uygun olacaktır. Zira görseller insan beyninde özelliğin ağırlığı göz önünde bulundurulacak biçimde hazırlanmamıştır.

Kullanım Hataları:

Yukarıda da belirtildiği üzere bir cihaza ait belirlenmiş özelliklerin her biri için bir bullet kullanıldığında bu özelliklerin toplam değeri için ayrı bir bullet grafiği oluşturulması daha uygun olacaktır. Örneğin bir telefonun özellikleri ve o sınıfta yer alan diğer telefonların değer aralıkları ile karşılaştırılması istendiğinde en sonunda toplam bir değerlendirme yapmamak kafa karıştırıcı olacaktır.

In [4]:

```
import plotly.graph_objects as go

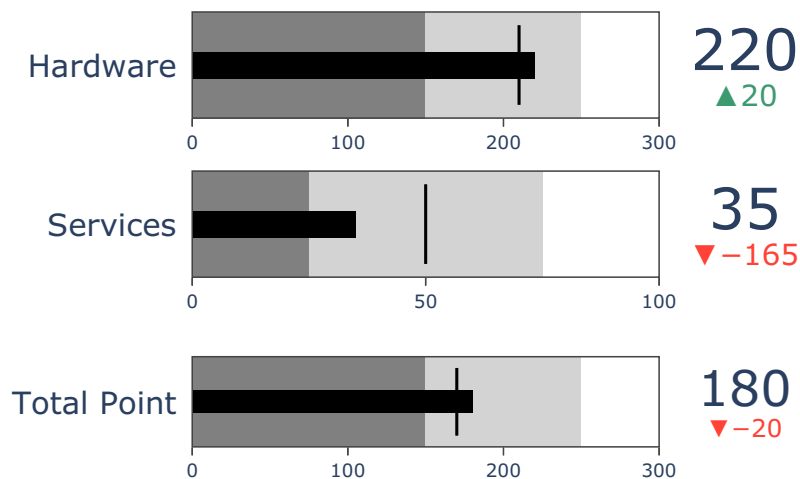
fig = go.Figure()

fig.add_trace(go.Indicator(
    mode = "number+gauge+delta", value = 180,
    delta = {'reference': 200},
    domain = {'x': [0.25, 1], 'y': [0.08, 0.25]},
    title = {'text': "Total Point"},
    gauge = {
        'shape': "bullet",
        'axis': {'range': [None, 300]},
        'threshold': {
            'line': {'color': "black", 'width': 2},
            'thickness': 0.75,
            'value': 170},
        'steps': [
            {'range': [0, 150], 'color': "gray"},
            {'range': [150, 250], 'color': "lightgray"}],
        'bar': {'color': "black"}}))

fig.add_trace(go.Indicator(
    mode = "number+gauge+delta", value = 35,
    delta = {'reference': 200},
    domain = {'x': [0.25, 1], 'y': [0.4, 0.6]},
    title = {'text': "Services"},
    gauge = {
        'shape': "bullet",
        'axis': {'range': [None, 100]},
        'threshold': {
            'line': {'color': "black", 'width': 2},
            'thickness': 0.75,
            'value': 50},
        'steps': [
            {'range': [0, 25], 'color': "gray"},
            {'range': [25, 75], 'color': "lightgray"}],
        'bar': {'color': "black"}}))

fig.add_trace(go.Indicator(
    mode = "number+gauge+delta", value = 220,
    delta = {'reference': 200},
    domain = {'x': [0.25, 1], 'y': [0.7, 0.9]},
    title = {'text': "Hardware"},
    gauge = {
        'shape': "bullet",
        'axis': {'range': [None, 300]},
        'threshold': {
            'line': {'color': "black", 'width': 2},
            'thickness': 0.75,
            'value': 210},
        'steps': [
            {'range': [0, 150], 'color': "gray"},
            {'range': [150, 250], 'color': "lightgray"}],
        'bar': {'color': "black"}}))

fig.show()
```



In []:

In [5]:

```
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib.ticker import FuncFormatter

%matplotlib inline
```

In [6]:

```
def bulletgraph(data=None, limits=None, labels=None, axis_label=None, title=None,
                size=(5, 3), palette=None, formatter=None, target_color="gray",
                bar_color="black", label_color="gray"):
    """ Build out a bullet graph image
    Args:
        data = List of labels, measures and targets
        limits = list of range valules
        labels = list of descriptions of the limit ranges
        axis_label = string describing x axis
        title = string title of plot
        size = tuple for plot size
        palette = a seaborn palette
        formatter = matplotlib formatter object for x axis
        target_color = color string for the target line
        bar_color = color string for the small bar
        label_color = color string for the limit label text
    Returns:
        a matplotlib figure
    """
    # Determine the max value for adjusting the bar height
    # Dividing by 10 seems to work pretty well
    h = limits[-1] / 10

    # Use the green palette as a sensible default
    if palette is None:
        palette = sns.light_palette("green", len(limits), reverse=False)

    # Must be able to handle one or many data sets via multiple subplots
    if len(data) == 1:
        fig, ax = plt.subplots(figsize=size, sharex=True)
    else:
        fig, axarr = plt.subplots(len(data), figsize=size, sharex=True)

    # Add each bullet graph bar to a subplot
    for idx, item in enumerate(data):
        # Get the axis from the array of axes returned when the plot is created
```

```

if len(data) > 1:
    ax = axarr[idx]

    # Formatting to get rid of extra marking clutter
    ax.set_aspect('equal')
    ax.set_yticklabels([item[0]])
    ax.set_yticks([1])
    ax.spines['bottom'].set_visible(False)
    ax.spines['top'].set_visible(False)
    ax.spines['right'].set_visible(False)
    ax.spines['left'].set_visible(False)

    prev_limit = 0
    for idx2, lim in enumerate(limits):
        # Draw the bar
        ax.barh([1], lim - prev_limit, left=prev_limit, height=h,
                color=palette[idx2])
        prev_limit = lim
    rects = ax.patches
    # The last item in the list is the value we're measuring
    # Draw the value we're measuring
    ax.barh([1], item[1], height=(h / 3), color=bar_color)

    # Need the ymin and max in order to make sure the target marker
    # fits
    ymin, ymax = ax.get_ylim()
    ax.vlines(
        item[2], ymin * .9, ymax * .9, linewidth=1.5, color=target_color)

# Now make some labels
if labels is not None:
    for rect, label in zip(rects, labels):
        height = rect.get_height()
        ax.text(
            rect.get_x() + rect.get_width() / 2,
            -height * .4,
            label,
            ha='center',
            va='bottom',
            color=label_color)
if formatter:
    ax.xaxis.set_major_formatter(formatter)
if axis_label:
    ax.set_xlabel(axis_label)
if title:
    fig.suptitle(title, fontsize=14)
fig.subplots_adjust(hspace=0)

```

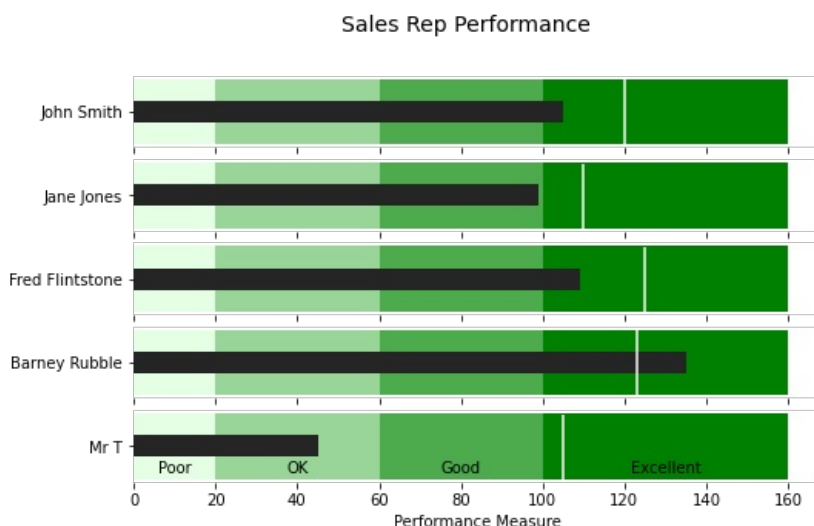
In [7]:

```

data_to_plot2 = [("John Smith", 105, 120),
                 ("Jane Jones", 99, 110),
                 ("Fred Flintstone", 109, 125),
                 ("Barney Rubble", 135, 123),
                 ("Mr T", 45, 105)]

bulletgraph(data_to_plot2, limits=[20, 60, 100, 160],
            labels=["Poor", "OK", "Good", "Excellent"], size=(8,5),
            axis_label="Performance Measure", label_color="black",
            bar_color="#252525", target_color='#f7f7f7',
            title="Sales Rep Performance")

```



In []: