

In []:

Donut Chart- Donut Grafiği:

Tanım:

Donut grafiği birden fazla değişkenin bir bütün içerisindeki oranını belirtmek için kullanılan grafik türüdür. Her bir porsiyon bir değişkeni temsil eder. Genellikle % değerleri göstermek için kullanılır. Pasta grafiğinden farkı sadece görsel farktır.

```
In [2]: import numpy as np
import matplotlib.pyplot as plt

fig, ax = plt.subplots(figsize=(6, 3), subplot_kw=dict(aspect="equal"))

recipe = ["225 g flour",
          "90 g sugar",
          "1 egg",
          "60 g butter",
          "100 ml milk",
          "1/2 package of yeast"]

data = [225, 90, 50, 60, 100, 5]

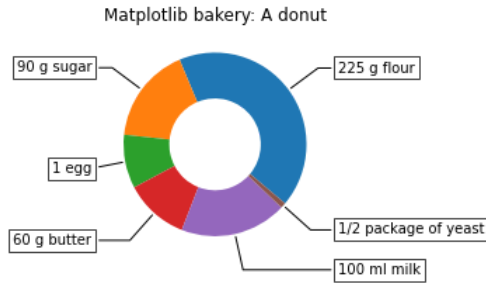
wedges, texts = ax.pie(data, wedgeprops=dict(width=0.5), startangle=-40)

bbox_props = dict(boxstyle="square,pad=0.3", fc="w", ec="k", lw=0.72)
kw = dict(arrowprops=dict(arrowstyle="-"),
          bbox=bbox_props, zorder=0, va="center")

for i, p in enumerate(wedges):
    ang = (p.theta2 - p.theta1)/2. + p.theta1
    y = np.sin(np.deg2rad(ang))
    x = np.cos(np.deg2rad(ang))
    horizontalalignment = {-1: "right", 1: "left"}[int(np.sign(x))]
    connectionstyle = "angle,angleA=0,angleB={}".format(ang)
    kw["arrowprops"].update({"connectionstyle": connectionstyle})
    ax.annotate(recipe[i], xy=(x, y), xytext=(1.35*np.sign(x), 1.4*y),
                horizontalalignment=horizontalalignment, **kw)

ax.set_title("Matplotlib bakery: A donut")

plt.show()
```



Ne için Kullanılır:

Donut grafiği birden fazla değişkenin bir bütün içerisindeki oranını belirtmek için kullanılan grafik türüdür. Her bir porsiyon bir değişkeni temsil eder. Genellikle % değerleri göstermek için kullanılır.

Çok bariz farklarda kullanılabilir veya bir porsiyondaki veriler yanda daha detaylı gösterilebilir .

Bu örnek pasta grafiğinden verildiğinden aynı örnek verilmeyecektir.

Çeşitleri:

Donut grafiğinin yaygın kullanılan bir alt çeşidi mevcuttur.

Nested Pie Chart: Nested Pie grafiği birden fazla değişkenin bir bütün içerisindeki oranını belirtmenin ötesinde alt kategorileri de belirtmek için kullanılan grafik türüdür. Her bir porsiyon bir değişkeni temsil eder. Genellikle % değerleri göstermek için kullanılır. Alt kategoriler ana kategorinin içinde gösterilerek grafik detaylandırılır.

Nested pie Chart grafiği daha ileride anlatılacağından burada detaya girilmeyecektir.

```
In [3]: import numpy as np
import matplotlib.pyplot as plt

#The effect of the donut shape is achieved by setting a width to the pie's wedges through the wedgeprops argument.

fig, ax = plt.subplots()

size = 0.3
vals = np.array([[60., 32.], [37., 40.], [29., 10.]])

cmap = plt.get_cmap("tab20c")
outer_colors = cmap(np.arange(3)*4)
inner_colors = cmap([1, 2, 5, 6, 9, 10])

ax.pie(vals.sum(axis=1), radius=1, colors=outer_colors,
       wedgeprops=dict(width=size, edgecolor='w'))

ax.pie(vals.flatten(), radius=1-size, colors=inner_colors,
       wedgeprops=dict(width=size, edgecolor='w'))

ax.set(aspect="equal", title='Nested Pie Plot')
plt.show()
```

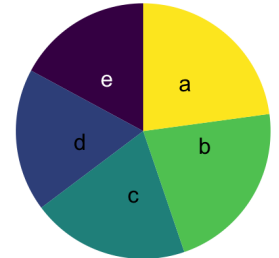
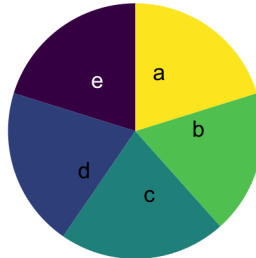
Nested Pie Plot



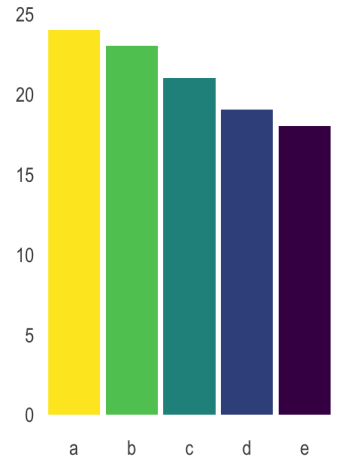
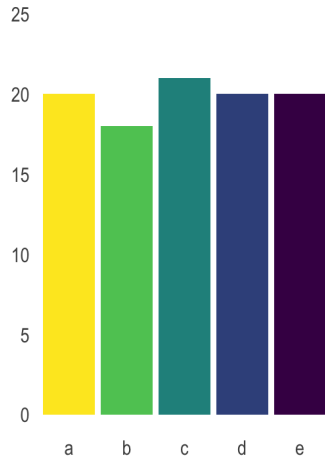
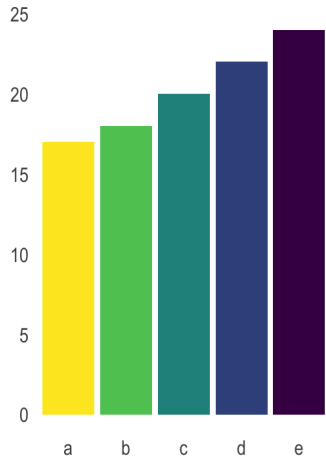
Kullanım Hataları:

Pie Chart ile benzer problemlere sahip olup en önemli sorun insan gözünün açılırlar okumada oldukça beceriksiz olmasıdır. Değerleri birbirine yakın olan yüzdelerde aradaki farkı görmek oldukça sorunludur.

Bu görsel **Pasta grafiğidir** genel görselidir. Şekilde görüldüğü üzere grafikler arası fark neredeyse yok gibidir.



Bu görsel **Pasta grafiğindeki ayrıntıları daha net gösteren bar grafiği** genel görselidir. Yukarıdaki grafik Bar Chart olarak çizildiğinde arada bariz farklar olduğu görülmektedir.



In []: