DataCraft

7

Tortendiagramm & Donut

Tortendiagramm

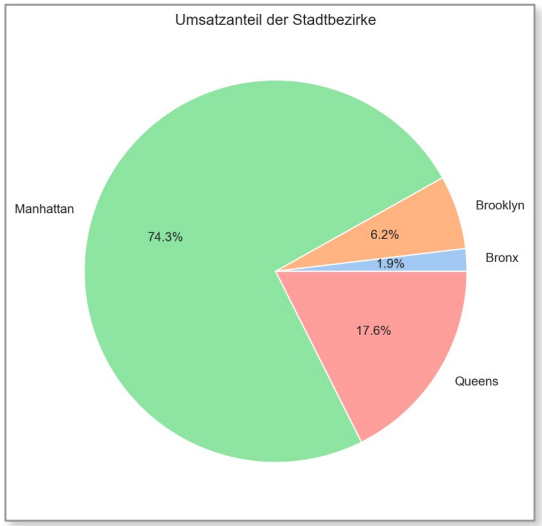
Tortendiagramme sind dafür geeignet, einige Anteile an einem Ganzen darzustellen. Die Anzahl Tortenstücke sollte nicht zu groß sein.

Seaborn unterstützt keine Tortendiagramme, daher muss man auf die matplotlib-Funktion plt.pie() ausweichen.

```
werte = taxis.groupby("pickup_borough")["total"].sum()

sns.set_style("white")
farben = sns.color_palette('pastel')

fig, ax = plt.subplots(figsize=(7,7))
plt.pie(x = werte,
        labels = df.index,
        colors = farben,
        autopct = '%1.1f%%')
plt.title("Umsatzanteil der Stadtbezirke")
ax.set_facecolor("white")
```

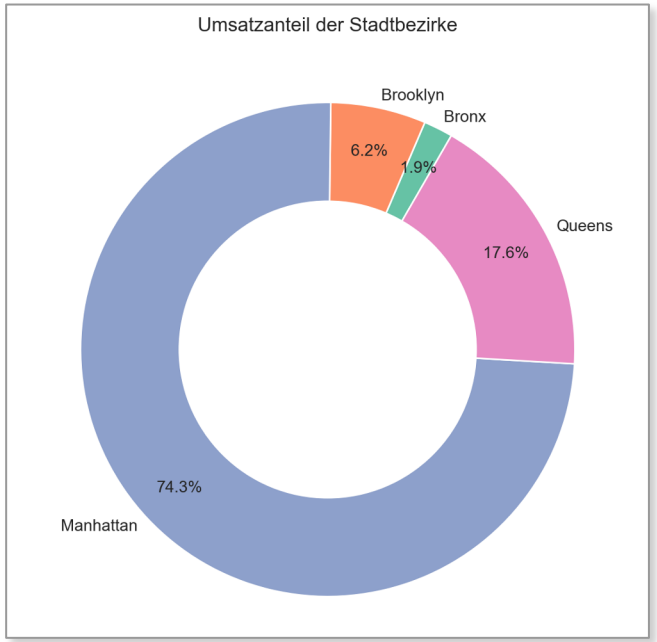


Donut

Ein Donut-Diagramm ist ein Tortendiagramm, welches ein Loch in der Mitte hat. Zeichnen kann man sie mit Matplotlib, indem man eine kleinere Größe für die Kuchenstücke (wedges) auswählt, als bei „radius“ angegeben ist. (z.b. radius = 1, wedge_props = dict(width=0.7))

```
fig, ax = plt.subplots(figsize=(7,7))
# hier sind noch einige Parameter gesetzt
plt.pie(x = werte,
        labels = df.index,
        wedge_props = {„width“: 0.7},
        colors = sns.color_palette('Set2'),
        autopct = '%1.1f%%',
        pctdistance = 0.82,
        labeldistance = 1.05,
        startangle = 60)
plt.title("Umsatzanteil der Stadtbezirke")

# einen weißen Kreis in die Mitte zeichnen
kreis=plt.Circle( (0,0), 0.6, color='white')
ax.add_artist(kreis)
```



DataCraft

8
Heatmap

Heatmap

Heatmaps werden verwendet, um eine metrische Variable in Beziehung zu zwei weiteren (meistens ordinalen) Variablen zu setzen.

Dabei werden die zwei Variablen auf x- und y-Achse aufgetragen und der Farbwert wird durch die eine metrische Variable bestimmt. Dabei wird die Farbskala so gewählt, dass ein gradueller Übergang von einer Farbe zu einer anderen. Häufig ist es der Verlauf von weiß zu einer Farbe.

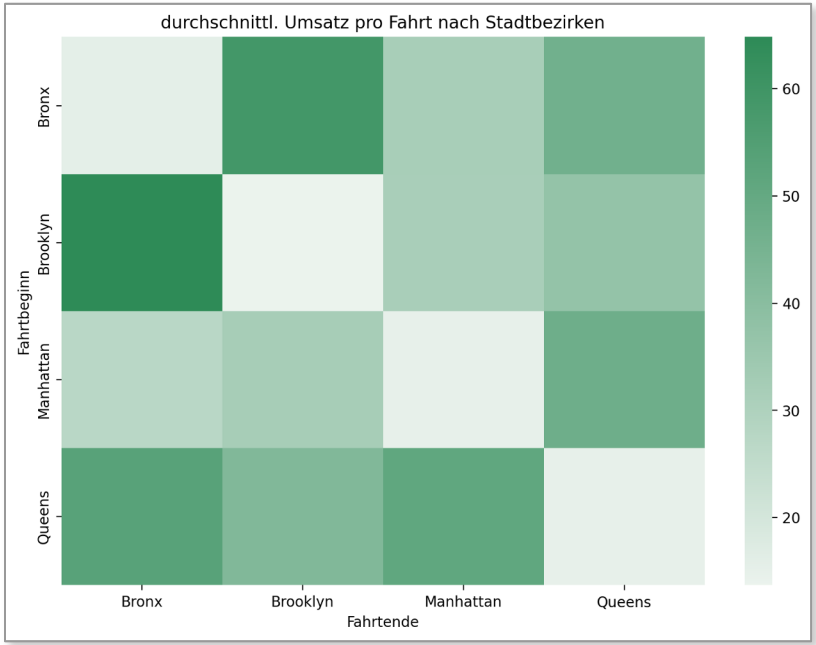
Eine Variante ist die Einfärbung von Landkarten entsprechend der Farbskala.

DataCraft

Seaborn hat die Funktion **sns.heatmap()**.

```
passagiere_bezirk = pd.pivot_table(
                    data=taxis,
                    values="total",
                    index="pickup_borough",
                    columns="dropoff_borough",
                    aggfunc="mean")

farbpalette = sns.light_palette(
                    "seagreen",
                    as_cmap=True)
fig, ax = plt.subplots(figsize=(10,7))
sns.heatmap(
                    data=passagiere_bezirk,
                    cmap=farbpalette)
ax.set(
                    title="durchschnittl. Umsatz pro Fahrt nach Stadtbezirken",
                    xlabel="Fahrtende",
                    ylabel="Fahrtbeginn")
```



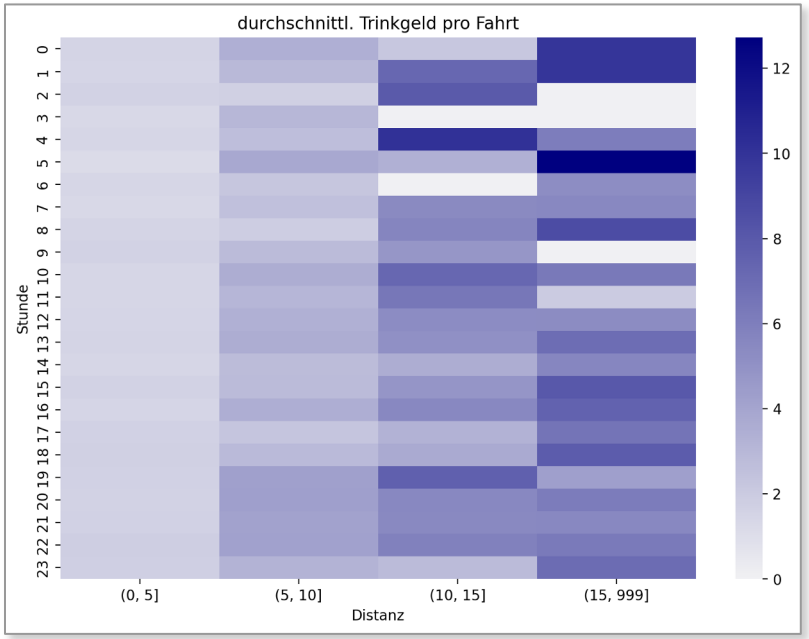
DataCraft

Eine Heatmap muss nicht symmetrisch sein

```
taxis["pickup_hour"] = taxis["pickup"].dt.hour
taxis["dist_cat"] = pd.cut(taxis["distance"],
                    bins=[0,5,10,15,999])

tips_hour_dist = pd.pivot_table(data=taxis,
                    values="tip", index="pickup_hour",
                    columns="dist_cat", aggfunc="mean")
tips_hour_dist[tips_hour_dist.isna()] = 0

farbpalette = sns.light_palette((0.0,0.0,0.5),
                    as_cmap=True)
fig, ax = plt.subplots(figsize=(10,7))
sns.heatmap(data=tips_hour_dist, cmap=farbpalette)
ax.set(title="durchschnittl. Trinkgeld pro Fahrt",
                    xlabel="Distanz",
                    ylabel="Stunde")
```



DataCraft

9

Pairplots

Pairplots

- Pairplots eignen sich, um die Verteilung mehrerer Variablen auf einmal darzustellen
- Es handelt sich um eine „Figure-level Function“, welche direkt ein ganzes Plot Gitter erzeugen kann.
- In Pairplots können Scatterplots, KDE-Plots oder Histogramme dargestellt werden. Auch Regressionen sind möglich.

```
data = sns.load_dataset("mpg")

sns.pairplot(
        data,
        vars=["mpg", "displacement",
              "weight", "acceleration"]
)
```