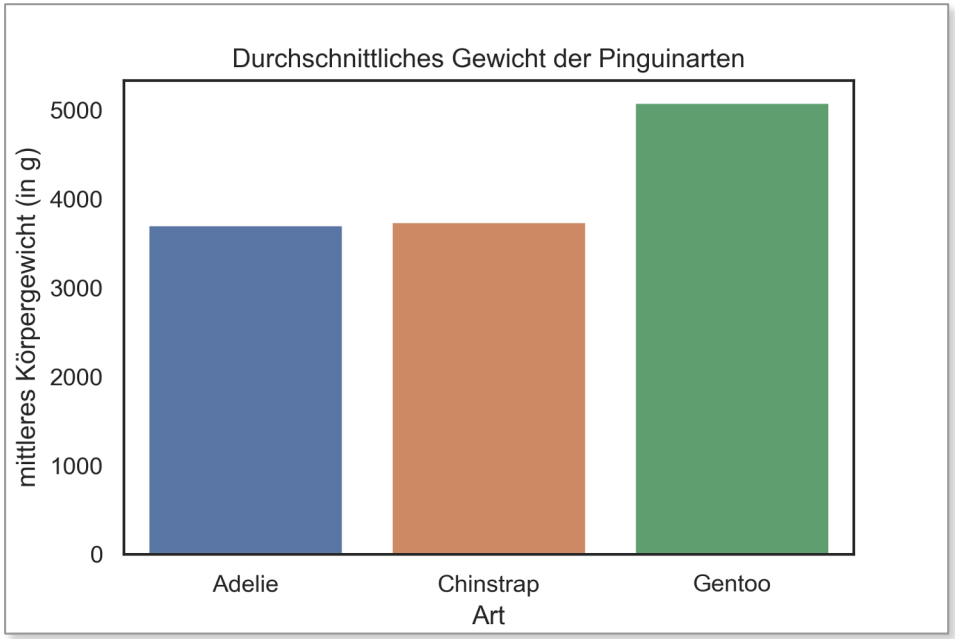
DataCraft

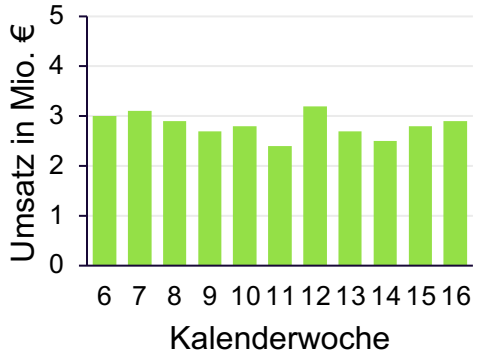
3

Säulen- & Balkendiagramme

Säulendiagramme (engl. column chart) erlauben es, einen Wert für mehrere Kategorien darzustellen. Die Höhe der Säule entspricht dem Wert.



Säulendiagramme werden auch für einen zeitlichen Verlauf eingesetze



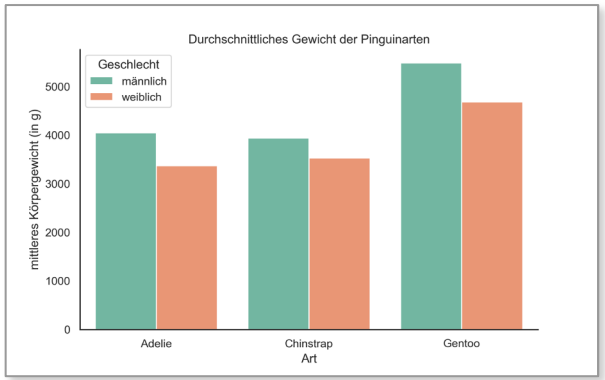
Seaborn hat die Funktion **barplot()**, die ein Säulendiagramm erzeugt

```
import pandas as pd
import seaborn as sns

penguins = sns.load_dataset("penguins")

# Einen passenden DataFrame erzeugen
df = penguins.groupby("species").mean()
df.reset_index(inplace=True)

sns.barplot(x="species", y="body_mass_g", data=df)

# barplot berechnet automatisch den Mittelwert
sns.barplot(penguins, x="species", y="body_mass_g", ci=None)
```

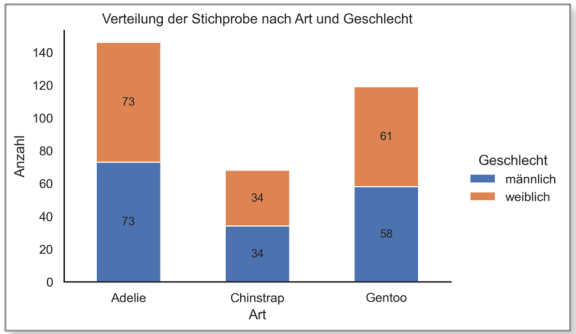
DataCraft

Es gibt drei Varianten des Säulendiagramms, um noch eine weitere Dimension darzustellen

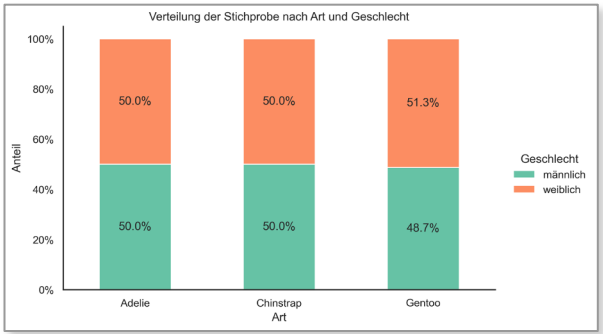
gruppiertes Säulendiagramm



gestapeltes Säulendiagramm



gestapeltes prozentuales Säulendiagramm

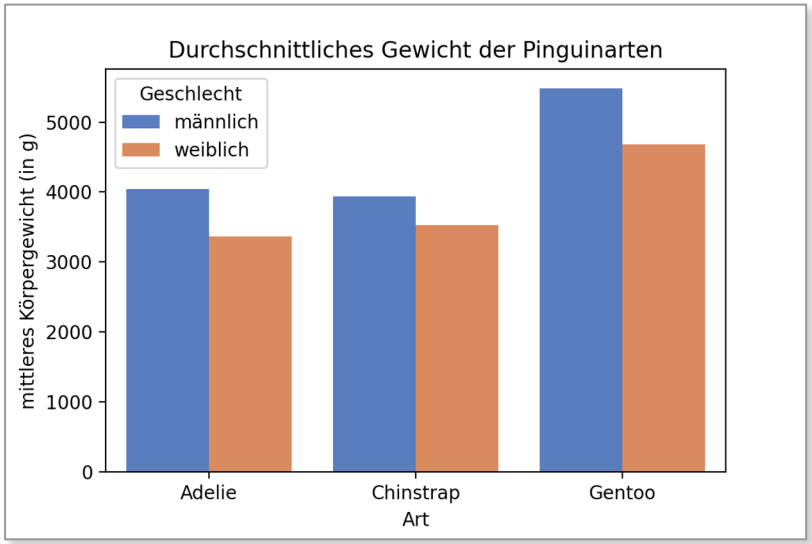


Gruppierte Säulendiagramme mit seaborn

DataCraft

Für ein gruppiertes Säulendiagramm wird einfach die weitere Dimension als Farbe (hue) hinzugefügt.

```
sns.barplot(penguins, x="species", y="body_mass_g", hue="sex", errorbar=None)
```



Anpassen der Beschriftungen

```
ax.set_title("Durchschnittliches
    Gewicht der Pinguinarten")
ax.set_xlabel("Art")
ax.set_ylabel(
    "mittleres Körpergewicht (in g)")
ax.legend(title="Geschlecht",
    labels=["männlich","weiblich"])
```

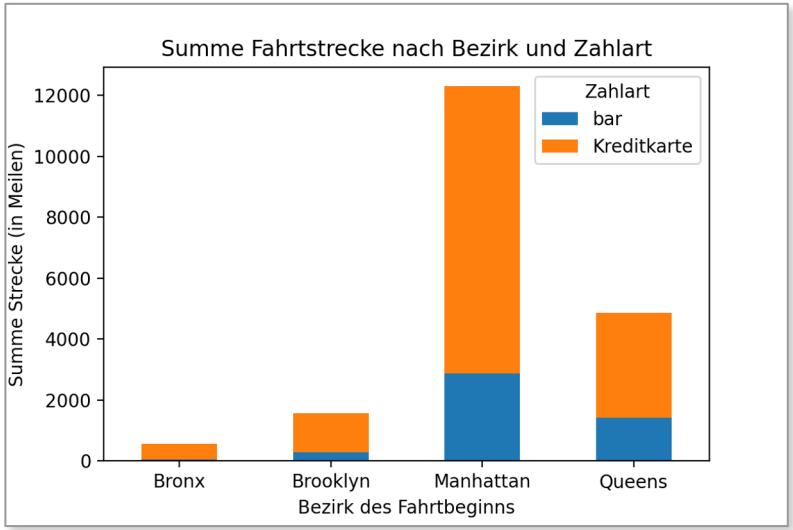
Gestapelte Säulendiagramme mit seaborn

DataCraft

Barplot hat keinen Parameter, um einfach gestapelte Säulendiagramm zu erzeugen. Man kann zwei Barcharts übereinanderlegen. Einfacher geht es mit der pandas-Funktion plot().

Die Daten müssen allerdings in der richtigen Form vorliegen (Gruppierung in zwei Spalten).

```
df = taxis.pivot_table(
    values="distance",
    index="pickup_borough",
    columns="payment",
    aggfunc="sum")
ax = df.plot(
    kind="bar",
    stacked=True,
    rot=0)
```

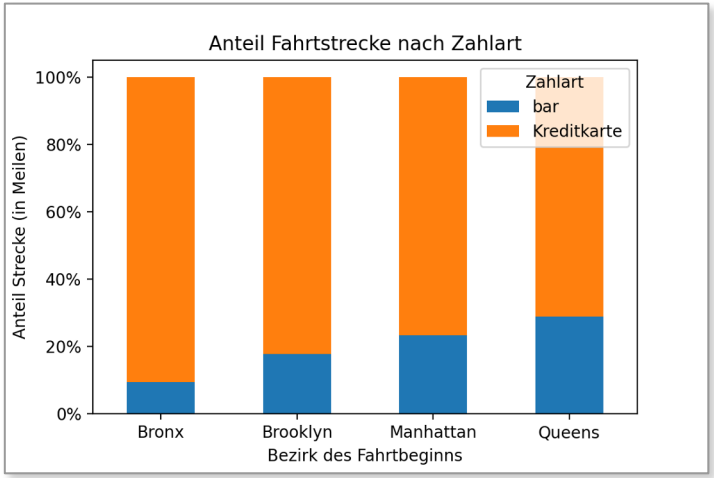


Prozentuale Säulendiagramme mit seaborn

DataCraft

Um ein prozentuales, gestapeltes Säulendiagramm zu bekommen, müssen die Daten in Anteile umgerechnet werden. Mit dem PercentFormatter aus matplotlib kann die Achse in % dargestellt werden.

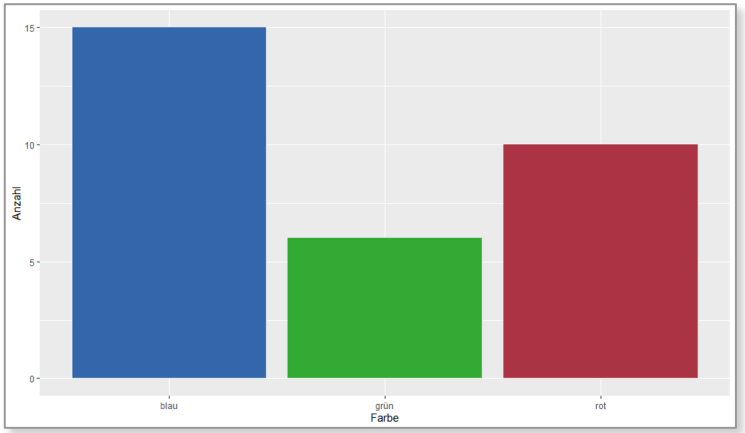
```
df["cash_proz"] = 100 * df["cash"]/(df["cash"] + df["credit card"])
df["credit_proz"] = 100 - df["cash_proz"]

ax = df[["cash_proz","credit_proz"]]
  .plot(kind="bar", stacked=True, rot=0)
ax.yaxis.set_major_formatter(
  mtick.PercentFormatter())

# Alternativ überschreiben der tickmarks
ax.set_yticklabels(
   [f"{x:.0%}" for x in ax.get_yticks()])
```



DataCraft

Das Histogramm ist ein spezielles Säulendiagramm, bei dem die Balkenhöhe der Anzahl Vorkommnisse entspricht. Die Anzahl kann als absolute Zahl oder relativ als Prozentzahl angegeben werden.

- Für kategorielle Variablen wird einfach gezählt, wie häufig ein Wert vorkommt.
- Für metrische Variablen werden Intervalle gebildet und dann gezählt, wie viele Werte es in jedem der Intervalle gibt.

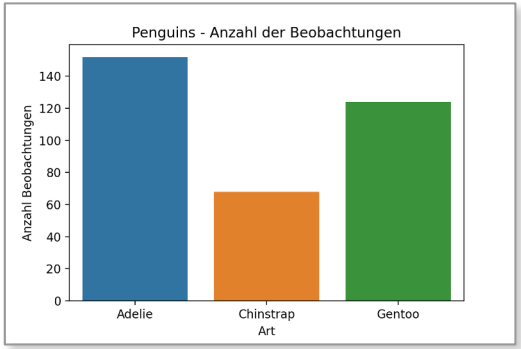


Histogramme mit seaborn

DataCraft

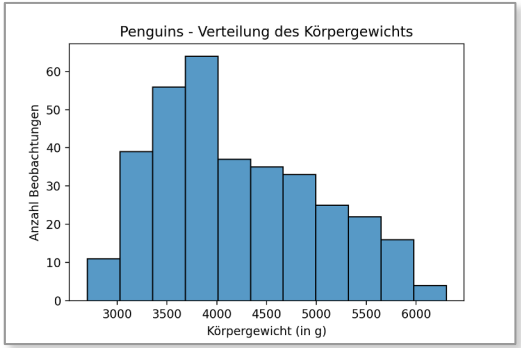
Für kategorielle Variablen wird die Funktion **countplot()** verwendet

sns.countplot(x="species" data=df)



Für metrische Variablen wird die Funktion **histplot()** verwendet

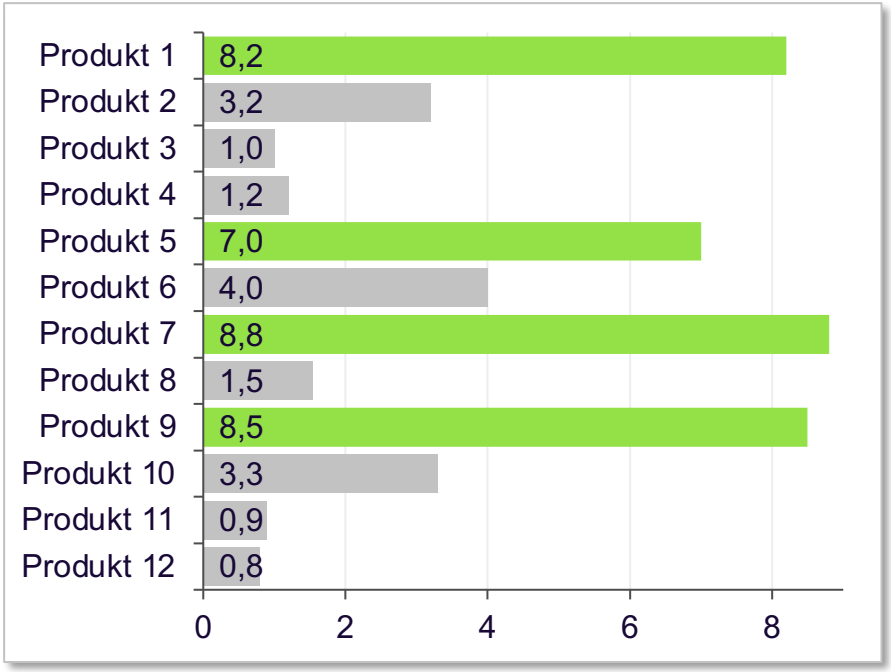
sns.histplot(x="body_mass_g", data=penguins)



Ein um 90° gedrehtes Säulendiagramm nennt man ein Balkendiagramm (engl. barchart).

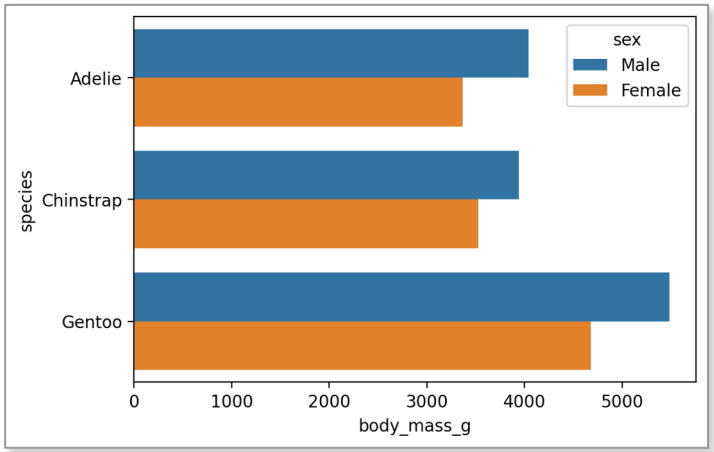
- Häufig wird der Begriff Balkendiagramm/barchart auch ein Säulendiagramme verwendet

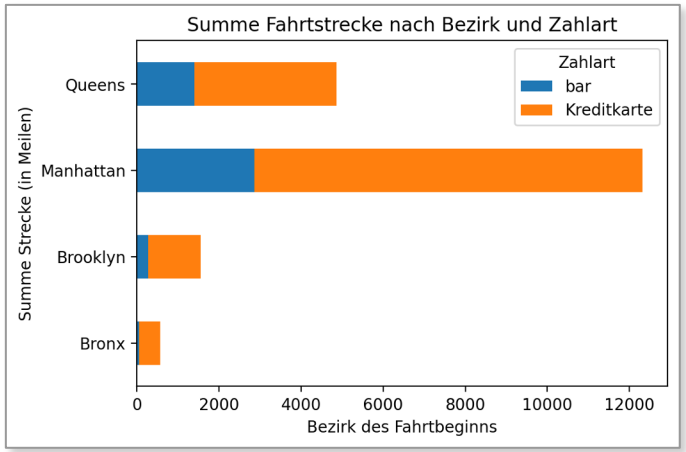
- Balkendiagramme sind Säulendiagrammen vorzuziehen, wenn
  - die Kategoriebezeichnungen lang sind
  - es viele Kategorien gibt



Ein Balkendiagramm ist nur ein um 90° gedrehtes Säulendiagramm. Das erreicht man in seaborn durch Vertauschen der Achsen. Pandas hat dafür den Typ barh.

```
ax = sns.barplot(x="body_mass_g", y="species", hue="sex", ci=None,          data=penguins)
ax = df.plot(kind="barh", stacked=True, rot=0)
```





DataCraft

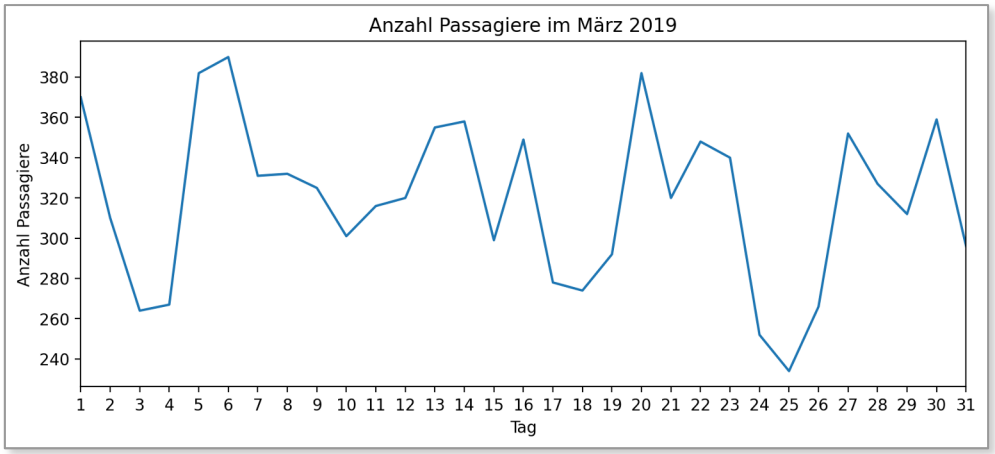
4

Linien- & Flächencharts

Mit Liniendiagrammen wird üblicherweise ein zeitlicher Verlauf dargestellt

Seaborn hat dafür die Funktion **lineplot()**. Es werden einfach die Spalten für die x-Achse und die y-Achse übergeben. Natürlich müssen die Daten in passender Form vorliegen.

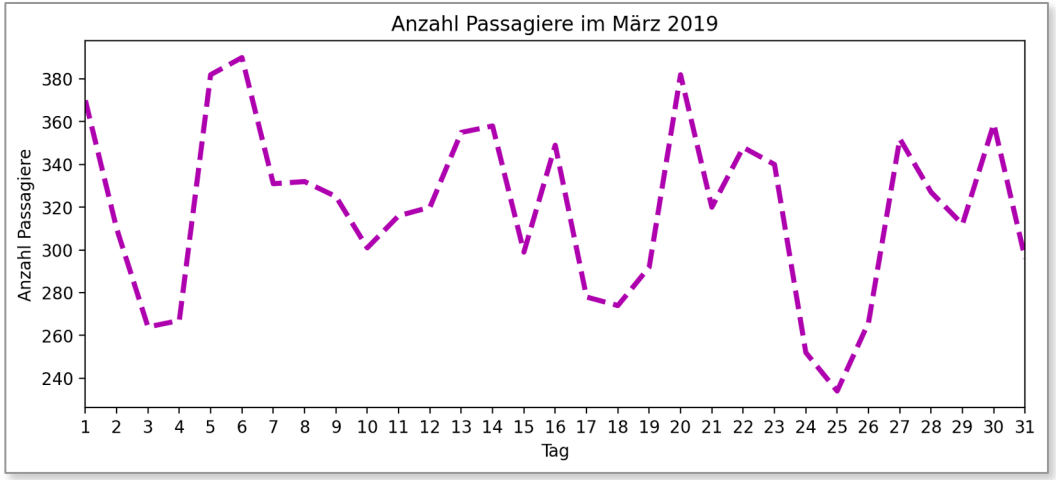
```
ax = sns.lineplot(x="tag", y="passengers", data=taxis_tag)
```



DataCraft

Anpassen von Farbe und Stil

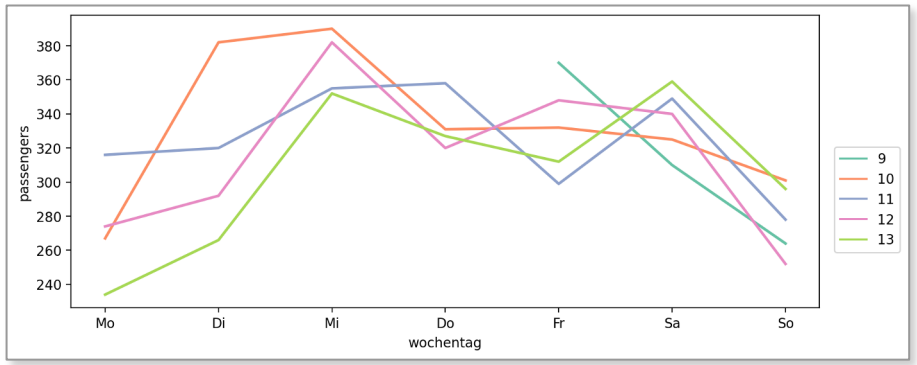
```
plt.figure(figsize=(10,4))
ax = sns.lineplot(x = "tag",y="passengers", data=pro_tag2, color="#AF00AF", linewidth=3, linestyle="--")
ax.set(title="Anzahl Passagiere im März 2019", xlabel="Tag", ylabel="Anzahl Passagiere")
ax.set_xlim(1, 31)
ax.set_xticks(range(1,32))
```



Liniendiagramm

Es können auch mehrere Linien in ein Chart eingezeichnet werden. Dafür wird z.B. die Färbung (hue) einer Spalte zugeordnet

```
pro_tag2["kalenderwoche"] = pro_tag2["pickup_date"].dt.week
pro_tag2["wochentag"] = pro_tag2["pickup_date"].dt.dayofweek
plt.figure(figsize=(10,4))
ax = sns.lineplot(x ="wochentag" ,y="passengers", hue="kalenderwoche",data=pro_tag2,
                  palette="Set2", linewidth=2)
plt.legend(bbox_to_anchor=(1.02, 0.55), loc='upper left', borderaxespad=0)
ax.set_xticks(range(7))
ax.set_xticklabels(["Mo","Di","Mi","Do","Fr","Sa","So"])
```



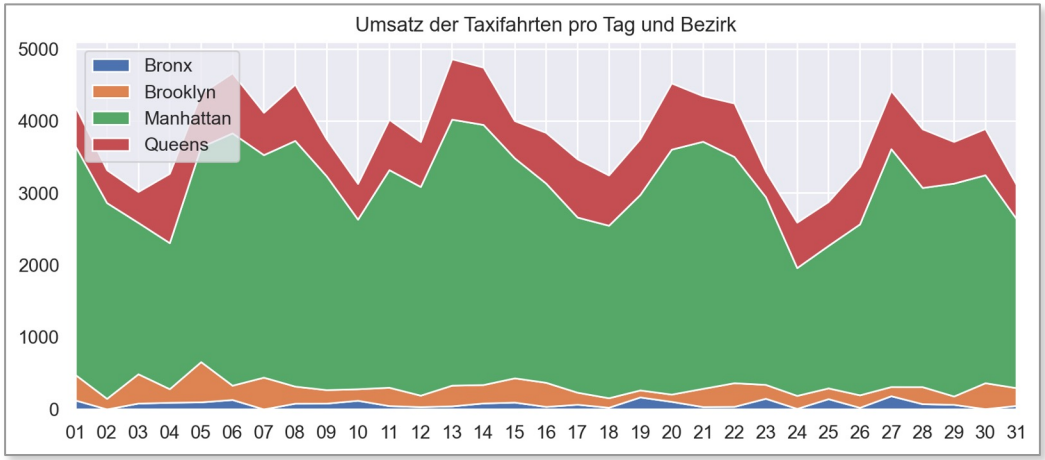
Flächendiagramm

DataCraft

Seaborn selber unterstützt keine Flächencharts, man muss auf matplotlib zurückgreifen. Dort gibt es die Funktion *plt.stackplot()*, die mehrere Spalten übereinander darstellt.

```
plt.stackplot(df["pickup_date"], df["Bronx"], df["Brooklyn"], df["Manhattan"], df["Queens"],
    labels = ["Bronx","Brooklyn","Manhattan","Queens"])
```

- Um mehrere Spalten aus den Ausprägungen einer Variable zu erzeugen, kann die pandas-Funktion *pd.pivot_table()* verwendet werden
- Die x-Achse kann über einen DateFormatter formatiert werden
- siehe Codebeispiel

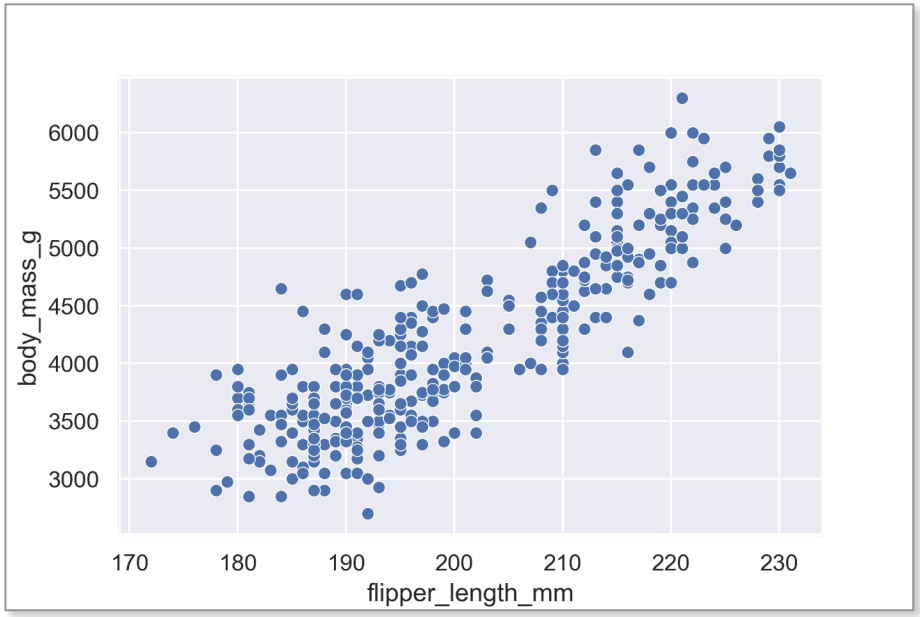


DataCraft

5
Punktwolken / Scatterplots

DataCraft

Mit einer Punktwolke lässt sich die Beziehung zwischen zwei metrischen Variablen visualisieren



Punktwolke / Scatterplot

DataCraft

Seaborn hat dafür die Funktion **scatterplot()**. Es werden einfach die Spalten für die x-Achse und die y-Achse übergeben.

```
ax = sns.scatterplot(x="bill_length_mm", y="body_mass_g", data=penguins)
```

Das Punktsymbol kann über den Parameter marker gesetzt werden

```
ax = sns.scatterplot(x="bill_length_mm", y="body_mass_g", marker="D",              data=penguins)
```

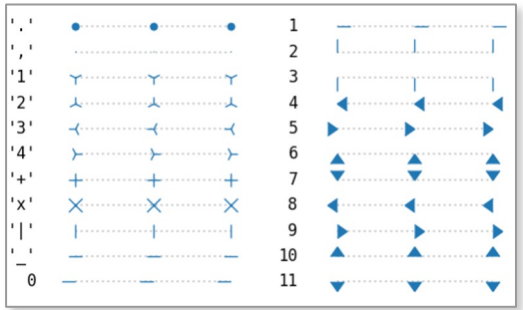
Punktwolke / Scatterplot - Markers

Es gibt verschiedene Marker zur Auswahl. Man kann sich diese sogar selber konstruieren (siehe https://matplotlib.org/stable/gallery/lines_bars_and_markers/marker_reference.html)

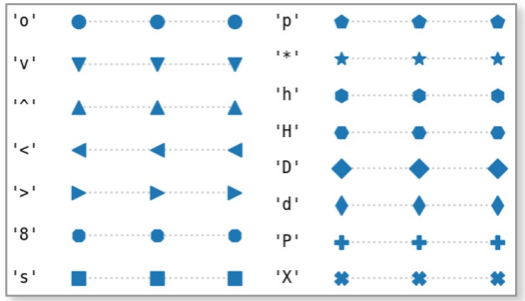
Die Größe wird mit *s* gesetzt. Die Füllfarbe mit *color*, die Randfarbe mit *edgecolor*.

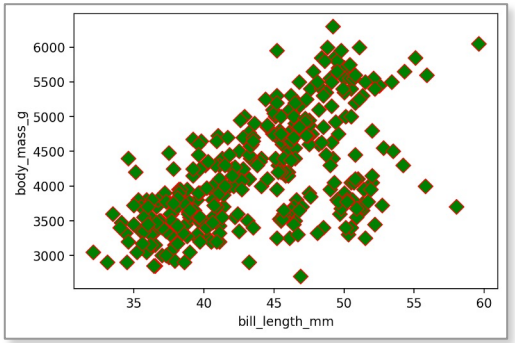
```
ax = sns.scatterplot(x="bill_length_mm", y="body_mass_g", marker="D", s=70, color="g",
edgecolor="r",data=penguins)
```

nicht gefüllte Marker



gefüllte Marker





Über Farbe, Form und/oder Größe der Punkte lassen sich auch weitere Dimensionen abbilden

Die Farbe lässt sich wie bei Säulendiagrammen über **hue** variieren, der Markertyp über **style**, die Größe über **size**.

```
ax = sns.scatterplot(
  x="bill_length_mm",
  y="body_mass_g",
  hue="species",
  style="species",
  data=penguins)
```