

視覚化 (matplotlib, seaborn)

2022/12/16

国立遺伝学研究所 大量遺伝情報研究室

望月孝子

謝辞

本講習の内容は、

2019年 先進ゲノム支援(PAGS)、DDBJ、DBCLS合同情報解析講習会
視覚化 (matplotlib, seaborn) (農研機構・孫建強)

農学生命情報科学特論Ⅰ@東京大学アグリバイオインフォマティクス教育研究ユニット
視覚化 (matplotlib, seaborn) (農研機構・孫建強)

実験医学別冊 独習 Pythonバイオ情報解析 (ISBN 978-4-7581-2249-8)
第8章 データの可視化

の資料、ソースコードを引用、 改変し使用させて頂いています。

Python による視覚化

Python では視覚化にはライブラリをインストールして使う必要がある。



pandas のデータ構造をグラフ化できる。
簡単だが、複雑なグラフや微調整が難しい。



R の ggplot2 とほぼ同じような使い方で
ほぼ同じような描画が可能。
The grammar of graphics と呼ばれる文法で記述。



Web 上に情報量が多い。
複雑なグラフや微調整が可能。



Web ベースのインタラクティブなグラフを作成。



matplotlib をベースにしている。
matplotlib よりも綺麗に描画できる。



Web ベースのインタラクティブなグラフを作成。
The grammar of graphics で記述。

Python による視覚化

Python では視覚化にはライブラリをインストールして使う必要がある。



pandas のデータ構造をグラフ化できる。
簡単だが、複雑なグラフや微調整が難しい。

本日の講習



Web 上に情報量が多い。
複雑なグラフや微調整が可能。



matplotlib をベースにしている。
matplotlib よりも綺麗に描画できる。



R の ggplot2 とほぼ同じような使い方で
ほぼ同じような描画が可能。
The grammar of graphics と呼ばれる文法で記述。



Web ベースのインタラクティブなグラフを作成。



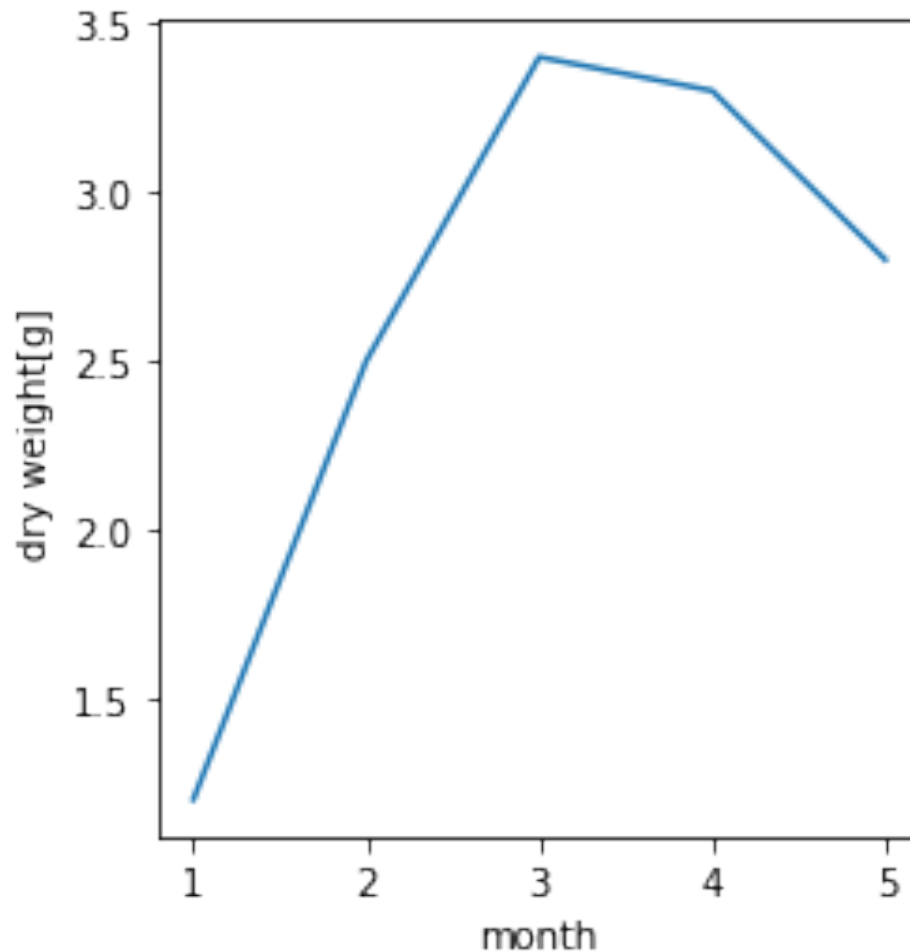
Web ベースのインタラクティブなグラフを作成。
The grammar of graphics で記述。

matplotlib グラフの作成

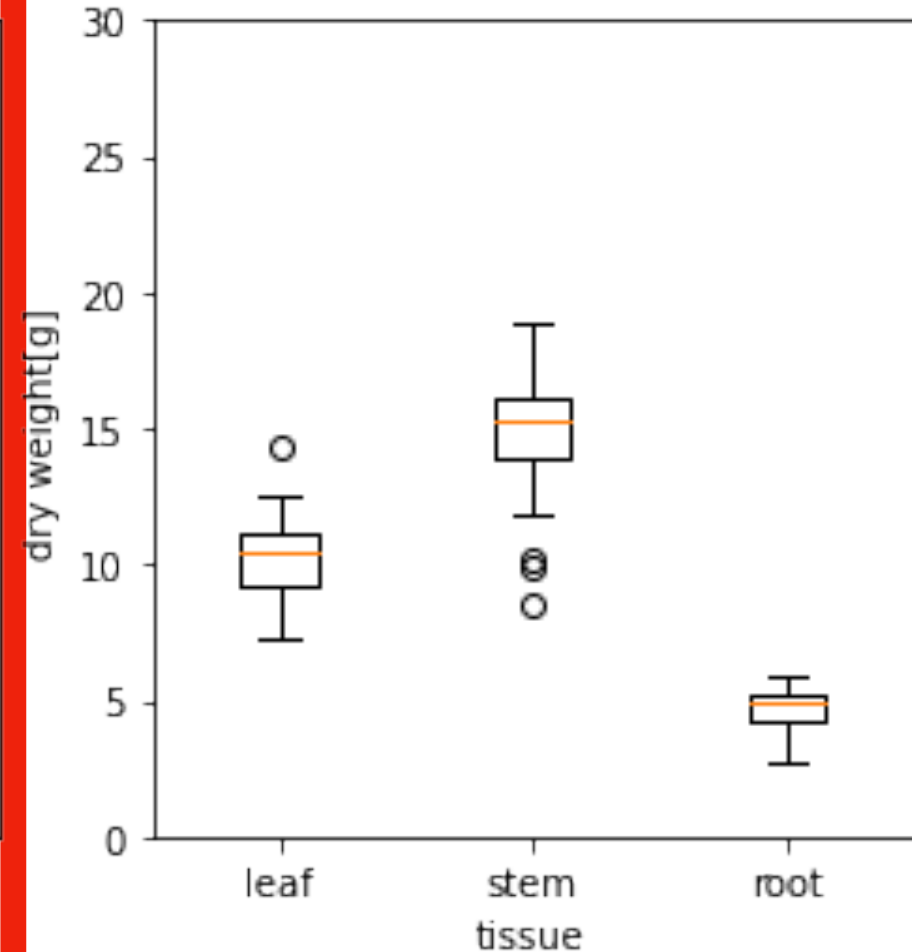
matplotlib グラフの作成

Figure クラス = ページ

Axes クラス = 枠



Axes クラス = 枠

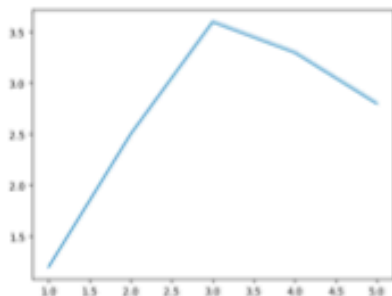


プロット領域を分割して、それぞれをクラスとして定義して使用する。

→ Jupyter notebook

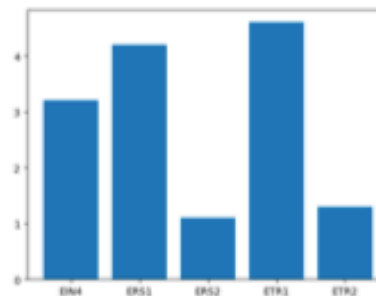
基本グラフ

基本グラフ



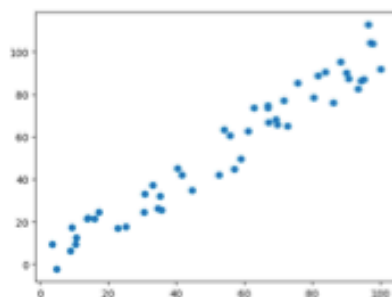
line chart

データの時系列的な変化傾向を視覚化するとき目的で使われる。



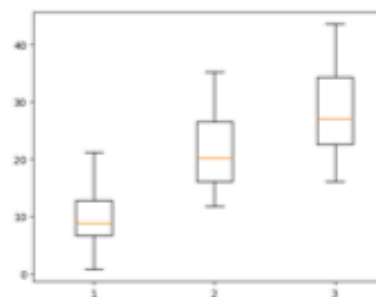
bar chart

カテゴリ毎に分類されるデータを視覚化するとき目的で使われる。



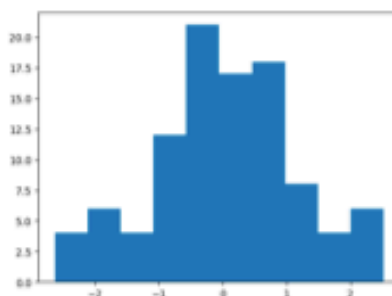
scatter chart

2 変量の連続値データ同士の相関や分布などを視覚する目的で使われる。



boxplot

カテゴリ毎に分類されるデータの分布などを視覚化するとき目的で使われる。



histogram

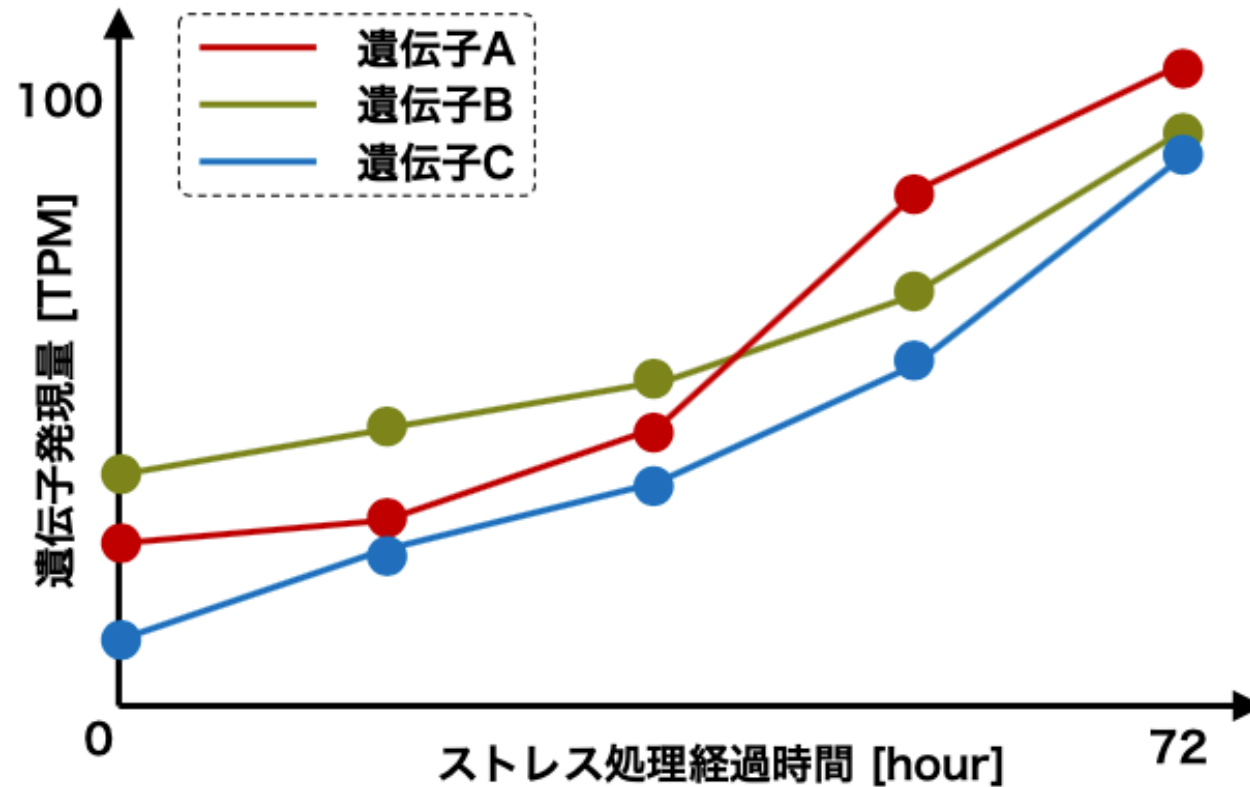
1 変量の連続値データの分布などを視覚化するとき目的で使われる。



pie chart

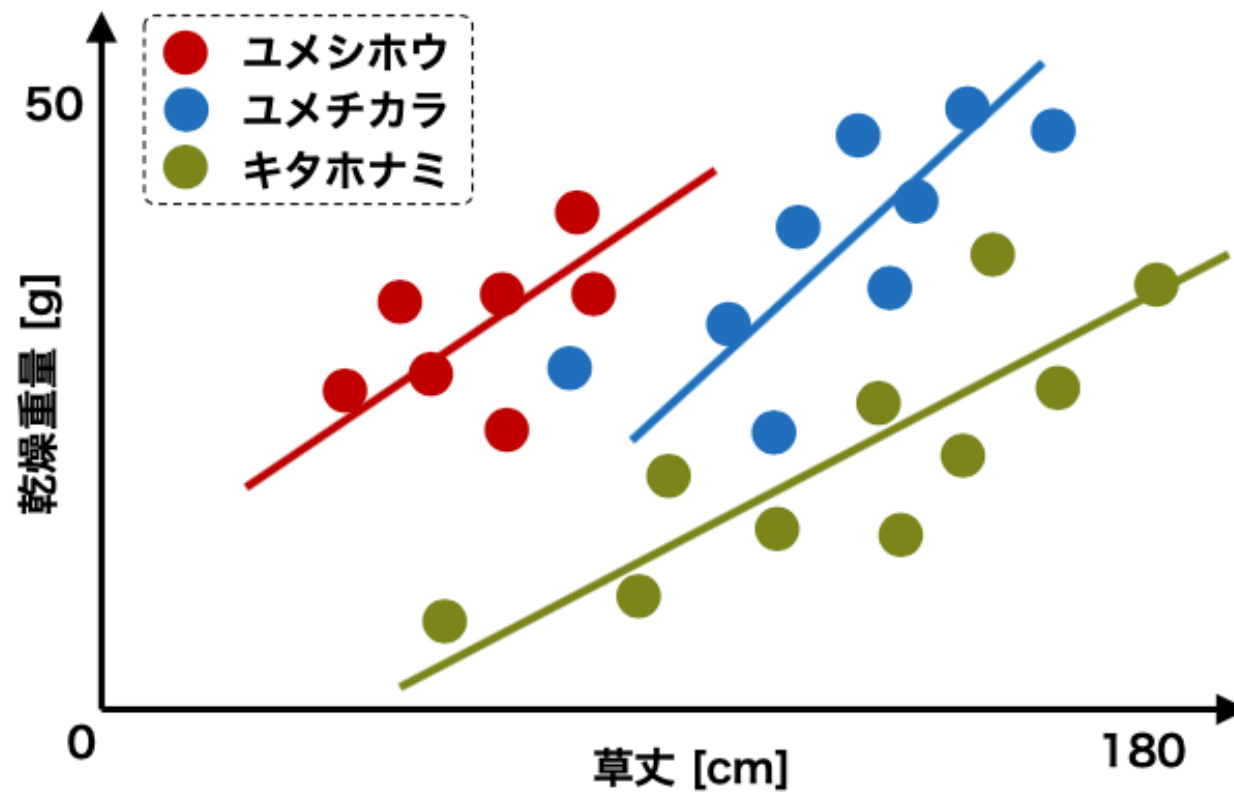
割合データを視覚する目的で使われる。

折れ線グラフ



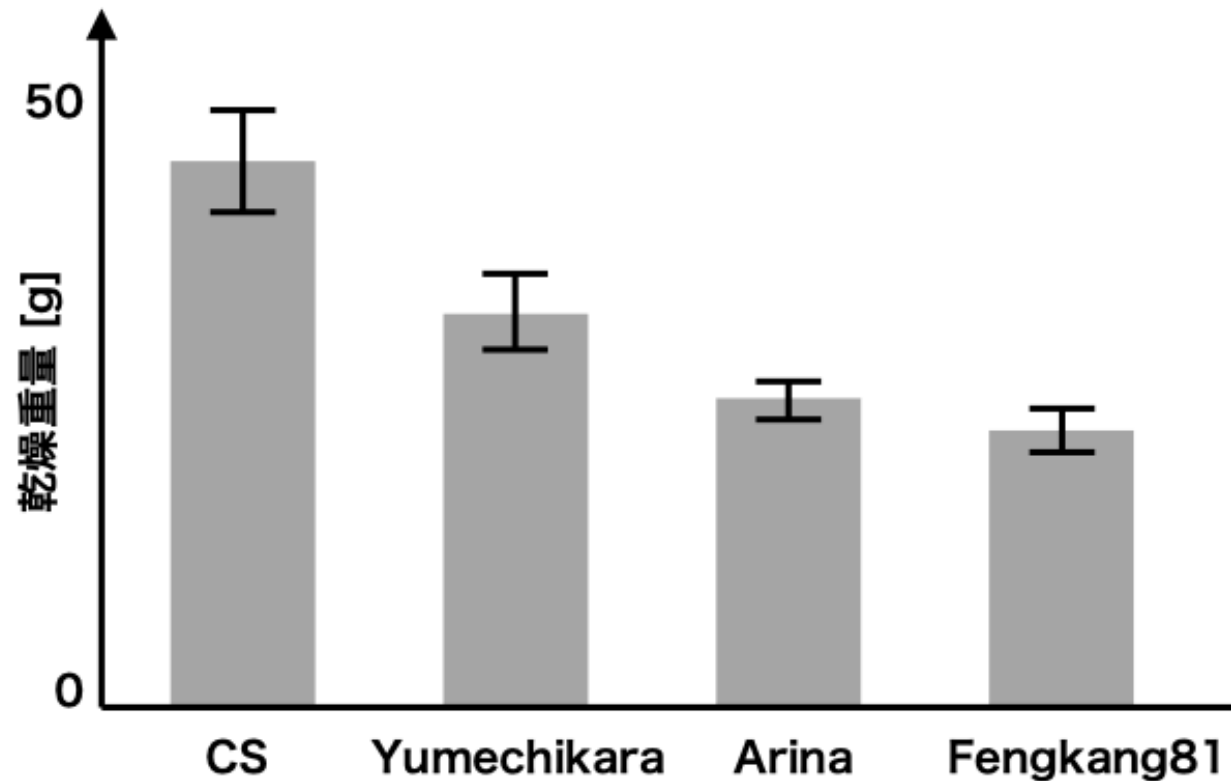
- 線グラフは基本的にデータの系列的な変化を見るためのグラフ
- 縦軸および横軸は連続量
- 原点 (0, 0) が省略されているグラフを十分に注意すること
- 観測値を強調するために、観測値には点を明示することもある

散布図



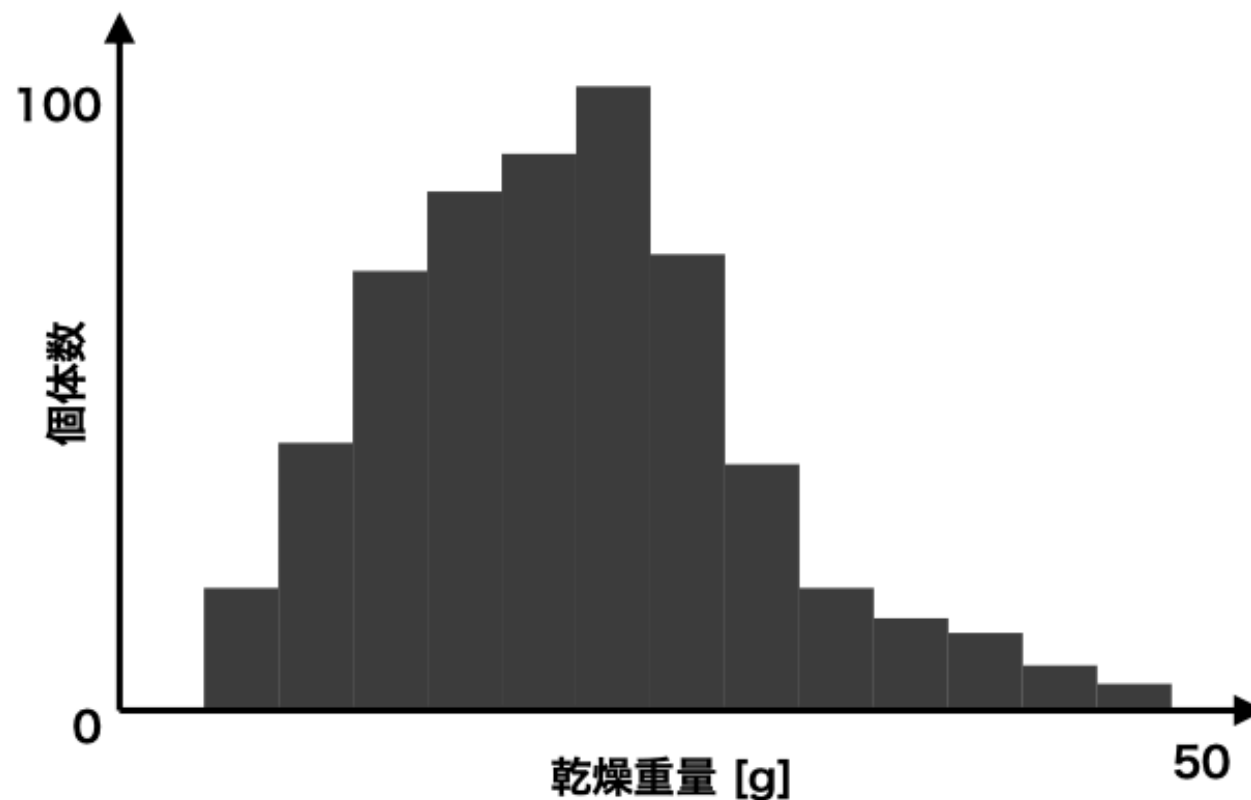
- 連続量からなる多変量
データ同士の相関や分布
などを見るためのグラフ
- 縦軸および横軸は連続量
- 原点 (0, 0) が省略されて
いるグラフを十分に注意
すること
- 回帰直線との併用もよく
見られる

棒グラフ



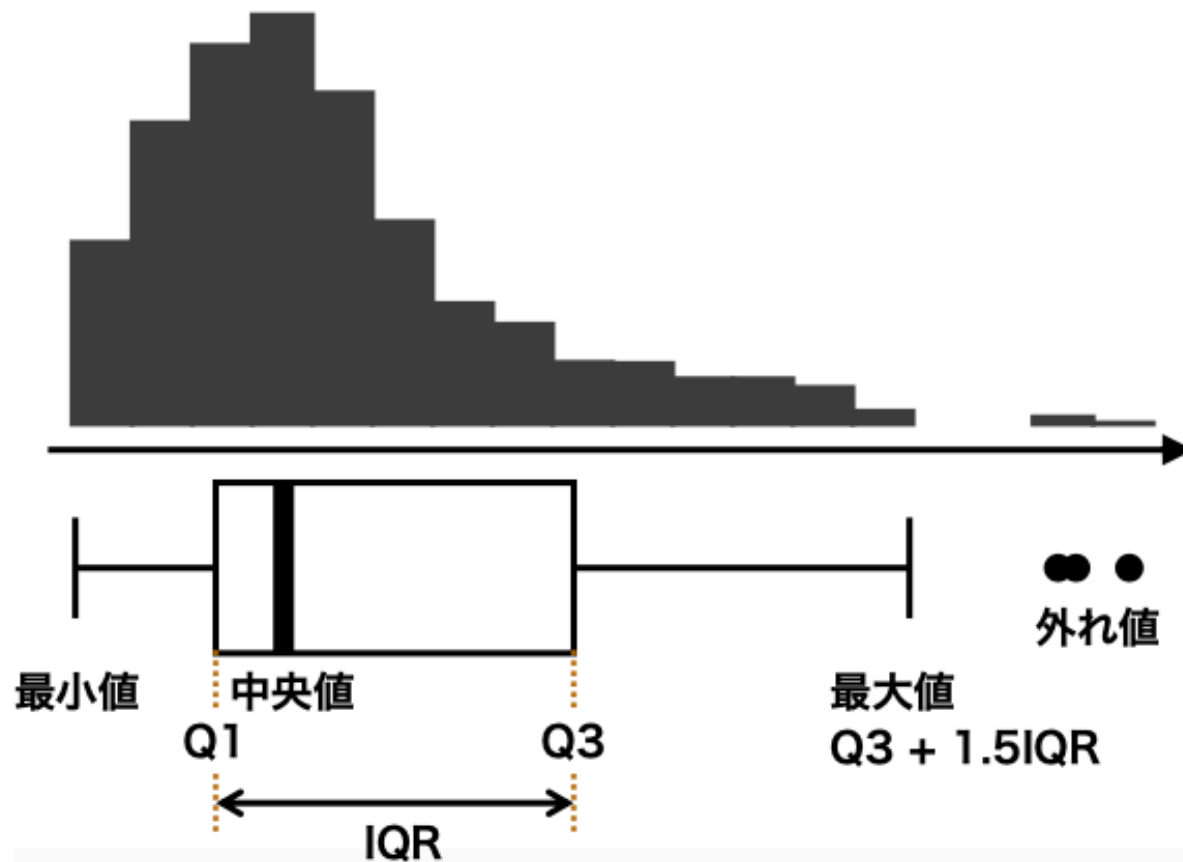
- 離散データを視覚化するためのグラフ
- 横軸が連続量、縦軸が離散量、あるいはその逆
- 原点が省略されているグラフを十分に注意すること
- エラーバーとともに用いられることがある

ヒストグラム



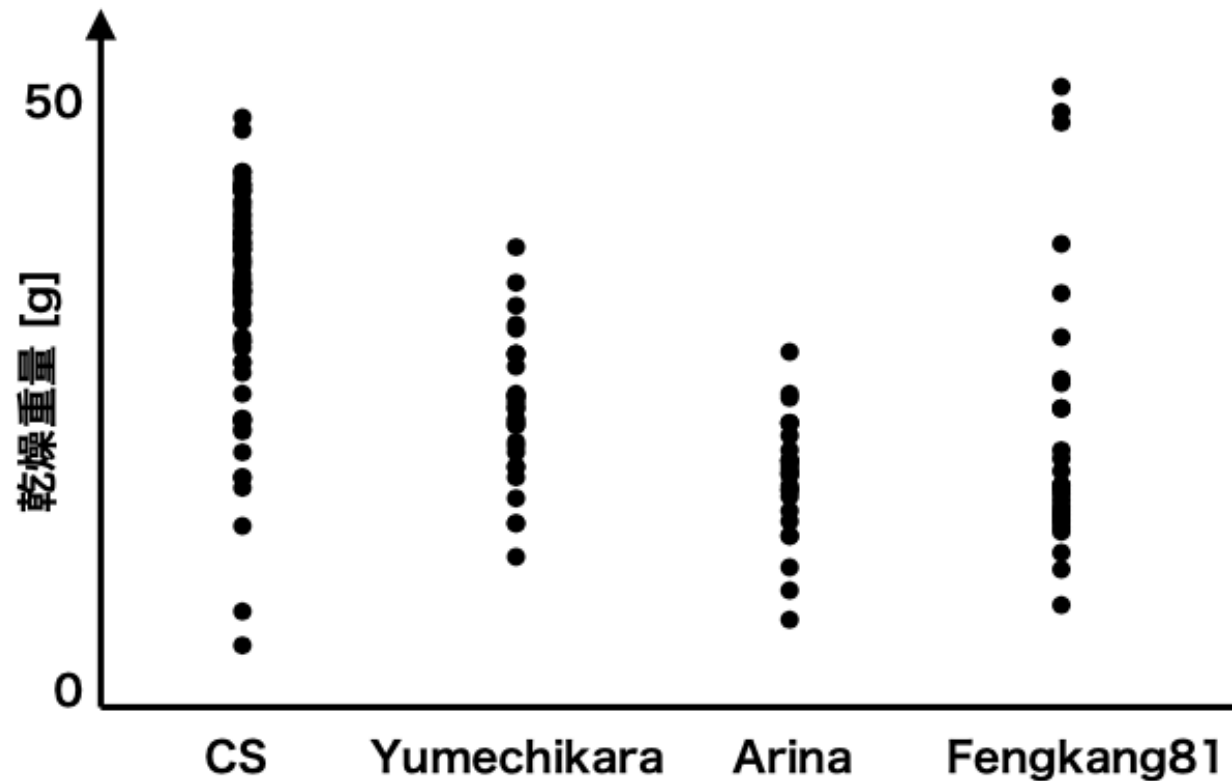
- 1変量の連続値データを視覚化するためのグラフ
- 横軸が連続量であり、縦軸は頻度・個数または確率である
- 横幅は恣意的（経験的）に決められることもあれば、スタージェスの公式などで決めることもある

ボックスプロット



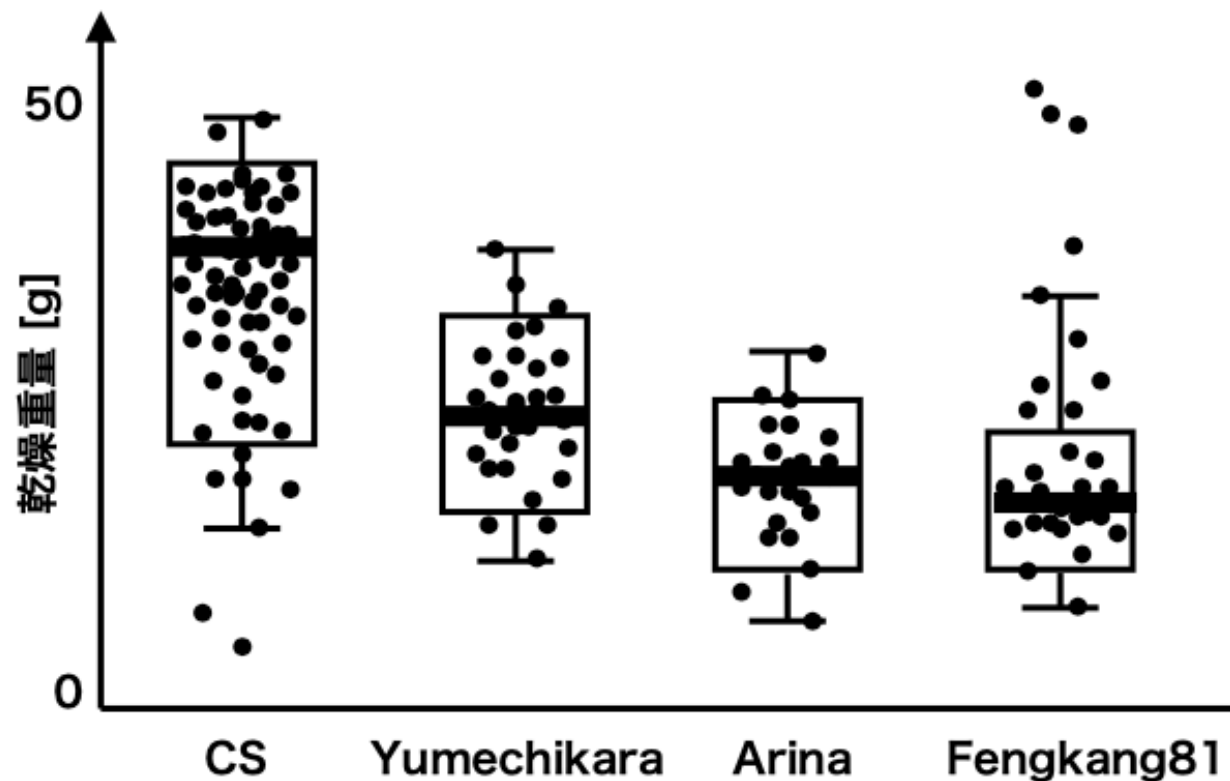
- 1変量の連続値データを視覚化するためのグラフ
- 最大値、最小値、第1四分位数 (Q1)、中央値、第3四分位数 (Q3) などを簡単に確認できる
- $Q3 \pm 1.5(Q3 - Q1)$ 範囲外にあるデータは外れ値と定義される

ジッタープロット



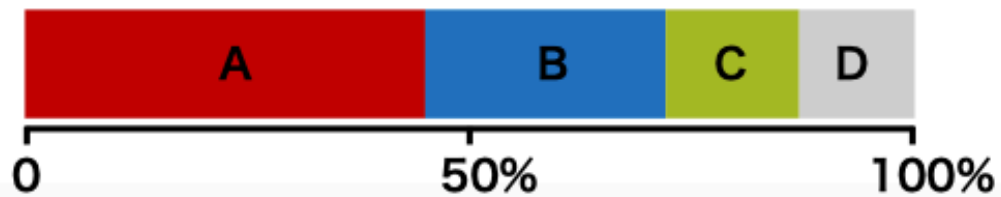
- 複数カテゴリの1変量連続値データの分布を確認ためのグラフ
- 実際の値をプロットすると、データが多いところが重なるため、点を左右にランダムにずらしてプロットする

ジッタープロット + ボックスプロット



- 複数カテゴリの1変量連続値データの分布を確認ためのグラフ
- 実際の値をプロットすると、データが多いところが重なるため、点を左右にランダムにずらしてプロットする
- ボックスプロットと合わせて使われる場合もある

円グラフ



- 円グラフは誤解されやすい、円グラフを用いる前に帯状グラフで代用できないかを確認すること
- 3D円グラフは基本的に人を騙すようなグラフであることに注意

seaborn

seaborn を利用したグラフ作成は、matplotlib と同様な手順で作成できる。

seaborn のインポート

seaborn クラスをインポートして、利用可能な状態にする。

seaborn.set の実行

seaborn.set メソッドを実行して、グラフ描画パラメータなどを seaborn 風書き換える。

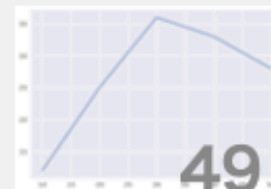
グラフの作成と表示

matplotlib.pyplot のメソッドをそのまま使用してグラフの作成と表示を行う。

```
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
sns.set()
```

```
x = np.array([1.0, 2.0, 3.0, 4.0, 5.0])
y = np.array([1.2, 2.5, 3.4, 3.3, 2.8])
```

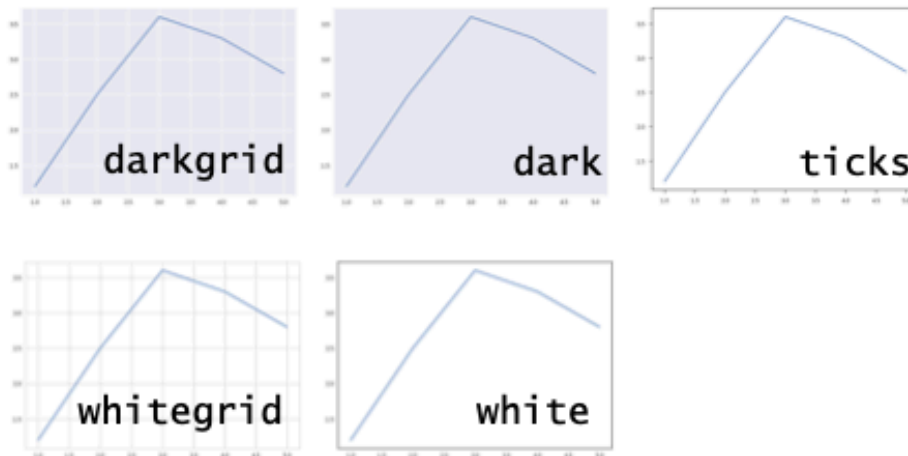
```
fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
ax.plot(x, y)
fig.show()
```



seaborn スタイルシート

グラフのスタイルを設定する。

seaborn ではスタイルシートが用意されている。スタイルシートを書き換えることで、グラフの描画エリアのスタイルを変更することができる。



```
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns

sns.set()
sns.set_style('whitegrid')
sns.set_palette('Set1')

x = np.array([1.0, 2.0, 3.0, 4.0, 5.0])
y = np.array([1.2, 2.5, 3.4, 3.3, 2.8])

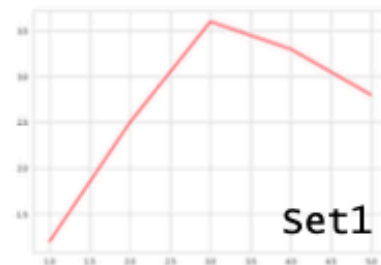
fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
ax.plot(x, y)
fig.show()
```

seaborn カラーパレット

カラーパレットを設定する。

あらかじめ決められた色の組み合わせ。カラーパレットを書き換えることで、グラフ全体の配色パターンを簡単に変更できる。

Set1



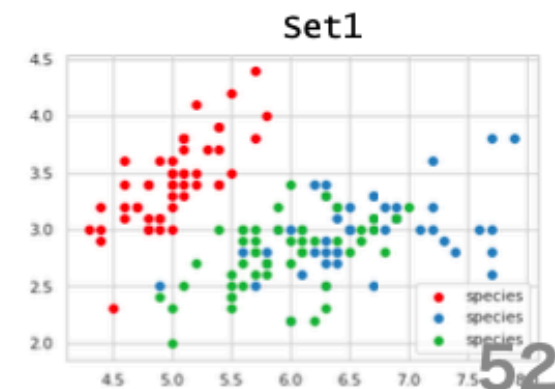
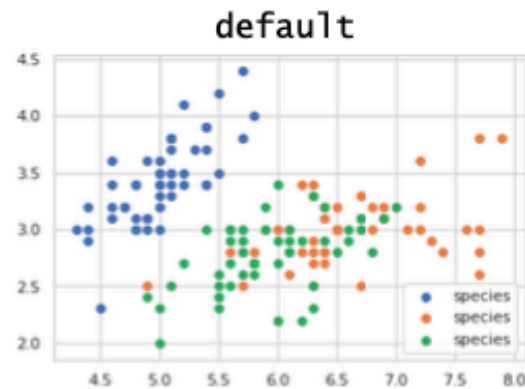
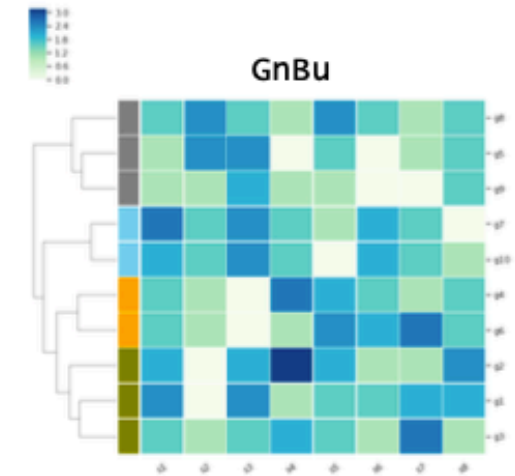
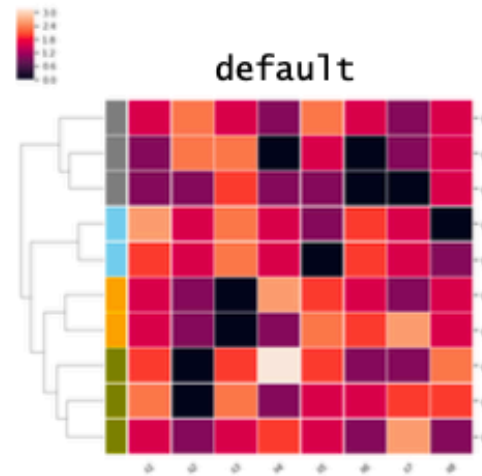
```
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
sns.set()
sns.set_style('whitegrid')
sns.set_palette('Set1')
```

```
x = np.array([1.0, 2.0, 3.0, 4.0, 5.0])
y = np.array([1.2, 2.5, 3.4, 3.3, 2.8])
```

```
fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
ax.plot(x, y)
fig.show()
```

seaborn カラーパレット

あらかじめ決められた色の組み合わせ。カラーパレットを書き換えることで、グラフ全体の配色パターンを簡単に変更できる。



seaborn カラーパレット

あらかじめ決められた色の組み合わせ。カラーパレットを書き換えることで、グラフ全体の配色パターンを簡単に変更できる。seaborn のデフォルトのカラーパレットは `deep` となっている。 `set_palette` メソッドを使うことで、カラーパレットの変更を行うことができる。



→ Jupyter notebook