Pythonプログラミング　中級

2024年12月14日（土）第4回
講師：渡邉 貴之

講座の内容

- タイムテーブル

第1回 （12月7日）	**9:30 ～ 10:15**
	Google Colaboratory環境の確認
第2回 （12月7日）	**10:25 ～ 11:10**
	pandasによるデータの集計と可視化1
第3回 （12月7日）	**11:15 ～ 12:00**
	pandasによるデータの集計と可視化2
第4回 （12月14日）	**9:30 ～ 10:15**
	scikit-learnによる回帰分析
第5回 （12月14日）	**10:25 ～ 11:10**
	MeCabによる自由回答の分析と可視化1
第6回 （12月14日）	**11:15 ～ 12:00**
	MeCabによる自由回答の分析と可視化2

※途中、休憩時間を適宜取ります

2日目の概要

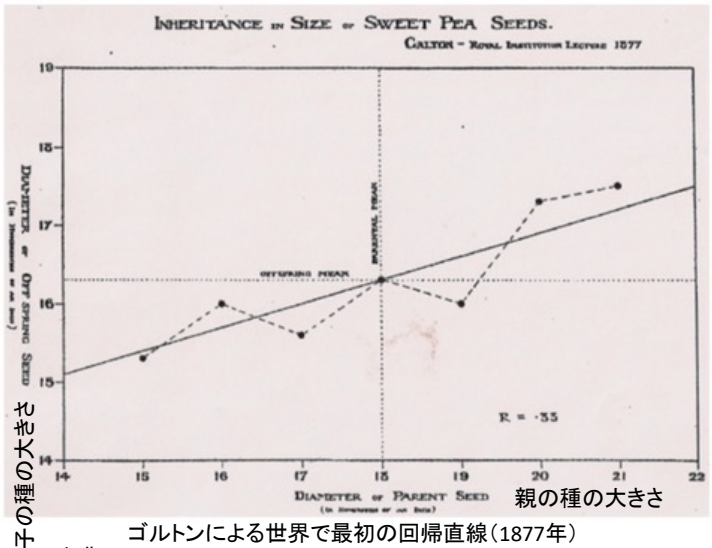
- Pythonプログラミング中級2日目について・・・
  - Python言語の具体的な活用例として、機械学習と自然言語処理を採り上げます。
  - 回帰分析
    - 最も基本的な機械学習の手法である回帰分析について試します
      - 教師データ(入力データと正解データ)をモデルに学習させる
      - 学習したモデルをもとに、未知の入力データの正解を予測する
    - 使用モジュール:scikit-learn/pandas/matplotlib/seaborn
  - 自然言語処理
    - アンケートの自由回答の自然言語処理による分析について試します
      - MeCabモジュールによる形態素解析
      - wordcloudモジュールによる可視化
    - 使用モジュール:MeCab/wordcloud/pandas/matplotlib/
- 参考図書
  - 下山ら, "Python実践データ分析100本ノック", 秀和システム

第4回
scikit-learnによる回帰分析

目次

回帰とは

- 回帰は実数値を予測する教師あり機械学習
  - 語源：ダーウィンの弟子ゴルトン
    - 大きい（小さい）種から育ったスイートピーの種からは大きい（小さい）種が得られるはずと考えたが、実際に得られた種の大きさの平均はある傾向の直線に向かって帰っていった。
    - 回帰直線

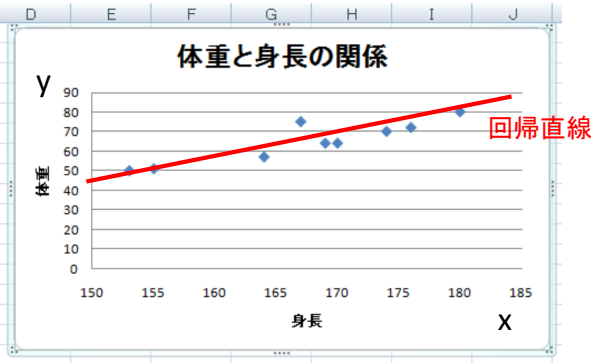


ゴルトンによる世界で最初の回帰直線（1877年）
出典：https://rss.onlinelibrary.wiley.com/doi/full/10.1111/j.1740-9713.2009.00379.x#sign379-sec-0050-title (Royal Statistical Society)

回帰分析

- 回帰分析とは、ある2つの変数にどのような関係があるのかを推定する機械学習による分析手法です。
  - 例えば、あるクラスの生徒について、身長xと体重yの一覧表があったとします。身長xが高い人ほど体重yが大きいといった関係がある場合に、xからy（またはその逆）を推定する式を求めるような分析が回帰分析です。
- 線形回帰モデル
  - 例えば、身長をx、体重をyとした時に、xからyを推定する式が直線の方程式（y = ax + b）で与えられたとします。このような直線のモデル式を線形回帰モデルを呼び、描かれる直線を回帰直線と呼びます。
    - このとき、xを説明変数、yを目的変数と呼び、aは傾きのパラメータ、bは切片のパラメータです。aとbを回帰係数と呼びます。

	A	B	C
1	氏名	身長	体重
2	山田太郎	170	64
3	鈴木陽一	155	51
4	三宅太一	164	57
5	杉浦勉	169	64
6	吉永英輔	180	80
7	加藤肇	153	50
8	大杉美紀久	167	75
9	望月敏夫	176	72
10	牧野翼	174	70



$$y = ax + b$$

線形回帰モデル（直線の方程式）

回帰直線と相関

- 相関とは？・・・（高校数学の、数Iのデータの分析の単元）
  - 回帰直線の傾きが正(直線が右上がり)の場合、2つの量（説明変数と目的変数）には正の相関(相関関係)があるといいます。
  - 回帰直線の傾きが負(直線が右下がり)の場合、2つの量には負の相関があるといいます。
  - 2つの量の分布がバラバラで関係性が見いだせない場合、2つの量は無相関であるといいます。
- 相関係数とは？
  - 相関の目安となる値を「相関係数」と呼びます。
  - 相関係数は-1.0～1.0 の値を取り、相関係数が負の場合は負の相関、正の場合は正の相関、0 に近いの場合は無相関であることを表します。また、絶対値が大きいほど強い相関があるこ とを表します。
- 決定係数（寄与率）とは？
  - 相関係数の2乗を決定係数といい、0.0から1.0までの値をとります。
  - 決定係数が1.0に近いほど、相関が強いことを表しており、説明変数が目的変数をよく説明できていることを意味します。

演習で使用するモジュール

- 下記の表の赤文字のモジュールを使用します。

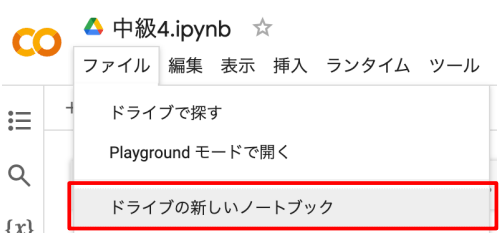
モジュール名	概要
matplotlib	グラフなどの可視化モジュール
seaborn	matplotlibの見栄えをより綺麗にするモジュール
NumPy	計算を効率的に行うためのモジュール
SymPy	数式・記号計算用モジュール
pandas	データ解析支援モジュール（Excelファイルの読み書きが可能）
openpyxl	Excelファイルの読み書きに特化したモジュール
xlwings	Excelアプリを直接制御できるモジュール
scipy	NumPyを利用した数値解析モジュール
scikit-learn	機械学習モジュール
NetworkX	グラフ・ネットワーク計算と可視化モジュール
Basemap	地図描画モジュール
MeCab	形態素解析モジュール
wordcloud	ワードクラウド可視化モジュール

簡単な回帰分析

- 回帰分析の一例として、床面積や築年数から賃貸物件（アパート等）の家賃を推定するような回帰直線を求めてみましょう（データはダミーです）。
- 準備
  - 資料電子データサイトの「kaiki.xlsx」ファイルを使用します。
  - 「Sheet1」シートには、100件の賃貸アパートの床面積、築年数、階数、家賃がまとめられています。

	A	B	C	D
1	床面積	築年数	階数	家賃
2	20.1	5	3	70000
3	15.8	20	5	43000
4	23.1	14	6	57000
5	17.6	15	2	56000

  - Google Colabで新しいノートブックを新規作成してください。



必要なモジュールのインポート

- pandasモジュールと、matplotlib.pyplotモジュールをインポートします。また、日本語フォントをインストールします。

```
#データ解析用モジュール
import pandas as pd
#日本語フォントのインストール
!pip install japanize-matplotlib
#グラフ描画用モジュール
import matplotlib.pyplot as plt
import japanize_matplotlib
```

- 実行すると、日本語フォントのインストールが開始されます。

データのGoogle Colabへのアップロード

- kaiki.xlsxをGoogle Colabにアップロードして、読み込めるようにします。

- 手順
  - Google Colab左のフォルダマークをクリックする
  - ファイルツリーが開くので、資料電子データサイトからダウンロードしたkaiki.xlsxをドラッグ＆ドロップしてアップロードする
    - ドラッグ＆ドロップができない場合はアップロードボタンをクリックしてファイルを選択します
  - アップロード後は、再びフォルダマークをクリックしてファイルツリーは閉じてもよい

データの取り込み

- pandasではExcelファイルを直接読み込むことができます。
  - 新しくコードセルを追加し、下記を記述して実行します。

```
#pandasのインポート
import pandas as pd

file = pd.ExcelFile("kaiki.xlsx")
data = file.parse("Sheet1")
data.head()
```

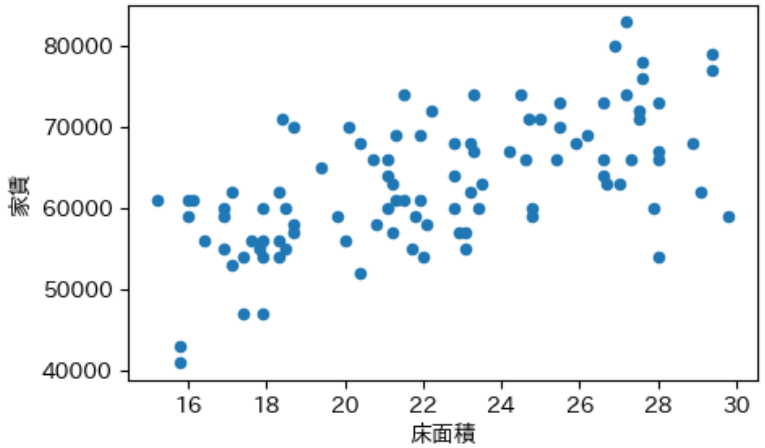
取り込んだデータには自動的に連番のindexが割り当てられます。

	床面積	築年数	階数	家賃
0	20.1	5	3	70000
1	15.8	20	5	43000
2	23.1	14	6	57000
3	17.6	15	2	56000
4	21.3	6	4	69000

2つの量の関係をグラフ化する

- xとyの2つの軸について、データの散らばり具合を描くのが散布図です。散布図を描くことによって、2つの量の関係性を図示することができます。
- 新しいコードセルを追加して、下記を記述して実行しましょう。

```
#plot.scatterメソッドで散布図が描けます。
#xはx軸の列名、yはy軸の列名、figsizeはグラフの横縦のサイズです。
#axにはグラフのオブジェクト（サブプロットと呼ばれる）が代入されます。
ax = data.plot.scatter(x="床面積", y="家賃", figsize=(5, 3))
```



機械学習のモジュール

- Pythonには様々な機械学習のモジュールが用意されています
- モジュールを用いる利点
  - それぞれのアルゴリズムを1から実装する必要がない
  - コードが簡潔に書ける
- 代表的な機械学習モジュール
  - scikit-learn
    - 回帰、ランダムフォレストなど様々なアルゴリズムを備えたモジュール
  - genism
    - 様々なトピックモデルを実装したテキスト解析モジュール
  - Tensor Flow
    - 代表的な深層学習モジュール
  - Keras
    - TensorFlowをベースにより扱いやすくした深層学習モジュール

scikit-learnを用いたパラメータの推定

- scikit-learnモジュールのlinear_modelパッケージのLinearRegressionクラスを使うと回帰直線の傾きaと切片bのパラメータ（回帰係数）を求めることができます。
  - linear model（和訳：線形モデル）、linear regression（和訳：線形回帰）
- 新しいコードセルを追加して、下記を記述して実行しましょう。

```
#LinearRegressionクラスのインポート
from sklearn.linear_model import LinearRegression

#LinearRegressionオブジェクトの生成
reg = LinearRegression()

x = data["床面積"].values.reshape(-1, 1)
y = data["家賃"]

#学習してパラメータを計算
reg.fit(x, y)
#傾きと切片の表示
print("傾き", reg.coef_[0])
print("切片", reg.intercept_)
```

【コードの解説】
下方のfitメソッドにx軸の値を与える際に2次元（行列）形式として渡す必要があるため、1次元のベクトル形式から2次元の行列形式に変換している。
.reshape（行数, 列数）として行列形式に変換する際に、引数として-1を指定すると、元のデータから行数（または列数）を自動的に決めてくれる。今回の例では、100次元ベクトルを100行1列の行列に変換している。
x = pd.DataFrame(x) でも可。

【コードの解説】
coefはcoefficient（係数）の略
interceptは日本語で切片のこと

回帰直線の描画(1)

- 求めたパラメータを使って回帰直線を描きましょう。

$$y = ax + b$$

家賃の推定値　傾き reg.coef_[0]　床面積　切片 reg.intercept_

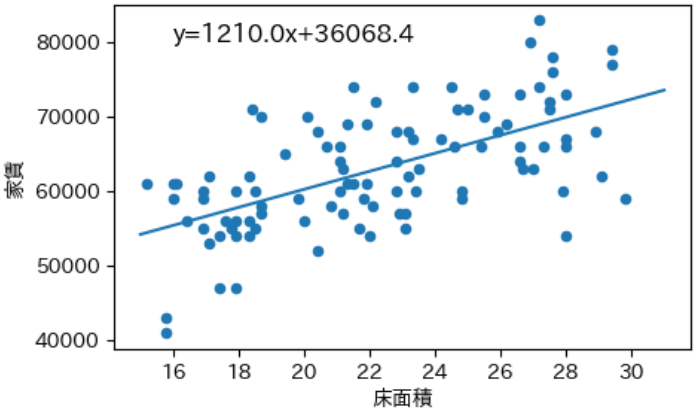
  - 新しいコードセルを追加して、下記を記述して実行しましょう。

```
#傾きをa、切片をbに代入
a = reg.coef_[0]
b = reg.intercept_

ax = data.plot.scatter(x="床面積", y="家賃", figsize=(5, 3))
#xが15から31までで直線を描く
ax.plot([15, 31], [a*15+b, a*31+b])
#回帰直線の方程式をannotateメソッドで描く
ax.annotate("y={0:.1f}x+{1:.1f}".format(a, b), xy=(16, 80000), size=12)
plt.show()
```


回帰直線の描画(2)

- 床面積から家賃を推定する線形回帰モデルが完成しました。



（補足）LinearRegressionオブジェクトのscoreメソッドで、決定係数が計算できます。

```
print(reg.score(x,y))
```

```
0.3736142422125919
```

- 一旦パラメータが求められれば、下記の式で家賃の推定値が計算できます。

床面積(m^2)

```
print(a * 25 + b)
```

```
66318.83089970793
```

(補足)文字列のフォーマット

- 変数の値をもとに文字列を作成するには、formatメソッドを使います。

```
ax.annotate("y={0:.1f}x+{1:.1f}".format(a, b), xy=(16, 80000), size=12)
```

- formatメソッド
  - 文字列に値を当てはめて整形するためのメソッド
  - 構文
    “{値のインデックス番号:.小数点以下の桁数 型}…{…}”.format(値1, 値2, …)

    ※値が1つの場合やformatメソッドの引数の順番通りの場合は、インデックス番号は省略できます。
    ※型としては、整数の場合はd、実数の場合はf、文字列の場合はs(省略可)を指定します。
  - 今回は、formatメソッドの引数の順番通りなので、下記でもOK

```
ax.annotate("y={:.1f}x+{:.1f}".format(a, b), xy=(16, 80000), size=12)
```

  - 値が1つの場合や引数の順番通りの場合、%を使用した省略表記も可能

```
ax.annotate("y= %.1f x+ %.1f" % (a, b), xy=(16, 80000), size=12)
```


付録

重回帰分析とは

- 複数の説明変数を用いた回帰を重回帰モデルと呼びます。

$$y = a_1x_1 + a_2x_2 + b$$

目的変数　説明変数1　説明変数2

  - 重回帰モデルのパラメータa1, a2, bを、偏回帰係数と呼びます。

- 仮定
  - 家賃は、床面積だけでなく、築年数や階数にも関係するのでは？
  - 例えば、説明変数１＝床面積、説明変数２＝築年数など・・・

- どの項を説明変数に加えるのか？
  - 目的変数と説明変数との相関を可視化して検討
  - Seabornモジュールを使用

Seabornによる変数間の関係性の表示

- Seabornとは？
  - matplotlibの機能をもとに、より簡単に美しいグラフ描画を実現するためのモジュールです。
  - pairplot関数を使うことで、各変数の関係性を一覧表形式で図示できます。
  - 新たなセルを追加して下記を記述します。

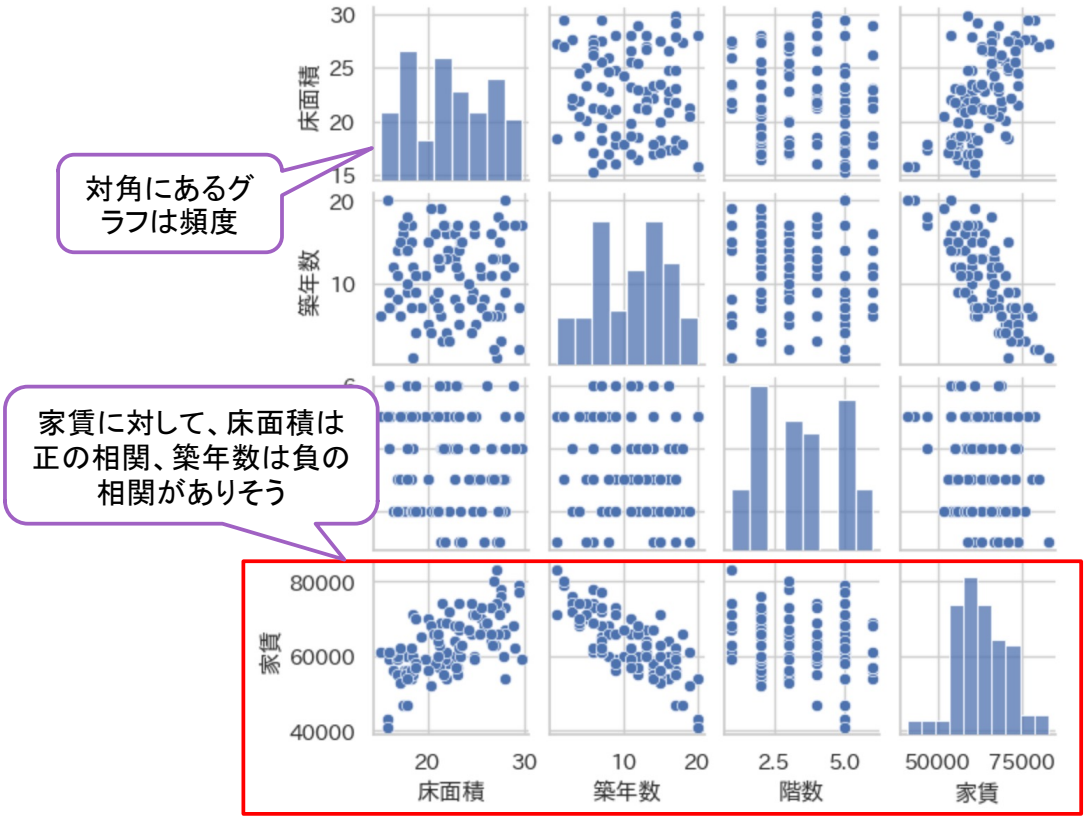
```
import seaborn as sb

sb.set(style="whitegrid", context="notebook", font="IPAexGothic")
sb.pairplot(data, size=1.5)
plt.show()
```

whitegridは背景が白で罫線有、notebookはラベルの大きさ小さめ、日本語フォントを指定

各変数間の関係を散布図として表示
sizeは各グラフのサイズ

可視化結果



別の可視化を試す

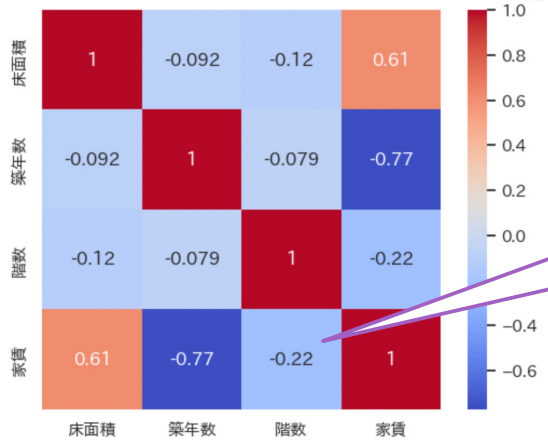
- heatmap関数を使うことで、各変数間の相関係数をヒートマップとして可視化できます。

```
import seaborn as sb

sb.set(style="whitegrid", context="notebook", font="IPAexGothic")
#sb.pairplot(data, size=1.5)
cmap = sb.color_palette("coolwarm", 200)
sb.heatmap(data.corr(), square=True, annot=True, cmap=cmap)
plt.show()
```

青から赤へのグラデーションの設定（色数200）

DataFrameのcorrメソッドで各変数間の相関係数を計算し、ヒートマップを描画



家賃に対して、階数は負の相関が見られるが、他の説明変数よりは弱い

築年数を説明変数に加える

- 新たなセルを追加して、重回帰分析のコードを記述します。

```
#LinearRegressionオブジェクトの生成
reg = LinearRegression()

x = data[["床面積", "築年数"]] #説明変数
y = data["家賃"] #目的変数

#学習してパラメータを計算
reg.fit(x, y)
#傾きと切片の表示
print("傾き1", reg.coef_[0])
print("傾き2", reg.coef_[1])
print("切片", reg.intercept_)
```

床面積についての傾き

築年数についての傾き

```
傾き1 1079.1570245020864
傾き2 -1161.7915136979154
切片 51787.16566099554
```


重回帰分析モデルによる家賃の推定

- 一旦パラメータが求められれば、下記のコードで家賃の推定値が計算できます。

```
a1 = reg.coef_[0]
a2 = reg.coef_[1]
b = reg.intercept_

yachin = a1 * 25 + a2 * 10 + b
print(yachin)
```

床面積

築年数

```
67148.17613656854
```

- 築年数を説明変数に追加した場合の決定係数を確認しましょう。

```
print(reg.score(x,y))
```

```
0.887765576276154
```

説明変数が床面積のみの場合の決定係数「0.3736」から、値が改善されていることがわかります。