第 7 章

統計分析と機械学習

本章では，まず，Excel などの表計算ツールで行うことのできる統計分析を Pandas で行う手法を紹介し，その後，機械学習の各種手法についてある程度量のあるデータを用いて実践的に学びます。Pandas の文法は複雑ではあるものの，Gemini や ChatGPT といった生成 AI ツールで容易にコードが書けるようになりましたので，正確に文法事項を覚える必然性はありません。とはいえ，「Pandas にはどのようなことができるのか」，そして最低限の文法をたとえ浅くとも体系的に知っておくことなしには，どう活用すればよいのかも分からなくなります。

7.1 Pandas による統計分析の基本（購買データ）

本節では，下図のようなあるスーパーマーケットの商品の 1 年間の販売個数のデータと，同日の最高気温のデータ（Excel ファイル）を読み込んで，Pandas を用いた統計分析の基本手法を紹介していきます（Pandas の基本事項は 3.3 節を参照）。

7.1.1 表計算シートを csv 形式に変換する

まず，Excel ファイルを csv 形式に変換して用います。csv 形式とは**カンマ区切り形式（Comma-Separated-Values）**の略称で，同じ行のデータが表形式ではなく，カンマで区切られた状態で連続して保存される形式のものです（例えば，2 行目は「2022/1/1,289,18,180,94,268」のように保存されますが，ファイルで開くと表形式で表示されます）。

	A	B	C	D	E	F
1		コーヒー	スポーツドリンク	焼き鳥	コロッケ	チョコレート
2	2022/1/1	289	18	180	94	268
3	2022/1/2	268	14	164	124	452
4	2022/1/3	213	25	182	113	293
5	2022/1/4	278	63	187	128	388
6	2022/1/5	298	41	164	162	420

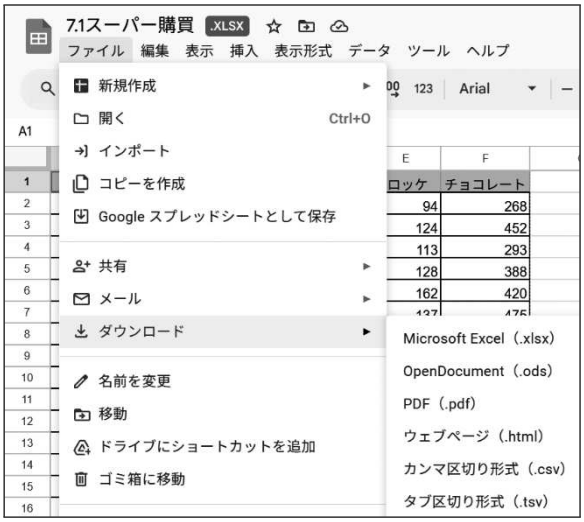
	A	B
1	日付	最高気温
2	2022/1/1	7.8
3	2022/1/2	7.9
4	2022/1/3	10.5
5	2022/1/4	12.4
6	2022/1/5	8.7

（重要な注意）セルの結合箇所や複数の表がある場合は前処理が必要

複数のセルの結合をしている箇所は，そのまま一つのセルのデータとして保存されてしまうため，あらかじめ結合する前の状態に調整しておく必要があります。また**同じシートに複数の表がある**と，これらも左詰めで保存されるため，表同士の区別がつかなくなります（今回用いる

Excel ファイル（7.1 スーパー購買.xlsx）では，セルの結合を一切してなく，かつ一枚のシートに表は一つにしてありますので調整は不要です）。Pandas のデータとして読み込んだ後でもできなくはないですが，本書のレベルを超えるので，表計算シートの段階で整理しておきます。

- Excel ファイルから csv ファイルに変換する方法
  → Excel ファイルを開いて，「名前を付けて保存」を選択，「ファイル形式」で CSV を選択すると保存できます。ただし，開いているシート 1 枚ごとに 1 枚のファイルとして保存する必要があります。
- Google スプレッドシートから csv ファイルに変換する場合
  → まず Google ドライブのマイドライブなどに今回用いる Excel ファイル「7.1 スーパー購買.xlsx」をアップロードして，Google スプレッドシートとして開きます。



［ファイル］→［ダウンロード］→［カンマ区切り形式］を選択して，購買データと最高気温のシートそれぞれを別ファイルとして保存，ローカルフォルダにダウンロードされます（本書では「7.1 購買データ.csv」「7.1 最高気温.csv」としています）。

7.1.2 csv ファイルの読み込み

それでは，作成した csv ファイルのデータを colab ファイルに読み込んで，Pandas のデータフレームに変換していきます。

まず，下のように入力して csv ファイルをアップロードします。

```
from google.colab import files
upload = files.upload()
```

ファイルは，「7.1 購買データ.csv」「7.1 最高気温.csv」の 2 つを選択します。

次に，japanize-matplotlib をインストールします。

```
pip install japanize-matplotlib
```

続けて以下の標準ライブラリをインポートします。

```
import pandas as pd
import datetime                    # 日付データの表示形式を変換する
import numpy as np
import matplotlib.pyplot as plt
import japanize_matplotlib
```

データフレームは，pd.read_csv(' ファイル名') で読み込んで作成します。(header=0 は，読みこんだ csv ファイルの 1 行目を列のラベル（日付，コーヒーなど）として設定することを表します。デフォルトで設定されているので，省略できます。)

```
df1 = pd.read_csv('7.1購買データ.csv',header = 0)
                # csvファイルの1行目を列のラベルとして設定。デフォルトで設定あり
df1.head()
```

データフレームを df1 として，df1 と入力すると一部省略されつつも，全容がわかります。通例は大きすぎるので，「(データフレーム名).head()」として 5 行目まで出力します。

	日付	コーヒー	スポーツドリンク	焼き鳥	コロッケ	チョコレート
0	2022/1/1	289	18	180	94	268
1	2022/1/2	268	14	164	124	452
2	2022/1/3	213	25	182	113	293
3	2022/1/4	278	63	187	128	388
4	2022/1/5	298	41	164	162	420

7.1.3 四分位数と箱ひげ図

まず，describe メソッドで全容を把握します。データ数，平均値（mean），標準偏差（std），四分位数が表示されます。四分位数は，データを昇順で並べ替えたとき，25 % ずつの区切り目の順位のデータを出力したものです。

四分位数の定義はさまざまあり，Excel や Pandas で出力される値は，高校教科書の「中央値の中央値」として出力されるものと少し異なります。ただしデータ数が多いときは，定義の違いによる差はなくなります。

```
df1.describe()
```

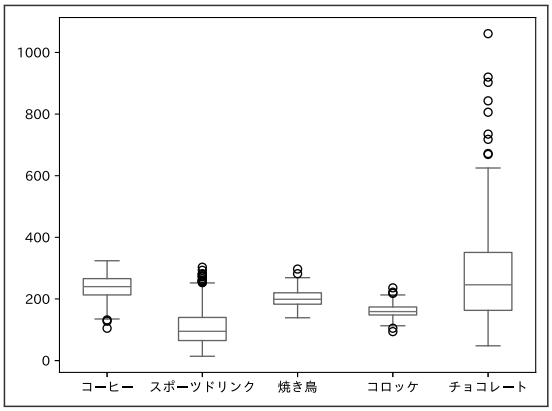
	コーヒー	スポーツドリンク	焼き鳥	コロッケ	チョコレート
count	365.000000	365.000000	365.000000	365.000000	365.000000
mean	237.504110	113.350685	202.175342	161.627397	275.158904
std	39.456028	65.167390	27.581088	19.837480	153.205518
min	105.000000	14.000000	139.000000	94.000000	48.000000
25%	213.000000	65.000000	183.000000	148.000000	163.000000
50%	240.000000	95.000000	199.000000	159.000000	246.000000
75%	266.000000	140.000000	220.000000	174.000000	351.000000
max	324.000000	303.000000	297.000000	236.000000	1061.000000

箱ひげ図は，「`df.plot(kind='box')`」で表示させます。
いまは `df1` のうち，日付以外の列だけ抽出するため，`df1.columns` でラベル名を出力して，

```
df1[['コーヒー','スポーツドリンク',……,'チョコレート']]
```

の代わりに，「`df1[df1.columns[1:6]]`」と入力が楽になる工夫を行っています。

```
df1[df1.columns[1:6]].plot(kind='box')  # 外れ値は四分位範囲の1.5倍
plt.show()
```



ここで白丸は**外れ値**，つまり第 1 または第 3 四分位数から，**四分位範囲**（第 1 と第 3 四分位の差）の 1.5 倍以上離れたデータを表します。この箱ひげ図から，データのばらつき具合が視覚的にわかります。

7.1.4 標準偏差と相関係数の定義からの算出

ここでは，Pandas の基本操作を確認する目的で，標準偏差や相関係数の値を定義から算出することを行います。

まず，最高気温のデータをデータフレーム df2 として格納します。

```
df2 = pd.read_csv('7.1最高気温.csv',header = 0)
df2.head()
```

	日付	最高気温
0	2022/1/1	7.8
1	2022/1/2	7.9
2	2022/1/3	10.5
3	2022/1/4	12.4
4	2022/1/5	8.7

以下，df1 のコーヒーと最高気温のデータの標準偏差と相関係数を求めます。

標準偏差は，**分散**と呼ばれる統計量の正の平方根として定義されます。分散は，データの散らばり具合の指標として，次のように定義されます。

分散の定義

変量 X の n 個のデータ $(x_1, x_2, \ldots, x_n)$ について，平均値を $\overline{x}$ で表すとき，X の分散は，

$$\frac{1}{n}\sum_{k=1}^{n}(x_k-\overline{x})^2=\frac{1}{n}\{(x_1-\overline{x})^2+\cdots+(x_n-\overline{x})^2\}$$

つまり，各データの平均値との差である偏差を 2 乗した偏差平方の平均値として求められます。

df3 に df1 のコーヒーのデータを「df1['コーヒー']」で抽出して格納します。

```
df3 = pd.DataFrame(df1['コーヒー'])
df3.head()
```

	コーヒー
0	289
1	268
2	213
3	278
4	298

問題 7.1: 標準偏差の算出と Pandas の基本操作

次の手順で，df3 の標準偏差を求めなさい。

- Numpy の mean メソッドで平均値を求める。
- ブロードキャスト機能を利用して，「偏差」「偏差平方」を算出して，それらを順に df3 の列データとして追加する。
- 標準偏差を求める。

問題 7.1 の解答

```
df3 = pd.DataFrame(df1['コーヒー'])
df3ave = np.mean(df3['コーヒー'])        # 平均値を求める
print('コーヒーの平均値:'+str(df3ave))
df3['偏差'] = df3['コーヒー']-df3ave     # 各データと平均値の差を求める
```

```
df3['偏差平方'] = df3['偏差']**2          # 各偏差を2乗(ブロードキャスト機能)
df3sd = (np.mean(df3['偏差平方']))**0.5
# (偏差平方の平均=)分散を求め，平方根を求める(=標準偏差)
print('コーヒーの標準偏差',str(df3sd))
df3.head()
```

```
コーヒーの平均値:237.5041095890411
コーヒーの標準偏差 39.401941536044994
```

	コーヒー	偏差	偏差平方
0	289	51.49589	2651.826729
1	268	30.49589	929.999332
2	213	-24.50411	600.451387
3	278	40.49589	1639.917140
4	298	60.49589	3659.752757

続いて，コーヒーのデータと最高気温の相関係数を求めます。ここで，相関係数の定義，その前段階として共分散の定義を確認します。

共分散の定義

変量 X の n 個のデータ $(x_1, x_2, \dots, x_n)$ と変量 Y の n 個のデータ $(y_1, y_2, \dots, y_n)$ について，各平均値を $\overline{x}$，$\overline{y}$ で表すとき，X と Y の共分散は，

$$\frac{1}{n}\sum_{k=1}^{n}(x_k-\overline{x})(y_k-\overline{y}) = \frac{1}{n}\{(x_1-\overline{x})(y_1-\overline{y})+\cdots+(x_n-\overline{x})(y_n-\overline{y})\}$$

（つまり，偏差の積の平均値）

相関係数の定義

2 変量の X，Y の相関係数は，

$$\frac{(X,\ Y\text{の共分散})}{(X\text{の標準偏差})(Y\text{の標準偏差})}$$

df4 に df1 からコーヒー，df2 から最高気温のデータを抽出し，格納します。ここでのちほど散布図に描画する上で，x 軸を最高気温，y 軸をコーヒーの販売数にするため，pandas の rename メソッドで，ラベルを「コーヒー」から「y コーヒー」にします。

```
df4 = pd.DataFrame(df1[df1.columns[1]])
df4['x最高気温'] = df2[df2.columns[1]]
df4 = df4.rename(columns={'コーヒー':'yコーヒー'})  # renameメソッドを利用
df4.head()
```

問題 7.2: 共分散と相関係数の算出

問題 7.1 を参考に，df4 の共分散と相関係数を順に算出しなさい。

問題 7.2 の解答例

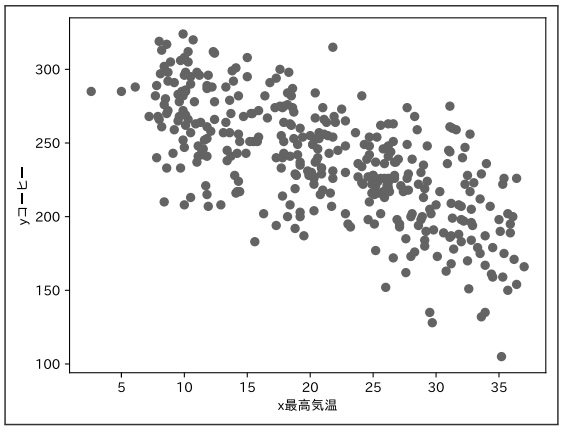
```
df4yave = df4['yコーヒー'].mean()            # df.mean()でも平均値は出せます
df4['y偏差'] = df4['yコーヒー']-df4yave
df4xave = df4['x最高気温'].mean()
df4['x偏差'] = df4['x最高気温']-df4xave
df4xsd = ((df4['x偏差']**2).mean())**0.5
print('最高気温の標準偏差',str(df4xsd))
df4['偏差の積'] = df4['x偏差']*df4['y偏差']
df4cov = df4['偏差の積'].mean()              # 共分散
print('共分散:',str(df4cov))
df4cor = df4cov/(df3sd*df4xsd)               # 相関係数
print('相関係数',str(df4cor))
df4.head()
```

```
最高気温の標準偏差 8.363843181912664
共分散: -226.74756614749484
相関係数 -0.6880486236940089
```

	yコーヒー	x最高気温	y偏差	x偏差	偏差の積
0	289	7.8	51.49589	-13.341096	-687.011612
1	268	7.9	30.49589	-13.241096	-403.799009
2	213	10.5	-24.50411	-10.641096	260.750580
3	278	12.4	40.49589	-8.741096	-353.978461
4	298	8.7	60.49589	-12.441096	-752.635174

散布図は次のようになります。負の強い相関関係とみてよいことがわかります。

```
plt.scatter(df4['x最高気温'],df4['yコーヒー'])
plt.xlabel('x最高気温') ; plt.ylabel('yコーヒー')
plt.show()
```



7.1.5 Numpy を用いた統計量の算出（分散共分散行列と相関行列）

ここでは，Numpy を用いた分散，共分散，相関係数の算出方法を紹介します。

まず，分散と共分散は，次のように np.cov により**分散共分散行列**（**covariance matrix**）として算出されます。

```
np.cov(df4['yコーヒー'],df4['x最高気温'],ddof = 0)
                                        # ddof=0は通常の分散，=1は不偏分散
```

```
array([[1552.51299681, -226.74756615],    # yの分散，y，xの共分散
       [-226.74756615,   69.95387277]])   # x，yの共分散，xの分散
```

（ここで**不偏分散**は，偏差平方をデータ数 n ではなく $n-1$ で割って算出する量であり，母集団全体が分からず標本のデータしかないときに，母集団の分散を推定するときに用います。ただ n の値が大きい（30 が基準）ほど，普通の分散との違いは少なくなります。なお，ddof=1 の不偏分散がデフォルトです。）

相関係数も同様に，**相関係数行列**（**correlation matrix**）として算出されます。

```
np.corrcoef(df4['yコーヒー'],df4['x最高気温'])  # 相関係数行列として相関係数を表示
```

```
array([[ 1.        , -0.68804862],      # y自身との相関係数，yとxの相関係数
       [-0.68804862,  1.        ]])     # x，yの相関係数，x自身との相関係数
```

7.1.6 回帰直線の算出と描画

相関がある程度強い 2 変量について，変量 X のデータから変量 Y のデータを予測するときに用いる 1 次式を**回帰直線**といい，その係数について次の公式が知られています（詳細は例えば「中高生からのデータサイエンス」参照）。

回帰直線の係数

$$（傾き）= \frac{（X，Yの共分散）}{（Xの分散）}，\quad （\text{y 切片}）= \overline{y} -（傾き）\times \overline{x}$$

問題 7.3: 回帰直線の算出と描画

df4 のデータについて，回帰直線の係数を求め，散布図に回帰直線を追加しなさい。

問題 7.3 の解答例

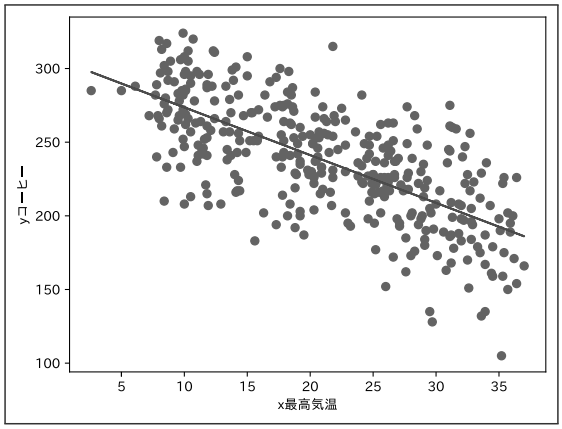
```
df4slope = df4cov/(df4xsd**2)              # 傾き
df4intercept = df4yave-df4slope*df4xave    # y切片
print('傾き',str(df4slope))
print('切片',str(df4intercept))
```

```
傾き -3.24138689058346
切片 306.03058066078705
```

```
df4['y予測'] = df4slope * df4['x最高気温'] + df4intercept # ブロードキャスト機能利用
df4.head()
```

	yコーヒー	x最高気温	y偏差	x偏差	偏差の積	y予測
0	289	7.8	51.49589	-13.341096	-687.011612	280.747763
1	268	7.9	30.49589	-13.241096	-403.799009	280.423624
2	213	10.5	-24.50411	-10.641096	260.750580	271.996018
3	278	12.4	40.49589	-8.741096	-353.978461	265.837383
4	298	8.7	60.49589	-12.441096	-752.635174	277.830515

```
plt.scatter(df4['x最高気温'],df4['yコーヒー'])
plt.plot(df4['x最高気温'],df4['y予測'],color='red')
plt.xlabel('x最高気温') ; plt.ylabel('yコーヒー')
plt.show()
```



なお，回帰直線を求めるツールは，本章 7.4 節にて機械学習の一手法として用意されていますので，ここでは割愛します。

7.1.7 figure 環境を用いた複数項目間の散布図の描画

ここでは，他の商品と最高気温の相関関係について，散布図を描いて調べることにします。3.2 節で導入した `figure` 環境に，各散布図をサブプロットとして追加していきます。

まず，`df5` に購買データと最高気温のデータをまとめます。

```
df5 = pd.DataFrame(df1[df1.columns[1:]])
df5['最高気温'] = df2.iloc[:,1]
df5.head()
```

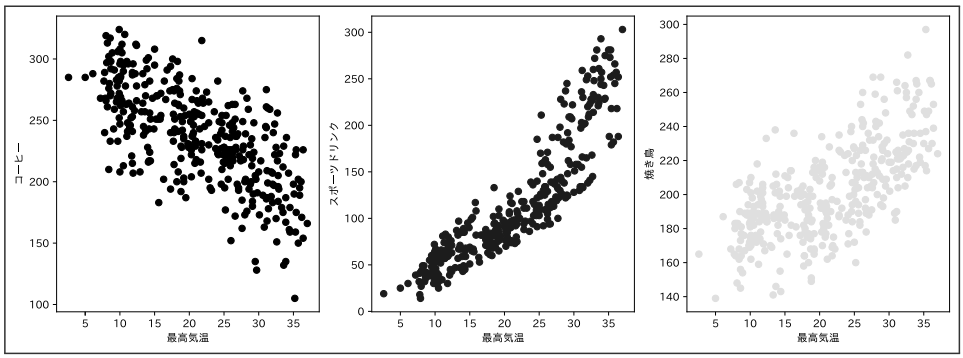
	コーヒー	スポーツドリンク	焼き鳥	コロッケ	チョコレート	最高気温
0	289	18	180	94	268	7.8
1	268	14	164	124	452	7.9
2	213	25	182	113	293	10.5

まず，3 品目について描画します。

```
fig = plt.figure(figsize=(15,5))    # figure環境を用意
ax1 = fig.add_subplot(1,3,1)    # まず散布図を3つ用意することを想定して1×3で並べる
x1 = df5['最高気温'].values      # numpy配列として抽出
y1 = df5['コーヒー'].values
ax1.scatter(x1,y1,c='black')
ax1.set_xlabel('最高気温') ; ax1.set_ylabel('コーヒー')

ax2 = fig.add_subplot(1,3,2)
ax2.scatter(df5['最高気温'],df5['スポーツドリンク'],c='blue')
                                                    # 直接入力値として指定も可
ax2.set_xlabel('最高気温') ; ax2.set_ylabel('スポーツドリンク')

ax3 = fig.add_subplot(1,3,3)
x3 = df5['最高気温'].values ; y3 = df5['焼き鳥'].values
ax3.scatter(x3,y3,c='khaki')
ax3.set_xlabel('最高気温') ; ax3.set_ylabel('焼き鳥')
plt.show()
```



for 文と enumerate 関数を用いることで以下のようにまとめることもできます。