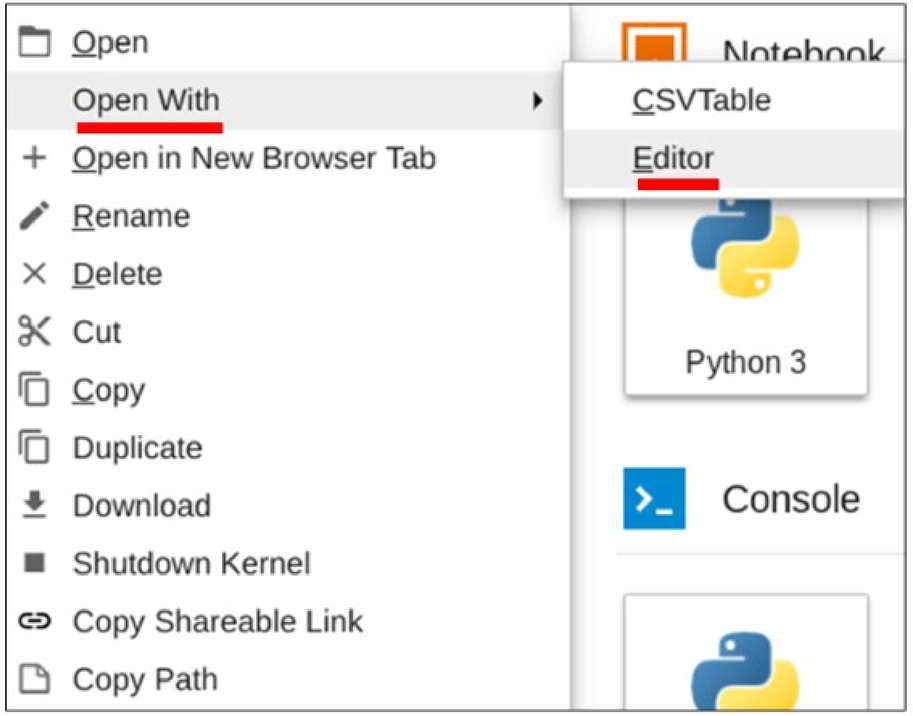
相関のプログラミング

CSVファイルの確認

- JupyterLab のファイル一覧から、cs3-07フォルダの下の batting_npb2021.csv を右クリック、Open With > Editor で開く。



CSVファイルのデータの概要

- batting_npb2021.csv は、日本のプロ野球の2021年シーズンにおける打撃成績上位61名の身長 (cm)、体重 (kg)、ホームラン数、三振数、盗塁数をまとめたものです。
  - 日本野球機構ホームページ(https://npb.jp/)より
  - 三振: 打者がボールを打てずに3つ目のストライクを取られてアウトになるプレー
  - 盗塁: 投手が投げている間に、走者がすばやく走って1つ先の塁に進むプレー

```
1  Height,Weight,HomeRun,StrikeOut,Steal
2  173,85,21,26,0
3  170,85,11,65,5
4  190,104,32,116,3
5  188,87,28,122,6
```

ノートブックの作成

- cs3-07フォルダの下に新規ノートブックを作成し、ファイル名を correlation.ipynb に変更します。
- 次ページからの内容にしたがって、各セルを作成、Shift+Enter で実行してください。そして、解説の内容をよく読み、各セルのプログラムと結果それぞれの意味について理解してください。
- まず、ipynb ファイルの先頭に、このノートブックの簡単な説明を入れておきましょう。

(Markdown)

```
### Pearson correlation coefficient
```

(Markdown)
```
#### Import libraries
```

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

必要なライブラリをインポートし、
CSVファイルを読み込みます。

(Markdown)
```
#### Read CSV file
```

```
csv_in = 'batting_npb2021.csv'
df = pd.read_csv(csv_in, sep=',', skiprows=0, header=0)
print(df.shape)
print(df.info())
display(df.head())
```

```
(61, 5)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 61 entries, 0 to 60
Data columns (total 5 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Height     61 non-null     int64
 1   Weight     61 non-null     int64
 2   HomeRun    61 non-null     int64
 3   StrikeOut  61 non-null     int64
 4   Steal      61 non-null     int64
dtypes: int64(5)
memory usage: 2.5 KB
None
```

	Height	Weight	HomeRun	StrikeOut	Steal
0	173	85	21	26	0
1	170	85	11	65	5
2	190	104	32	116	3
3	188	87	28	122	6
4	171	85	11	84	4

ピアソン相関係数の計算

- 列同士の相関係数を求めてみましょう。

(Markdown)

```
#### Calc. of Pearson correlation coefficients
```

```
# all vs all columns
df_corr = df.corr()
display(df_corr)
```

データフレーム.corr() で、全列総当たりの相関係数を計算できます。
(結果の戻り値(df_corr)はデータフレーム)

自分自身との相関係数は 1.0 (完全な正相関) になっています。

	Height	Weight	HomeRun	StrikeOut	Steal
Height	1.000000	0.700030	0.520405	0.425124	-0.235158
Weight	0.700030	1.000000	0.561627	0.261680	-0.484311
HomeRun	0.520405	0.561627	1.000000	0.443375	-0.310043
StrikeOut	0.425124	0.261680	0.443375	1.000000	0.084401
Steal	-0.235158	-0.484311	-0.310043	0.084401	1.000000

```
# Get one correlation coefficient
print(df_corr.at['Weight', 'Height'])
```

df_corrはデータフレームなので、atで各値を取り出せます。

0.700301248513807

```
# one vs all columns
print(df.corrwith(df['Weight']))
```

```
Height        0.700030
Weight        1.000000
HomeRun       0.561627
StrikeOut     0.261680
Steal        -0.484311
dtype: float64
```

データフレーム.corrwith(シリーズ) で、
全列対指定したシリーズ(列) の相関係数を
計算できます (結果の戻り値はシリーズ)

	Height	Weight	HomeRun	StrikeOut	Steal
Height	1.000000	0.700030	0.520405	0.425124	-0.235158
Weight	0.700030	1.000000	0.561627	0.261680	-0.484311
HomeRun	0.520405	0.561627	1.000000	0.443375	-0.310043
StrikeOut	0.425124	0.261680	0.443375	1.000000	0.084401
Steal	-0.235158	-0.484311	-0.310043	0.084401	1.000000

```
# one vs one column
print(df['Weight'].corr(df['Height']))
```

```
0.7000301248513806
```

シリーズ.corr(シリーズ) で、指定したシリーズ(列)同士の相関係数を計算できます。

	Height	Weight	HomeRun	StrikeOut	Steal
Height	1.000000	0.700030	0.520405	0.425124	-0.235158
Weight	0.700030	1.000000	0.561627	0.261680	-0.484311
HomeRun	0.520405	0.561627	1.000000	0.443375	-0.310043
StrikeOut	0.425124	0.261680	0.443375	1.000000	0.084401
Steal	-0.235158	-0.484311	-0.310043	0.084401	1.000000

ヒートマップの描画

- 全列総当たりの相関係数の可視化方法としてヒートマップがあります。
- 2次元の表形式の数値の大きさを色で表現します。

(Markdown)

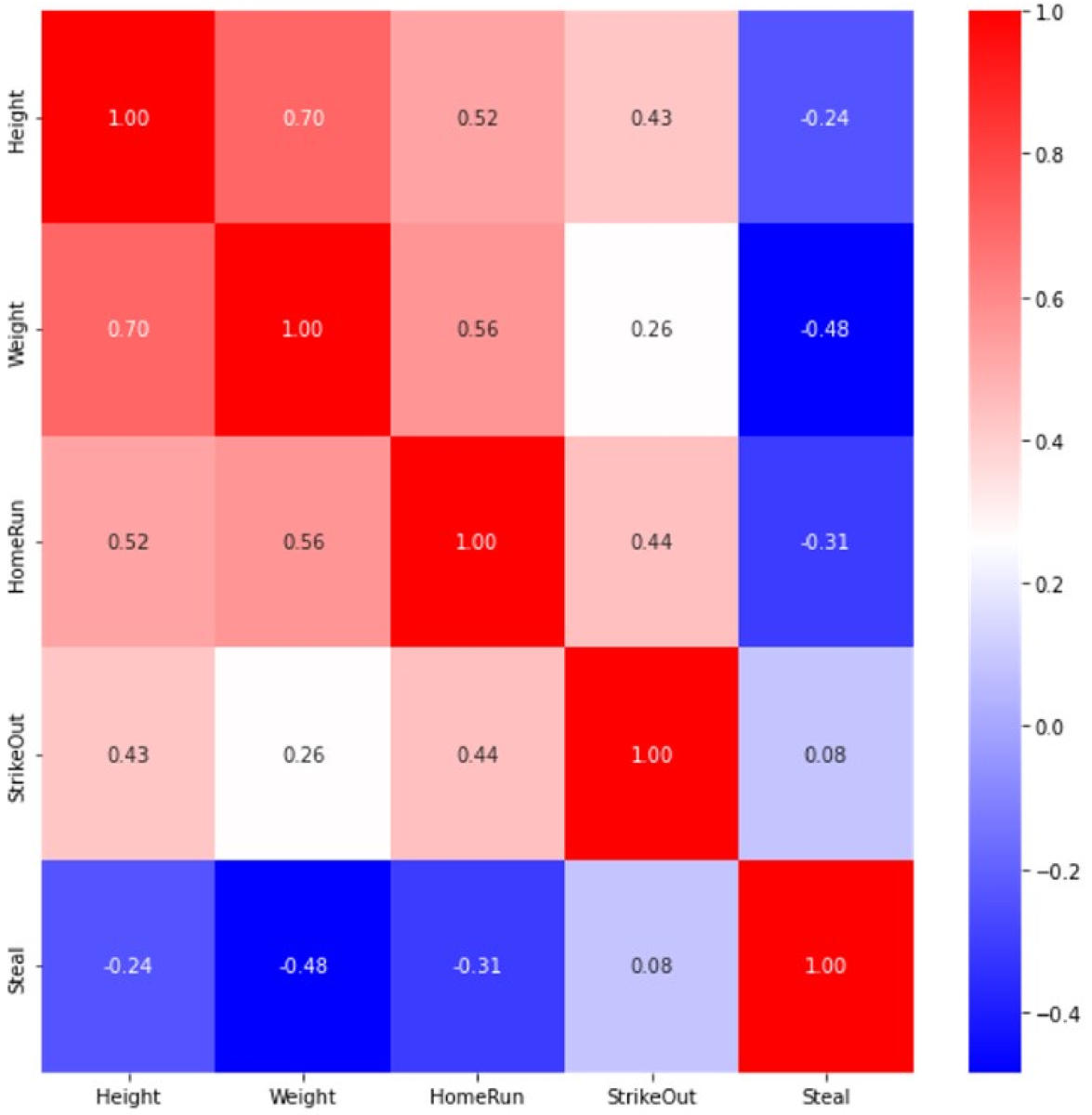
```
#### Heatmap
```

```
plt.figure(figsize=(10,10))
sns.heatmap(df_corr,annot=True,fmt='.2f',cmap='bwr')
plt.show()
```

ここでは、ヒートマップの描画に、seabornライブラリのheatmap()を用いています。

sns.heatmap(**2次元データ**, annot=True or False,
fmt=表示フォーマット, cmap=カラーマップ)

- annot = True なら各セルに数値を表示 (Falseなら非表示)。表示させるフォーマットは fmt オプションで指定。
- フォーマット指定の例:
  - '.2f'：小数点以下2桁まで(.2)の小数(f)で表示
  - '5d'：5桁の整数(d)で表示
  - …
- cmapのカラーマップ指定
  - https://matplotlib.org/stable/gallery/color/colormap_reference.html



正相関

負相関

いくつかの列の組み合わせについて、散布図を描画してみましょう。

(Markdown)

```
#### Scatter plots
```

```
plt.scatter(df['Weight'], df['Height'])
plt.title('corr: {:.2f}'.format(df_corr.at['Weight', 'Height']))
plt.xlabel('Weight [kg]')
plt.ylabel('Height [cm]')
plt.show()
```

x: Weight列、y: Height列で散布図描画。
グラフタイトルを「corr: 相関係数の値 (小数第2位まで)」としている。
(相関係数の値は df_corr から at を使って取得)

'{}を含む文字列'.format(変数または値)
文字列の中の {} 部分に変数の値を埋め込んだ文字列が生成される。

- {} の中では「:」のあとに書式指定が可能。
  - d で整数指定。dの前に整数を置くと桁数指定。桁数の先頭に0をつけると、空いた桁を0で埋める。
  - f で小数指定。fの前に x.y を置くと桁数指定(全体x桁, 小数点以下y桁。xは省略可能。x.y をつけなければ .6 と同等)。

https://docs.python.org/ja/3/library/string.html#formatspec

例:

```
a = 3; b = 1.666
print( 'a:{}, b:{}'.format(a, b) ) → a:3, b:1.666
print( '{:d}, {:3d}, {:03d}'.format(a, a, a) ) → 3,   3, 003
print( '{:f}, {:5.2f}, {:.1f}, {:.0f}'.format(b, b, b, b) ) → 1.666000,  1.67, 1.7, 2
```

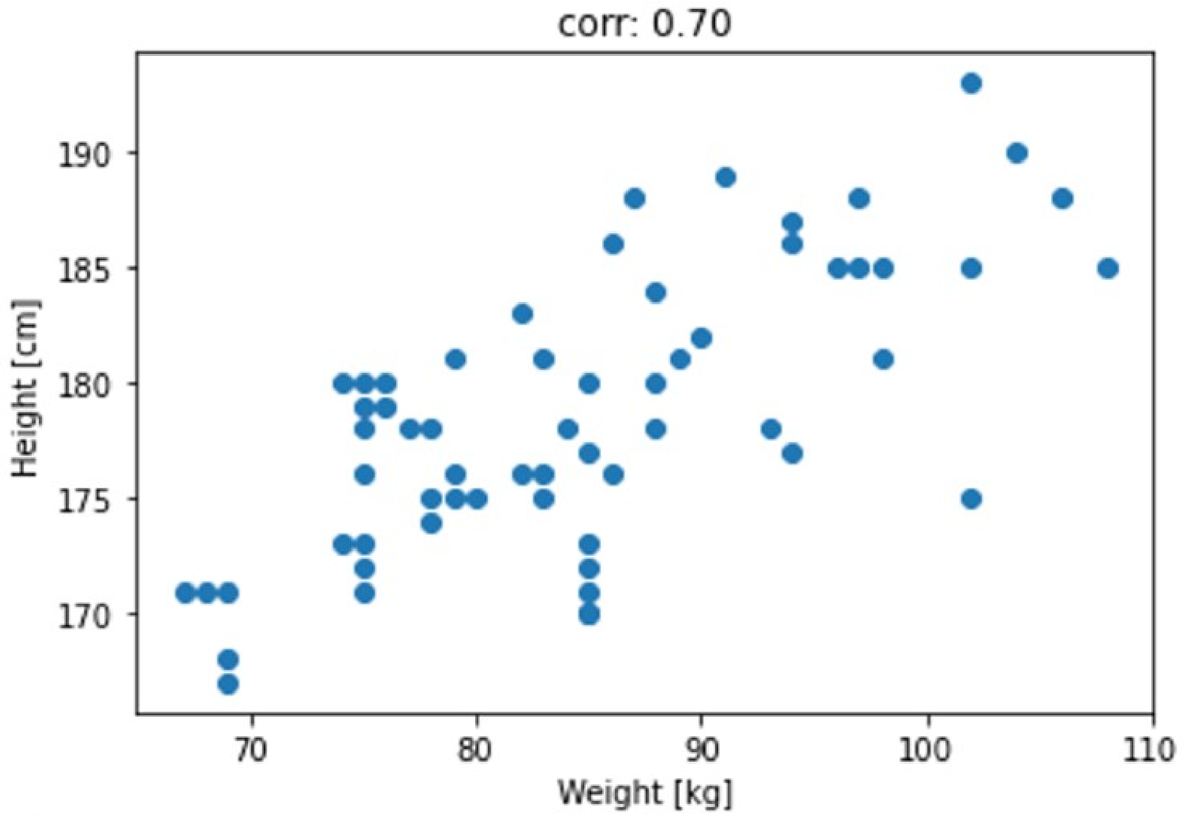
3桁

小数点の前3桁
小数点以下2桁

format.html 参照

'corr: {:.2f}'.format(df_corr.at['Weight', 'Height']))

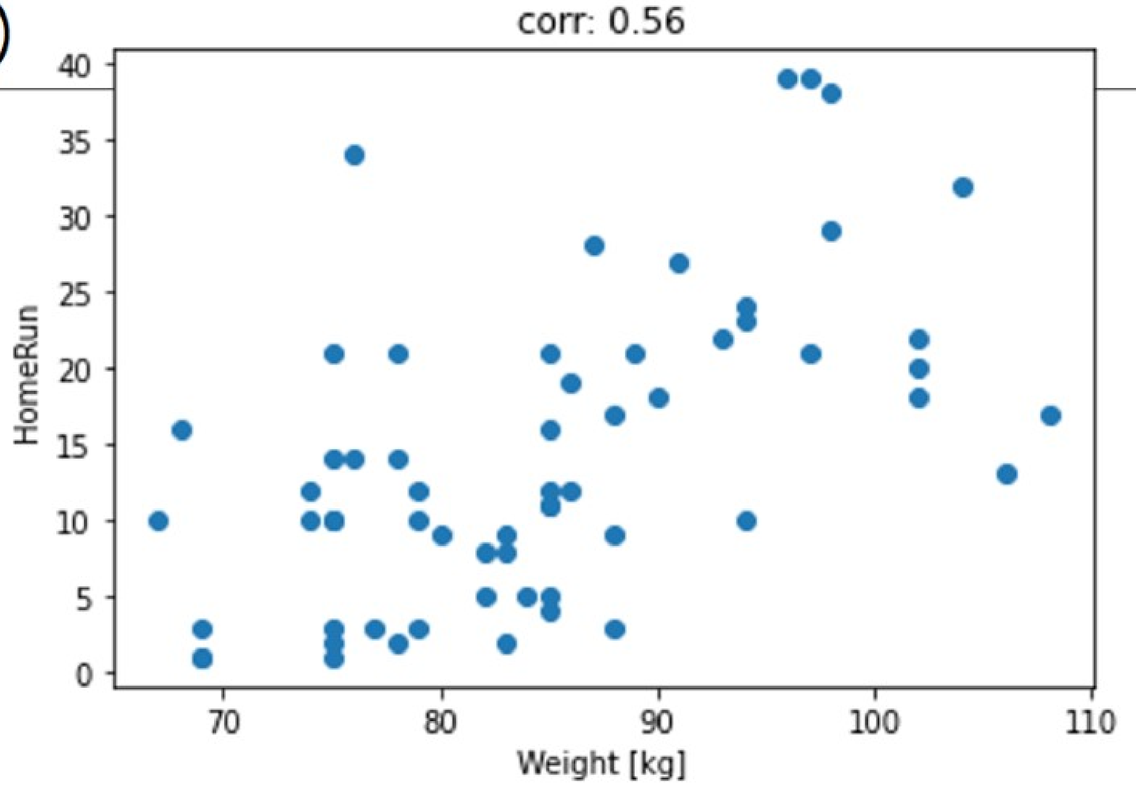
df_corr.at['Weight', 'Height'] が 0.7000301248513806
なので、'corr: {:.2f}'.format(…) は 0.70 となる。



体重と身長はかなり強い
正相関。つまり、体重が
重い選手は身長が大きい。

```
plt.scatter(df['Weight'], df['HomeRun'])
plt.title('corr: {:.2f}'.format(df_corr.at['Weight', 'HomeRun']))
plt.xlabel('Weight [kg]')
plt.ylabel('HomeRun')
plt.show()
```

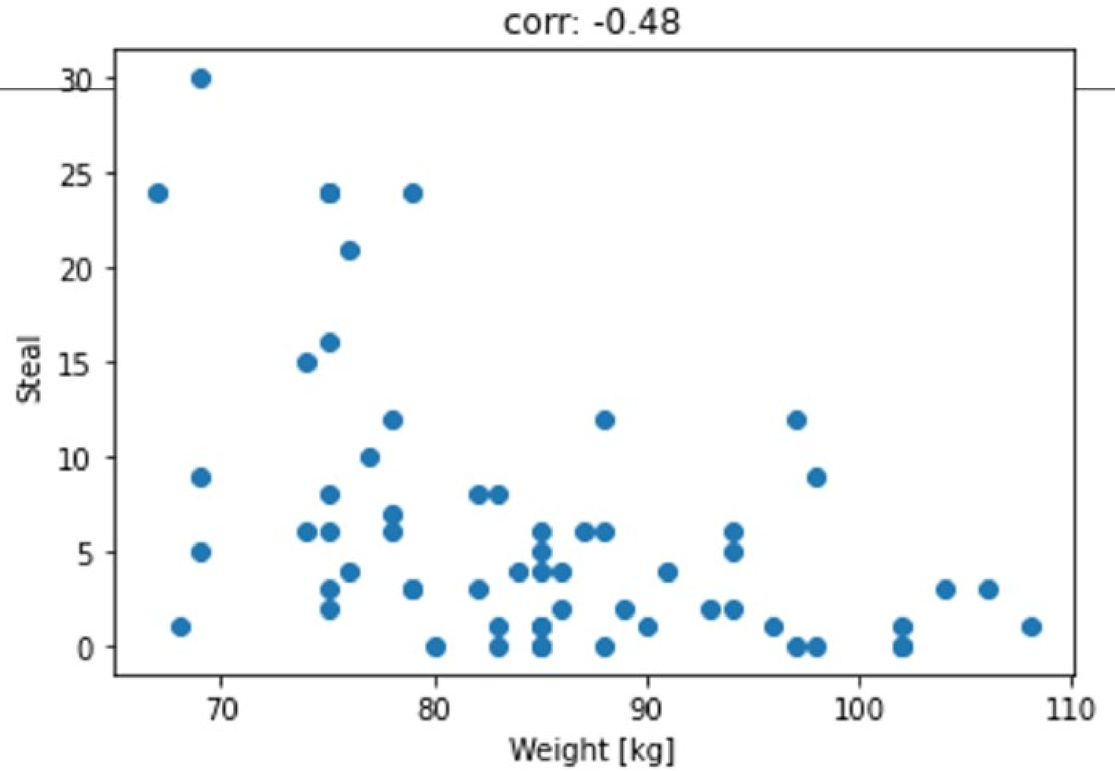
x: Weight列, y: HomeRun列 で散布図描画。



体重が重いほどホームラン数が多い、という正相関がみられる。
(体重が重い選手は多くの場合、力が強いためだと考えられる)

```
plt.scatter(df['Weight'], df['Steal'])
plt.title('corr: {:.2f}'.format(df_corr.at['Weight', 'Steal']))
plt.xlabel('Weight [kg]')
plt.ylabel('Steal')
plt.show()
```

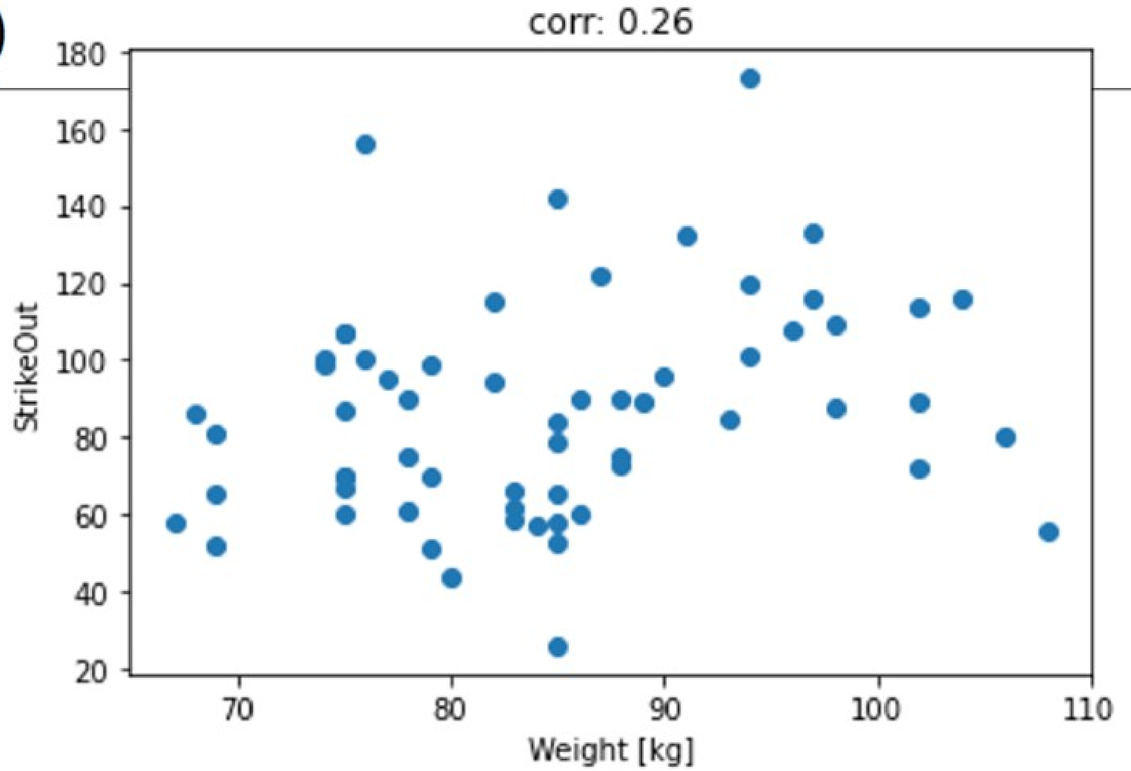
x: Weight列, y: Steal列 で散布図描画。



体重が重いほど盗塁数が少ない、という負相関がみられる。
(体重が重い選手は多くの場合、足が遅いためだと考えられる)

```
plt.scatter(df['Weight'], df['StrikeOut'])
plt.title('corr: {:.2f}'.format(df_corr.at['Weight', 'StrikeOut']))
plt.xlabel('Weight [kg]')
plt.ylabel('StrikeOut')
plt.show()
```

x: Weight列, y: StrikeOut列 で散布図描画。



体重と三振数の間の関係性は弱い(無相関に近い)。