データサイエンス特論
Data Science

データの視覚化

1. グラフ作成パッケージ
2. matplotlib
3. pandas, Titanicデータのプロット
4. Seaborn, Irisデータのプロット

(C) 創造技術コース　橋本洋志／大久保友幸
{hashimoto, ohkubo-tomoyuki}@aiit.ac.jp
http://hhlab.org/

グラフ作成パッケージ

1. パッケージ各種

パッケージ各種

matplotlib

- 代表的なグラフライブラリ
- https://matplotlib.org/
- Gallery: https://matplotlib.org/gallery/index.html

pandas

- 統計分析関数に加えてグラフ化もセットで提供しているライブラリ
- visualization: http://pandas.pydata.org/pandas-docs/stable/visualization.html

seaborn

- 統計用データのグラフ作成によく用いられる
- https://seaborn.pydata.org/

mlxtend

- 機械学習やパターン認識結果のプロットに有用
- https://github.com/rasbt/mlxtend

matplotlib

matplotlib

よく参照するドキュメントを示す。

matplotlib API

- Application Programming Interfaceの説明
- https://matplotlib.org/api/index.html
- この中の項目で，よく参照するのが
- cm (color map): 使えるカラーマップ，colors: 使えるカラー，markers:使えるマーカー（記号）

matplotlib.pyplot

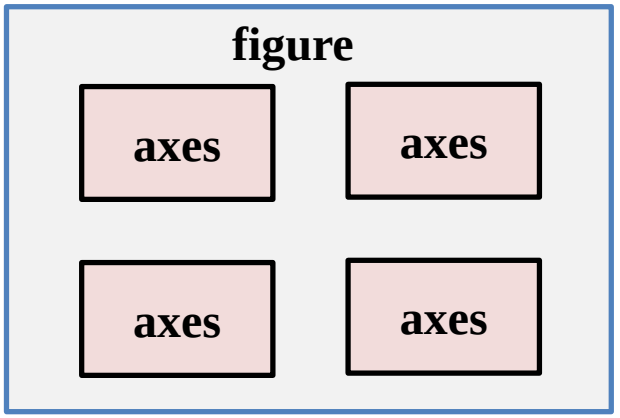
- 多数のグラフ化ツールを要している
- https://matplotlib.org/api/_as_gen/matplotlib.pyplot.html
- この中の関数から
  - plot　　折れ線，　　scatter　　散布図，　　hist　　ヒストグラム

本講義では

- グラフの装飾に複数の流儀があるため，惑うことが多く，これを避けるため，
- 一つのグラフ，複数のグラフを描く場合でも，**plt.subplots()**　の使用で統一する。
- fig = plt.subplots(figsize=(10,3))
- fig, axs = plt.subplots(nrows=2, ncols=2, figsize=(6,4))

matplotlib 複数のグラフ

- **figure, axes**
  - 複数グラフを描く場合，図のように，figureは図全体，axesはその内部にある座標軸系を意味する。
  - figure 1つに，axes 1つ，axes 2つでも良く，もっと複雑な配置も可能である(各自に委ねる)。
- **例：1つのグラフに複数波形**
  - プログラムを用いて解説
- **例：axesが2列2行**



毎回、Notebookは実行して、その結果を確認してください。

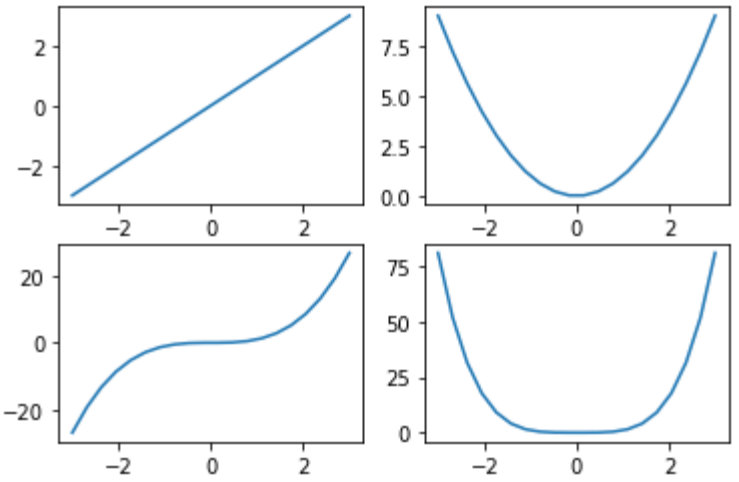
PLT_MultiplePlot

拡張子の省略は，".ipynb"

matplotlib 複数のグラフ

PLT_MultiplePlot

In [2]:
```
x = np.linspace(-3, 3, 20)
y1 = x
y2 = x ** 2
y3 = x ** 3
y4 = x ** 4
```



例：axesが2列2行

- 下記は，二つとも同じ動作を示す

```
fig, axs = plt.subplots(nrows=2, ncols=2,
figsize=(6, 4))
# 左上
axs[0, 0].plot(x, y1)
# 右上
axs[0, 1].plot(x, y2)
# 左下
axs[1, 0].plot(x, y3)
# 右下
axs[1, 1].plot(x, y4)
plt.show()
```

```
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(
nrows=2, ncols=2, figsize=(6,4))
# 左上
ax1.plot(x, y1)
# 右上
ax2.plot(x, y2)
# 左下
ax3.plot(x, y3)
# 右下
ax4.plot(x, y4)
plt.show()
```

注意：
subplots()を使わないとき：　plt.xlabel(), ylabel(), title()
subplots()を使うとき：　　plt.set_xlabel(), set_ylabel(), set_title()

pandas

Titanic

Titanicの概要

- 英国の客船で，処女航海の1912 年に北大西洋上で氷山に接触し沈没し，犠牲者が多数出た。映画で何度か上映されたため，世界的に非常に有名となった。
- 映画などはYouTubeで、そのダイジェストを見ることができる。

Titanicのデータ

- 実際の乗客データが保存されており（欠損あり）、統計分析の例題として良く用いられている。このデータを説明するサイトは多くあり，例えば次もその一つである。

  R Documentation: https://www.rdocumentation.org/packages/datasets/
- 沈没したRMS Titanic (RMS: Royal Mail Ship またはSteamer，郵便船としての機能があった) そのものの調査も行われており，この調査結果は次にある。
- Encyclopedia Titanica : Titanic Facts, History and Biography, https://www.encyclopedia-titanica.org/

Titanic を実際に映したドキュメンタリー

- 映画「タイタニック」（1997）の映画監督James.F. Cameron も実際に科学的調査を実施し，そのレポートを次に示している。

  TITANIC: 20 YEARS LATER WITH JAMES CAMERON

  http://www.natgeotv.com/int/titanic-20-years-later-with-james-cameron

pandas Titanicデータのプロット

データの取得

- titanic_url = ″http://s3.amazonaws.com/assets.datacamp.com/course/Kaggle/train.csv″ #トレニーングデータ読込み
- df = pd.read_csv(titanic_url) # df; DataFrame　の略

PLT_Titanic

毎回、Notebookは実行して、その結果を確認してください。

pandas Titanicデータのプロット

データの説明

- ラベルの説明
  - PassengerId　乗客ID
  - Survived　生存結果(1:生存, 0:死亡)
  - Pclass　客室クラス 1が最上位, 3が最下位
  - Name　氏名
  - Sex　性別
  - Age　年齢
  - SibSp　兄弟, 配偶者の人数
  - Parch　両親, 子供の人数
  - Ticket　チケット番号
  - Fare　チケット料金
  - Cabin　部屋番号
  - Embarked　乗船した港　C:Cherbourg, Q:Queenstown, S:Southampton

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

pandas Titanicデータのプロット

PLT_Titanic

欠損値の補完

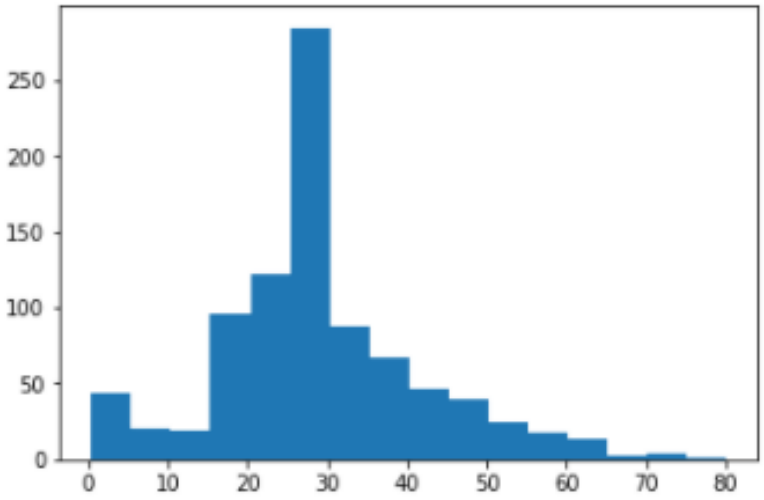
- 実際の記録をもとにしたため，欠損データある。欠損の補完の考え方は幾つかあり，ここでは，中央値を与えることとした。

```
1 df['Age'].fillna(df.Age.median(), inplace=True) #inplace=True は，処理軽減のため，元データを処理する
```

Ageのヒストグラム

```
1 plt.hist(df['Age'], bins=16)
```

```
(array([ 44.,  20.,  19.,  96., 122., 285.,  88.,  67.,  47.,
         39.,  24.,  18.,  14.,   3.,   4.,   1.]),
 array([ 0.42   ,  5.39375, 10.3675 , 15.34125, 20.315  , 25.28875,
        30.2625 , 35.23625, 40.21   , 45.18375, 50.1575 , 55.13125,
        60.105  , 65.07875, 70.0525 , 75.02625, 80.     ]),
 <a list of 16 Patch objects>)
```



pandas Titanicデータのプロット

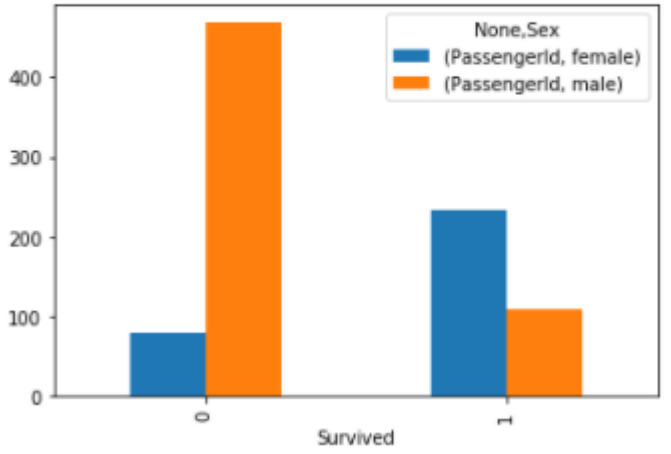
クロス集計のグラフ化

```
1 cross_01 = df.pivot_table(index=['Survived'], columns=['Sex'], values=['PassengerId'], ¥
2                           aggfunc='count', fill_value=0)
3 cross_01
```

	PassengerId	
Sex	female	male
Survived		
0	81	468
1	233	109

```
1 cross_01.plot(kind='bar')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a4c6a043c8>
```



aggfunc='count'を指定だから，どのラベルでも良い。ただし，indexとcolumnsで指定したラベル以外とする。

	PassengerId	
Sex	female	male
Survived		
0	81	468
1	233	109

columnsで指定

indexで指定

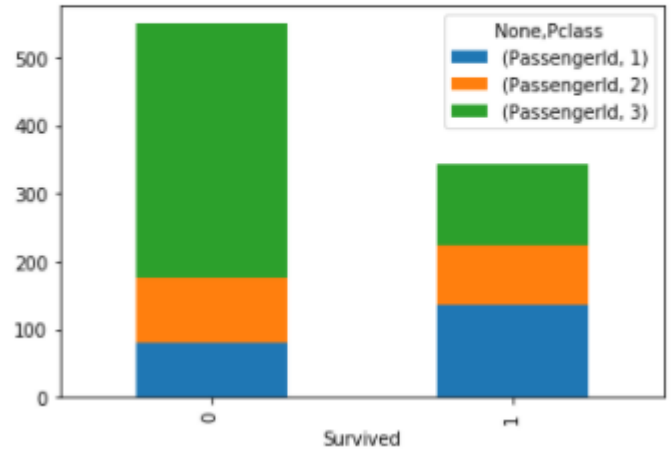
pandas Titanicデータのプロット

客室クラス別の生存

```
cross_02 = df.pivot_table(index=['Survived'], columns=['Pclass'], ¥
                          values=['PassengerId'], aggfunc='count', fill_value=0)
cross_02
```

```
cross_02.plot(kind='bar', stacked=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a4c6ae81d0>
```



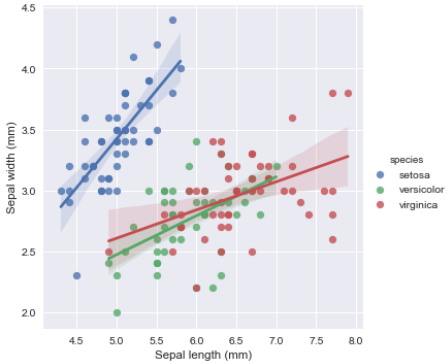
備考

- Titanicデータの分析は数多くなされている。これらの参照はデータ分析学習に役立つ。
- Titanicデータを見ると，女性の生存率が高いことがわかった。データを良く見ると子供の生存率も高いことが知られている。(Women and children first，古くからの船乗りの矜持)
- これらのことは，一般性があるのか，という疑問に対する調査分析を行った論文が次にある。これを見ると，Titanic号の事例は特殊で，一般性は無いようである。
- 論文：赤塚：Womend and children first，IFSMA便り，No.23，(社) 日本船長協会事務局，http://captain.or.jp/?page_id=4231

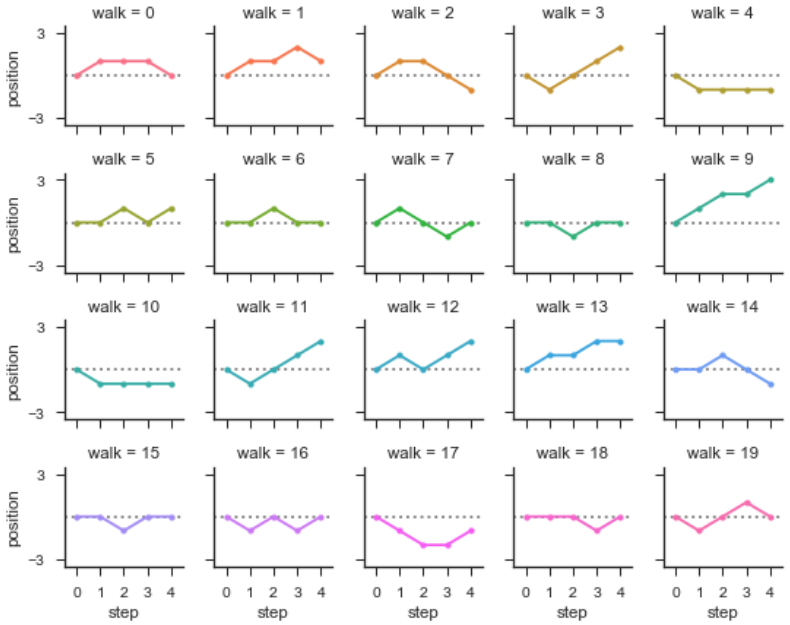
seaborn

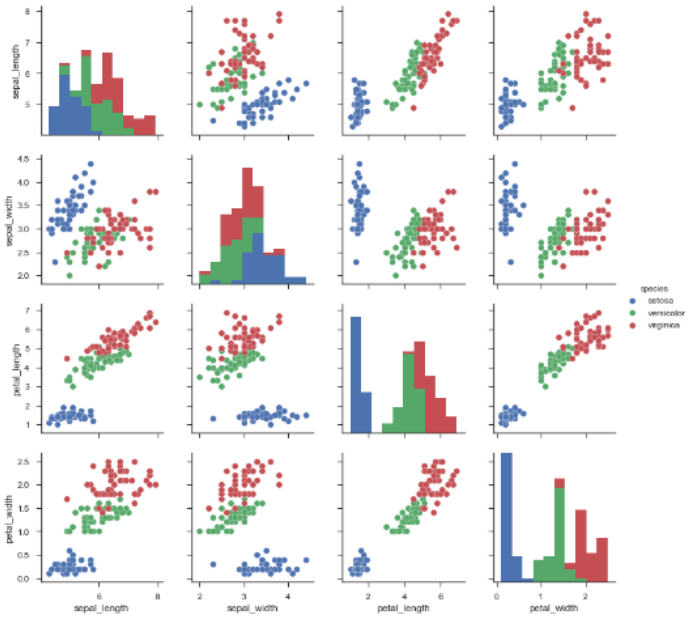
seaborn: statistical data visualization

- http://seaborn.pydata.org/



散布図と
単回帰モデル





seaborn irisデータのプロット

PLT_Iris

由来

- フィッシャー（Sir R.A. Fisher，英国，統計学者）のirisデータ（アヤメ）として統計学で良く用いられる

Irisの説明

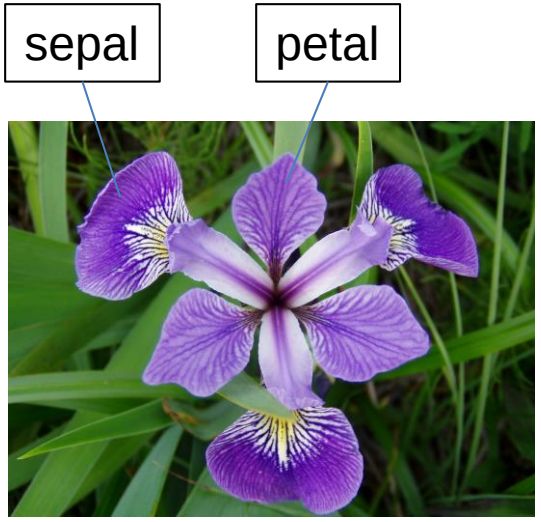
```
1 from IPython.display import Image
2 # Iris Setosa
3 url = 'https://upload.wikimedia.org/wikipedia/commons/a/a7/Irissetosa1.jpg'
4 Image(url,width=300, height=300)
```

- 参照：https://en.wikipedia.org/wiki/Iris_flower_data_set

seaborn irisデータのプロット

irisデータの説明

- アイリスの大きなはなびらがSepal（ がく片），小さな花びらがPetal（ 花弁）である．
- それぞれのlength（長さ）とwidth（幅）を特徴量として用いている
- 種類（species）は，次の3種setosa，versicolor，virginicaとしている（プログラムで見られる）




```
1  sns.set()
2  # Load the example tips dataset
3  iris = sns.load_dataset("iris")
4  iris.head()
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

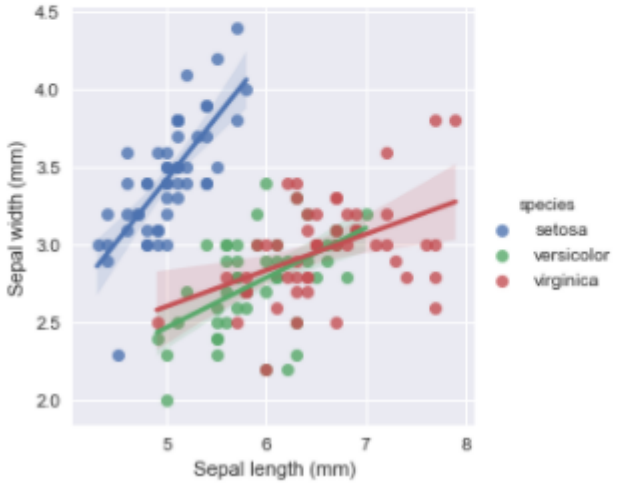
seaborn irisデータのプロット

回帰モデルの作成とプロット

- lmplot()は，回帰モデルの作成とグラフ化を同時に行う
- 次の例は，3種のirisそれぞれに対する回帰モデルを求めている。truncated =Trueは回帰モデルのプロットをデータの範囲内だけとする，=Falseは横軸レンジ一杯にプロットする。
- hueは色調，sizeはグラフのサイズ[inch]，統計量は出力しない。

```
[ ]:
1  # Plot tip as a function of toal bill across days
2  g = sns.lmplot(x="sepal_length", y="sepal_width", hue="species",
3                 truncate=True, size=4, data=iris)
4
5  # Use more informative axis labels than are provided by default
6  g.set_axis_labels("Sepal length (mm)", "Sepal width (mm)")
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x2c17c00cf98>
```



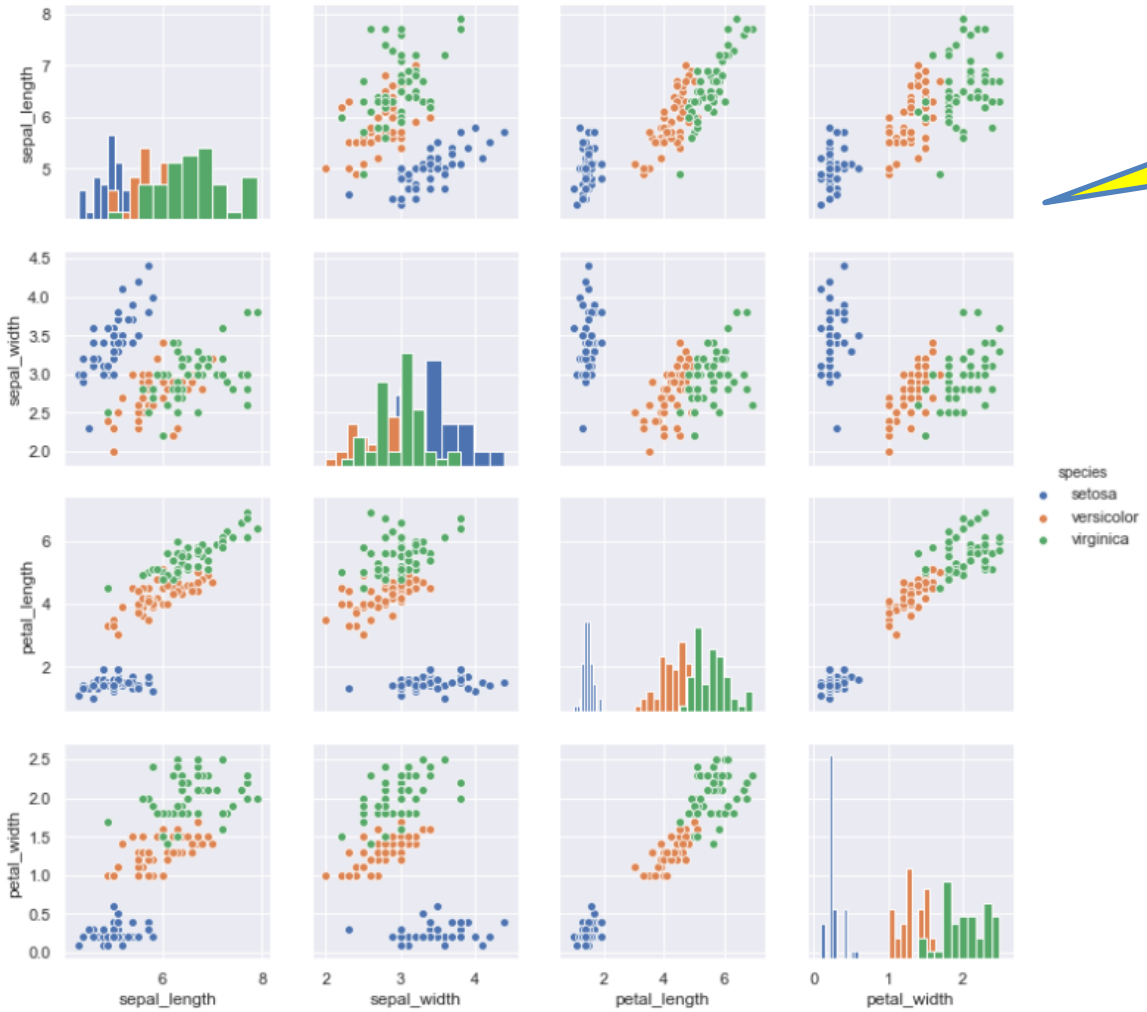
なぜ、直線を引く？
この意味は？
この見方を考えよう

- 参照:
- http://seaborn.pydata.org/generated/seaborn.lmplot.html
- https://seaborn.pydata.org/generated/seaborn.FacetGrid.html

seaborn irisデータのプロット

散布図行列

```
In [8]: 1 sns_plot = sns.pairplot(iris, hue="species", diag_kind='hist')
```



この見方を
考えよう

演習

グラフ化することの効用を考える

演習：ローン計算 （Loan calculation; 融資計算）

問題

- 50万円借りて、月々1万円返済する。金利が年利15%のとき、何カ月で、総額幾らの返済となるか？

数字で見ると，どう見る？

- month: 1　RisokuBun: 6249　GankinBun: 3751　Zandaka: 496249
- month: 2　RisokuBun: 6203　GankinBun: 3797　Zandaka: 492452
- month: 3　RisokuBun: 6155　GankinBun: 3845　Zandaka: 488607

このプログラムを実行して，何をどう考える？

- 先の問題の設定の場合の返済過程と
- 30万円前倒しして返還し，20万円借りたとしたときの返済過程
- この二つを比較すると，何が言える？
- **比較するには**，どのようにデータを表現したら良いか？

フォルダValuable_Kit内の
Ex_Loan　⇒　実行

備考：アルゴリズム（これを理解する必要は無い）

- 年利 inter_r　とおくと　月割り利息を inter_m = inter_a / 12
- 月々の返済金 pay，　前月の借入残高 debt とおくと
- 今月の利息分　　inter = debt0 * inter_m
- 今月の返済元金分　ret = pay - inter
- 今月の借入残高　　debt = debt0 - ret

余談：リボ払い（Revolving payments）が世にあるが、これは、元金分を一定にして、手数料（変動）を加え、この合算をほぼ一定にするローンの一種

演習：ローン計算 （Loan calculation; 融資計算）

考察する点

- 結果の表現を数字だけの場合と、グラフを見る場合の相違点はどこにあるか？
- グラフを見て、全ての線は直線でなく曲線である。
- それも、グラフは凹状、凸状である。この凹凸状から何が言えるか？

演習：CSVデータをプロットする

フィボナッチ数列（Fibonacci sequence）

ヒマワリの種の配置、オウムガイの殻、銀河系のらせん形、など自然界に見られる

n 番目のフィボナッチ数を F_n で表すと、F_n は再帰的に

$$F_0 = 0,$$
$$F_1 = 1,$$
$$F_{n+2} = F_n + F_{n+1} \ (n \geqq 0)$$

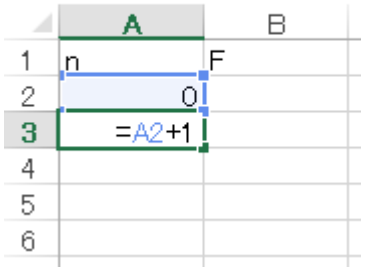
で定義される。これは、2つの初期条件を持つ漸化式である。

引用：https://ja.wikipedia.org/wiki/フィボナッチ数

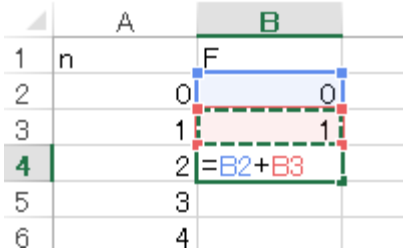
Excelファイルと
CSVファイルの
作成法を学ぶ

fibonacci.xlsxの作成

- Excelのオートフィル機能を用いる



セルA2に0を入力
セルA3を上記のように入力
セルA4以降はコピペ
A12：n=10まで作成



セルB2, B3に0,1を入力
セルB4に上記のように入力
セルB5以降はコピペ
B12まで作成

xlsxで保存した後に
CSVファイルで保存
ファイル名を
fibonacci.csv
とする。

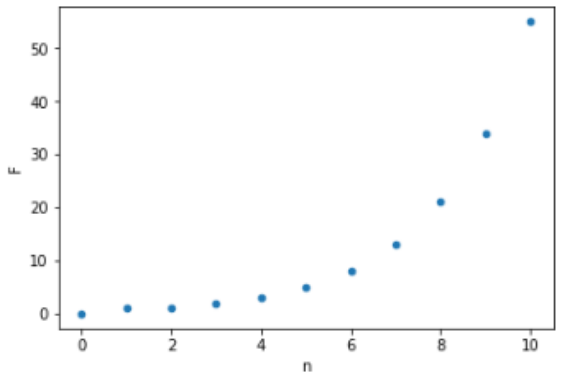
演習：CSVデータをプロットする（１）

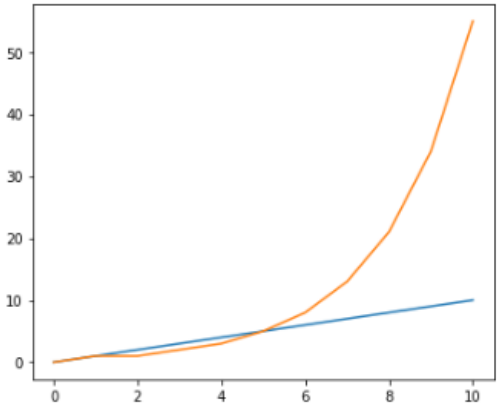
- pandasとmatplotlibで描く
- Fibonacci

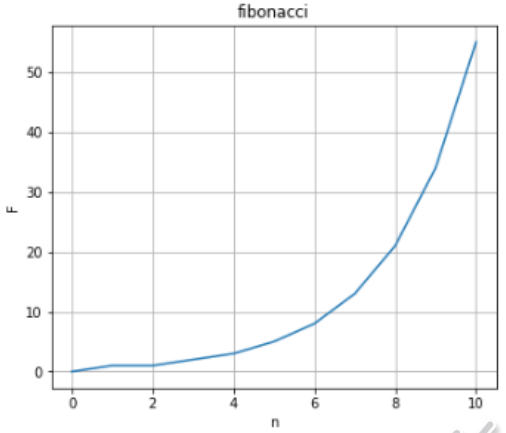
Fibonacci_Spiral
フォルダValuable_Kitの中にある

```
#pandas
df = pd.read_csv('fibonacci.csv') # df; DataFrame　の略
df.plot(kind='scatter', x='n', y='F')

#matplotlib
data = np.loadtxt('fibonacci.csv',delimiter=',', skiprows=1)
fig = plt.subplots( figsize=(6,5))
plt.plot(data)
plt.plot(data[:,0], data[:,1])
```







演習：CSVデータをプロットする（2）

- 螺旋（spiral）を描く。

直交座標における媒介変数表示として、

$$x(\theta) = r\cos\theta = ae^{b\theta}\cos\theta$$
$$y(\theta) = r\sin\theta = ae^{b\theta}\sin\theta$$

引用：https://ja.wikipedia.org/wiki/対数螺旋

- spiral.xlsxの作成

=EXP(D1*A2)*COS(A2*2*PI()/180)

=EXP(D1*A2)*SIN(A2*2*PI()/180)

	A	B	C	D
1	theta [deg]	x	y	-0.005
2	0	1	0	
3	10	0.936778	0.165179	
4	20	0.850269	0.309473	
5	30	0.745395	0.430354	
6	40	0.627184	0.52627	
7	50	0.500603	0.596596	

式中のb

Excelファイルの作成
実践してみよう！

=A2+10

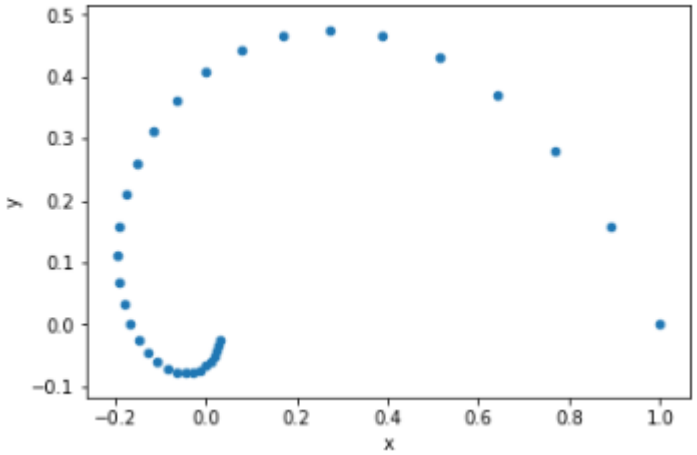
変換したCSVファイルから，D1を削除すること

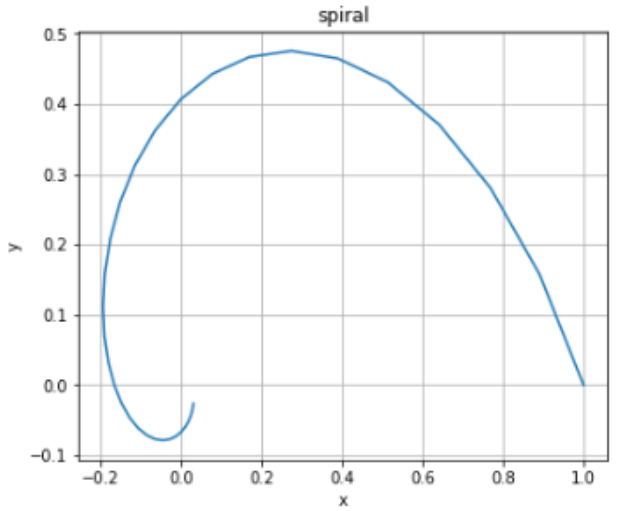
演習：CSVデータをプロットする（2）

Spiral

```
#pandas
df = pd.read_csv('spiral.csv') # df; DataFrame　の略
df.plot(kind='scatter', x='x', y='y')

#matplotlib
datax = df['x'].values
datay = df['y'].values
fig = plt.subplots( figsize=(6,5))
plt.plot(datax, datay)
```





付録

1. Practical Business PythonによるOverview of Python Visualization Tools (pandas, seaborn, matplotlibなどの比較）：http://pbpython.com/visualization-tools-1.html