データサイエンス特論
Data Science

回帰分析

1．　回帰分析とは
2．　単回帰分析
3．　多項式回帰分析

(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/

回帰分析とは
What is Regression Analysis

1. **単回帰分析**
   - **正規分布の見方**
2. **多項式回帰モデル**

説明変数が1つの場合の回帰分析を説明する。この回帰分析を行うPythonライブラリとして、機械学習で有名なscikit-learnや、pandasのメソッドがある。しかし，これらでは、p値などの各種統計量を計算されないので，統計的有意性の検証が行えず、結果の信頼性が保証できない。
ここでは，statsmodels（統計分析パッケージで時系列分析や一般化線形モデルなど様々な分析モデルに対応）を用いる。ただし、Rほどの機能は充実していないが，通常の分析をするなら十分である。このドキュメントは次にある。
http://www.statsmodels.org/stable/index.html

回帰という用語の由来

国語辞書による回帰の意味

- 一回りして元へ戻ること。(新明解辞典)
- y = ax + b (代表的な回帰式)を見ると, どうして元に戻るかわからない。
- 後に説明するAR(Auto-regression)モデルは, 回帰の意味がぴたりとはまる。

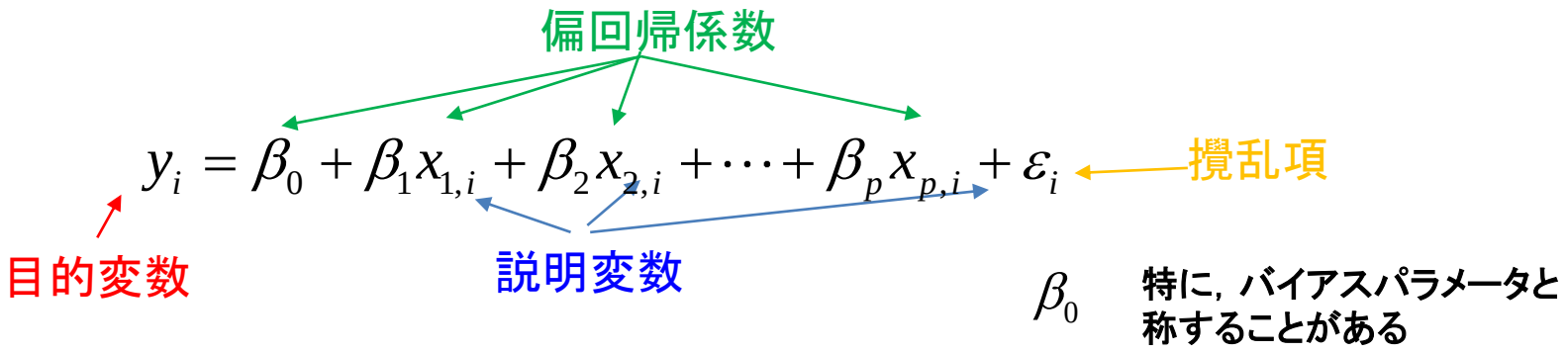
回帰の由来

- Sir Francis Galton (英, 統計学者, 1822 –1911)が次に意味で使ったことに由来する。
- 父親の身長と息子の身長との相関関係を調べ,
  - 背の高い父親 ＞ 息子の平均身長
  - 背の低い父親 ＜ 息子の平均身長
- 息子が父親になり, 息子を持てば・・・→ 世代を経ると, いずれ, 身長はある値に収束する。
- これをGaltonは**平均回帰**と言った。これから、慣習的に回帰という用語が使われるようになった。
- よって, 統計における回帰とは, 平均回帰の意味である。
- 現在の回帰分析では, 必ずしも直線関係だけではなく, 曲線(非線形)も用いられる。平均回帰は望めない。この場合,
  - 「目的変数(従属変数)yを説明変数(独立変数)xに回帰する」という意味で用いているらしい(私は理解できませんが, , , , ,)

参考文献

- データサイエンス・スクール(図がありわかりやすい) http://www.stat.go.jp/dss/course/701.htm
- Wikipedia, regression analysis https://en.wikipedia.org/wiki/Regression_analysis
- Wikipedia, Francis Galton https://en.wikipedia.org/wiki/Francis_Galton

用語の説明



□ 説明変数と目的変数について幾つかの表現

- 【説明変数 x】
  - 説明変数　explanatory variable
  - 予測変数　predictor variable
  - 独立変数　independent variable
  - 外生変数　exogenous variable -> **python statsmodels** で exog　（経済系が多い）
- 【目的変数 y】
  - 目的変数、応答変数、反応変数　response variable
  - 結果変数　outcome variable
  - 従属変数　dependent variable
  - 基準変数　criterion variable
  - 内生変数　endogenous variable -> **python statsmodels** で endog（経済系が多い）
- ○○変数（ variable ）ではなく○○変量（ variate ）とする本も多い。

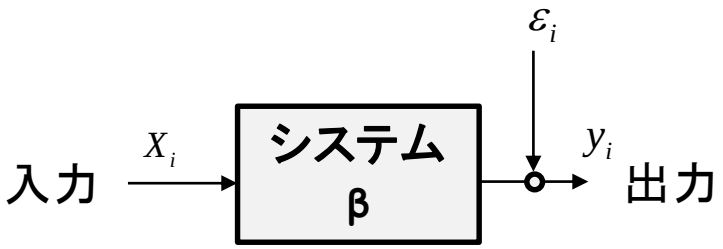
数学やシステム工学論から見ると

- 先ほどの回帰式で、複数のサンプル数を得たものとして、これを行列方程式で表現すると

$$\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\varepsilon}$$

- この連立方程式では、係数ベクトルβを求めることが問題となる
- 偏回帰係数が定数で，攪乱項＝0ならば線形システムである
- 攪乱項は，システム論的には外乱と見なせる。どちらも英語ではdisturbanceである
- 次の仮定，$\varepsilon_i \sim N\left(0,\sigma^2\right)$と置くことが多い。この仮定には留意する必要がある。

経済系では，バイアスパラメータ β_0 を問題にすることが多く，この場合には，行列式の外に出すこともある



- 統計分析を駆使する経済学，心理学，ファイナンス，社会調査分野では，beta value, beta coefficientと称してβという表記に意味を持たせている背景があるため，本書もそれに倣って回帰分析分野では係数をβとおく。
- 他の分野（数学，システム工学，ディジタル信号論，画像処理，機械学習など）では，係数を英小文字のaで表現することが多い。本書も，分野の主流に沿って係数の記号を変えることとする。

回帰を行う意義

右図を例にとる

条件

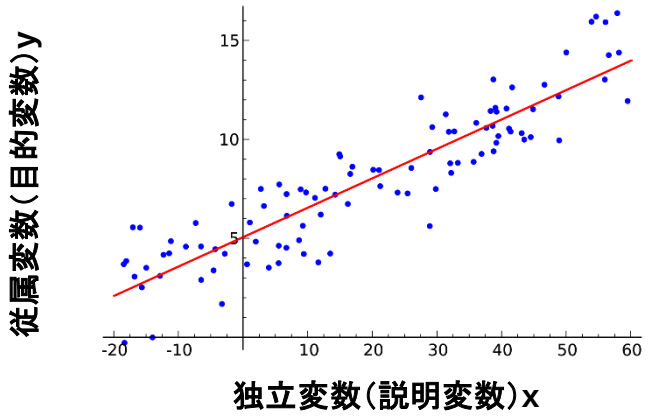
- 観測データは離散値である（青い点）。
- 横軸は独立変数（または説明変数と称される）x
- 縦軸は従属変数（または目的変数と称される）y
- この場合の独立とは，他の要因に影響を受けないという意味。例えば，時間など（ただし，量子力学論では重力や速度に影響を受ける）。
- この場合の従属とは，ある変数の影響を受けるものであり，この例では独立変数の影響を受けることを意味する。

離散値の特徴

- 離散点の間は無い（どういう意味？）
- xの範囲が有限である。この場合，x≦－20，または，x≧60の観測データは無い

要求

- 離散点の間の値を知りたい
- xの範囲外を知りたい　⇒　過去を推定，未来を予測などのため



https://en.wikipedia.org/wiki/Simple_linear_regression より引用

数式を当てはめる

- 適当な数式ならば，連続値を表現できるので，離散点の間の値，過去や未来の値を推定することができる。
- 当てはめ方（カーブフィッティング）
  - 観測データを必ず通る（？）
  - 観測データを通らなくても全体の傾向が示せれば良い

単回帰モデル

- $y = \beta_0 + \beta_1 x$ 一次式を当てはめる（図の赤い線）
- 切片と傾きをどうやって求める？
- 最小2乗問題

単回帰分析
(Simple Linear Regression)

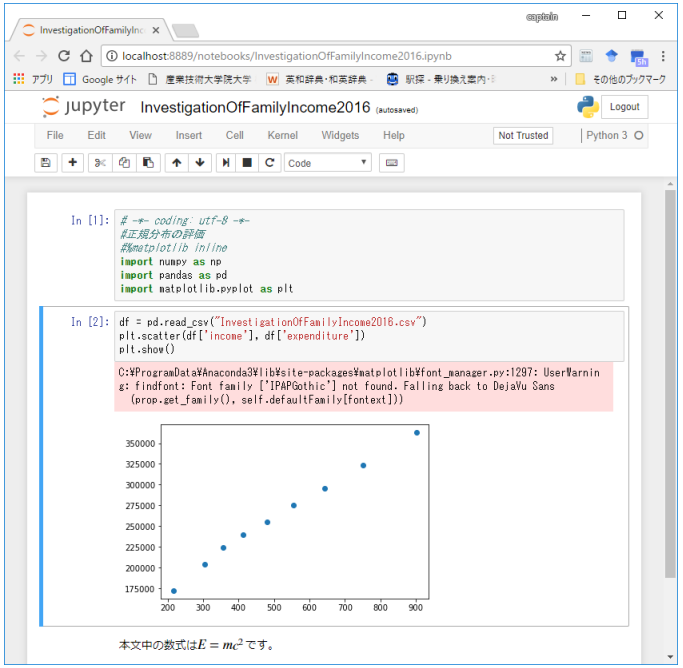
収入と支出の関係を見る

データ

- 家計調査結果，総務省統計局 http://www.stat.go.jp/data/kakei/index.htm
- 家計調査の概要，結果等 http://www.stat.go.jp/data/kakei/index2.htm#kekka
- 家計調査年報（家計収支編）2016年 http://www.stat.go.jp/data/kakei/2016np/index.htm
- **統計表e-Stat，二人以上の世帯** http://www.e-stat.go.jp/SG1/estat/List.do?lid=000001183264

  ⇒ **年間収入五分位・十分位階級別（二人以上の世帯・勤労者世帯）Excelデータより，「1世帯当たりの年間収入と1か月支出」データを抽出し，次のファイル名に保存した。**
- url ='https://sites.google.com/site/datasciencehiro/datasets/InvestigationOfFamilyIncome2016.csv'

1世帯当たり 年間収入 と 1か月支出



	A	B
1	income	expenditure
2	216	172462
3	304	204599
4	356	224776
5	413	240153
6	481	255497
7	555	275490

結果の評価

単回帰分析の場合の係数の評価

- 傾きβ_1, 切片β_0 は幾つか？

何を見る？

- 年収500万円の人の1か月支出は幾らか
- 年収200万円の人の1か月支出は幾らか

得られた回帰係数（パラメータ）は信頼できるのか？　⇒　検証を行う

表記は次とする。　(x_i, y_i)　$i = 1 \sim N$

- 取得データ
- 真のモデル　$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \quad \varepsilon_i \sim N(0, \sigma^2)$
- 推定モデル　$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$
- x, y の標本平均　$\hat{\mu}_x, \hat{\mu}_y$
- 回帰残差（residual）（モデル誤差とも言う）

$$e_i = y_i - \hat{y}_i = y_i - \left(\hat{\beta}_0 + \hat{\beta}_1 x_i\right)$$

この条件は理論を考えるときだけでなく，実データへ適用できるかを考えるときに重要

何らかの方法で，真のモデルを近似する推定モデルの係数を求めたものとする。
実際には，最小2乗法を用いて求める。

回帰係数（パラメータ）の性質

証明抜きで（他書参考），天下り的に説明

- **推定した回帰係数の統計的性質（実際には使わないが知っていることが大事）**
  - 不偏推定量
    - 右式のイメージ的理解が大事

$$E\left[\hat{\beta}_0\right]=\beta_0,\ E\left[\hat{\beta}_1\right]=\beta_1$$

  - 一致推定量
    - サンプル数nが増加→ $\sum_{i=1}^{N}(x_i-\hat{\mu}_x)\to\infty$ ならば，分散は0に近づく（通常は，∞になる）

$$\sigma^2_{\hat{\beta}_0}=V\left[\hat{\beta}_0\right]=\left(\frac{1}{N}+\frac{\hat{\mu}_x^2}{\sum_{i=1}^{N}(x_i-\hat{\mu}_x)}\right)\sigma^2$$

$$\sigma^2_{\hat{\beta}_1}=V\left[\hat{\beta}_1\right]=\frac{\sigma^2}{\sum_{i=1}^{N}(x_i-\hat{\mu}_x)}$$

  - 正規性
    - 右式のイメージ的理解が大事

$$\hat{\beta}_0\sim N\left(\beta_0,V\left[\hat{\beta}_0\right]\right),\quad \hat{\beta}_1\sim N\left(\beta_1,V\left[\hat{\beta}_1\right]\right)$$

上記が述べていることは、
何らかのライブラリ（Python statsmodels, R, SPSSなど）で求めた回帰係数は、
やはり確率変数であり、サンプル毎に異なる値を示す。

回帰分析における各分散

変動	平方和	分散	自由度
全変動	$\sum_{i=1}^{N}\left(y_i-\hat{\mu}_y\right)^2$	$\frac{\sum_{i=1}^{N}\left(y_i-\hat{\mu}_y\right)^2}{N-1}$	$N-1$
回帰変動	$\sum_{i=1}^{N}\left(\hat{y}_i-\hat{\mu}_y\right)^2$	$\frac{\sum_{i=1}^{N}\left(\hat{y}_i-\hat{\mu}_y\right)^2}{p}$	p
残差変動	$\sum_{i=1}^{N}\left(y_i-\hat{y}_i\right)^2=\sum_{i=1}^{N}e_i^2$	$\frac{\sum_{i=1}^{N}\left(y_i-\hat{y}_i\right)^2}{N-p-1}$	$N-p-1$

平方和の関係式

全変動　＝　回帰変動　＋　残差変動　　（※）

ここに，次の関係を用いた。

$$\left(y_i-\hat{\mu}_y\right)=\left(\hat{y}_i-\hat{\mu}_y\right)+\left(y_i-\hat{y}_i\right)$$

左辺の2乗の総和を考えたとき，右辺の二つの項は無相関であるから，(※)式が言える。

自由度

- 全変動の場合，不偏分散を使用している。回帰変動の場合，回帰係数（パラメータ）の数に依る。残差変動の場合，この両方を考慮しているため。

回帰モデルの評価

回帰モデル（回帰直線）のフィッティングの度合いを見る

決定係数(Coefficient of determination)

- 回帰直線のあてはまりの良さを表す指標

$$R^2 = \frac{\sum_{i=1}^{N}\left(\hat{y}_i - \hat{\mu}_y\right)^2}{\sum_{i=1}^{N}\left(y_i - \hat{\mu}_y\right)^2} = 1 - \frac{\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{N}\left(y_i - \hat{\mu}_y\right)^2} \qquad 0 \le R^2 \le 1$$

- R^2 が1に近いほど, あてはまりが良いと言える

自由度調整済み決定係数（adjusted coefficient of determination）

- 一般に，説明変数xの数pが増えると，下の自由度調整済み決定係数が用いられる。単回帰モデルでは関係無し。
- 説明変数が 1 で次数が2以上の場合（多項式回帰モデル），オッカムの剃刀原理（Occam's razor, ある事象を説明するのに、必要以上に多くのことを仮定しない。ケチの原理も類似した考え方）より、説明変数xの数をpとおいて、次のような自由度調整済み決定係数が提案されている
- 説明変数が2以上の場合（重回帰モデル（後述）），目的変数に関係のない説明変数を加えると決定係数が大きくなることがあり，これを避けるために用いる。

$$adj.R^2 = 1 - \frac{\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2 \Big/ \left(N - 1 - p\right)}{\sum_{i=1}^{N}\left(y_i - \hat{\mu}_y\right)^2 \Big/ \left(N - 1\right)}$$

決定係数の問題

この式において 1 に近いほど，良いあてはまりであるとされ，説明変数が目的変数をよく説明していると言われている。一般によく言われているのが，R^2 が 0.6 以下ならば良くないが，0.8 以上ならば，ある程度良いモデルであるとされている。しかし，この値は絶対的評価ではないので，0.6 以下ならば絶対にダメで，0.8 以上ならば絶対に良いとは言えない。

例えば，図 **5.3** において，単回帰モデル A の決定係数を R_A^2，モデル B のそれを R_B^2 としたとき，図 (a) ケース 1 の場合，$R_B^2 < R_A^2$ という結果はデータの分布から見て納得がいくであろう。図 (b) ケース 2 の場合，破線で囲まれたデータ群の影響により $R_A^2 < R_B^2$ となったとしよう。しかし，見た目では単回帰モデル A の方が良いようにも見える。この判断は，データの背景や使用条件に左右されるので，どちらのモデルが良いかの判断は

R^2 だけでは決められないことがある。この例が示すように，決定係数 R^2 の値は絶対的な指標ではなく，あくまでも目安であることを認識してほしい。

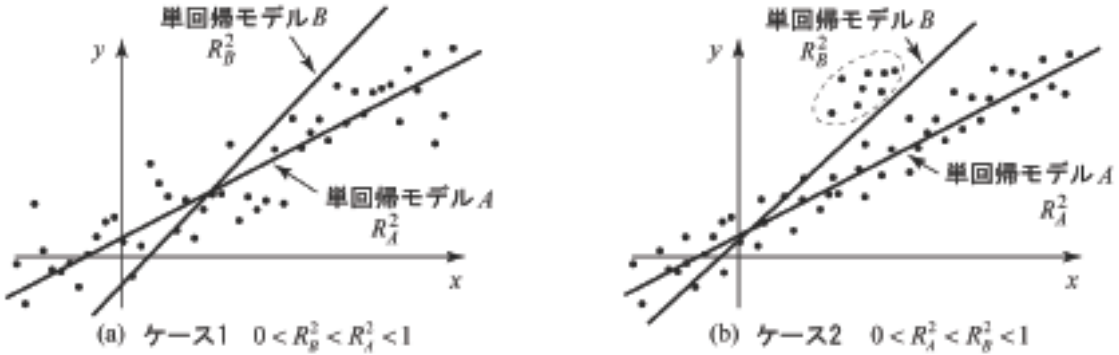


図 5.3: 決定係数 R^2 の例

偏回帰係数の検定

t検定

- 回帰係数 $\hat{\beta}_1$ がある特定の値(＝ ß_1)に等しいという仮説
  - H0： $\hat{\beta}_1 = ß\_1$　vs　H1: $\hat{\beta}_1 \neq ß\_1$ ← 両側検定です！
- t検定
  - 母分散σ^2は未知であるから，この代わりに次を用いる。

（※）

$$\hat{\sigma}^2 = \frac{1}{n-2}\sum_{i=1}^{N} e_i^2 \qquad e_i = y_i - \hat{y}_i = y_i - \left(\hat{\beta}_0 + \hat{\beta}_1 x_i\right)$$

  - これを代わりに用いた$\hat{\beta}_1$の分散を次のように表す

$$V\left[\hat{\beta}_1\right]_{\sigma^2 \to \hat{\sigma}^2} = \frac{\hat{\sigma}^2}{\sum_{i=1}^{N}(x_i - \hat{\mu}_x)}$$

  - これを用いて$\hat{\beta}_1$ の検定量を次式で表すと，これは自由度(N-2)のt分布に従う

$$t = \frac{\hat{\mu} - \mu_0}{\hat{\sigma}/\sqrt{N}} \sim t(N-1)$$

$$t = \frac{\hat{\beta}_1 - \beta_1}{\sqrt{V[\hat{\beta}_1]_{\sigma^2 \to \hat{\sigma}^2}}} \sim t(N-2)$$

自由度n-2のt分布
左辺のtと右辺のt()は
異なるものです

（※）n-2で割る理由は，係数を最小2乗法で求めると，e_iの総和は0，かつ，このベクトルと係数ベクトルとの内積が0となる，という二つの制約が加わり，自由度が2失われるためである。

偏回帰係数の検定

- **経済学や経営学で考える回帰分析では，しばしば傾きがゼロか否かを問題とする。**
  - xが収入，yを支出とするとき，β_1=0ならば，y = β_0（切片のみ）となり，支出は収入に全く関係無くなる。これは問題であると考えることが多い。
  - 有意水準5%の検定を考えると（下記の0.025は両側検定を考慮）

$$t = \frac{\hat{\beta}_1}{\sqrt{V[\hat{\beta}_1]_{\sigma^2 \to \hat{\sigma}^2}}} > t_{0.025}(N-2)$$

仮説H0を棄却

$$t = \frac{\hat{\beta}_1}{\sqrt{V[\hat{\beta}_1]_{\sigma^2 \to \hat{\sigma}^2}}} \leq t_{0.025}(N-2)$$

仮説H0を受容（棄却できないという意味）

Statsmodels

概要

- Pythonの統計ライブラリである。
- 通常の関数呼び出しと統計ソフトのRに似たスタイルの関数呼び出しの2種を提供している

Patsy

- Rスタイルの呼び出しを可能とするためのライブラリ、ユーザは意識することは不要である
- Patsy自身は、自分のことを“It’s only a model.”と称している
- このことは、http://patsy.readthedocs.io/en/latest/index.html Overviewに書いてある。
- Rスタイルとは、様々なモデルの形式を、例えば、“y ~ x + a + b + a:b”のようにテキストで表現できることをいう。

Statsmodelsのドキュメント

- Welcome to Statsmodels’s Documentation（http://www.statsmodels.org/stable/index.html）
- Getting started(http://www.statsmodels.org/stable/gettingstarted.html)に、データをpandasで表現すると述べている。

Rスタイルで回帰モデルを表現する formula.api

- 説明 http://www.statsmodels.org/dev/example_formulas.html
- Rスタイルの場合は、関数名を小文字とする、例：statsmodels.formula.api.ols()
- 従来スタイルの場合は、関数名を大文字とする。例：statsmodels.formula.api.OLS()
- Rスタイルをols()を用いても、数値計算はOLS()を用いる。

statsmodels OLS

OLS (Ordinary Least Squares)

- 数値計算分野の最小2乗法を用いたフィッティングを行う。
- ドキュメントは次を参照
  - Linear Regression http://www.statsmodels.org/stable/regression.html
  - Ordinary Least Squares http://www.statsmodels.org/stable/examples/notebooks/generated/ols.html
  - statsmodels.regression.linear_model.OLS http://www.statsmodels.org/stable/generated/statsmodels.regression.linear_model.OLS.html

Rスタイルのモデル式の表現

モデル式	式の内容
y ~ x	単回帰モデル。yはxにより説明される
y ~ x1 + x2	重回帰モデル。yは独立変数 x1 と x2 により説明される
y ~ x1 + x2 - 1	y は切片（intercept）を使用せずに、x1とx2のみで 説明される
y ~ x1:x2	yは交互作用項 (x1 * x2) で説明される
y ~ x1 * x2	yはx1 と x2、及び交互作用項 (x1 * x2) で説明される (x1 + x2 + x1 * x2 と同じ)

左の表にない表現：
y～x－１　左のように“-1”を指定することで、切片（intercept）が無く、原点を通る単回帰モデルを表す

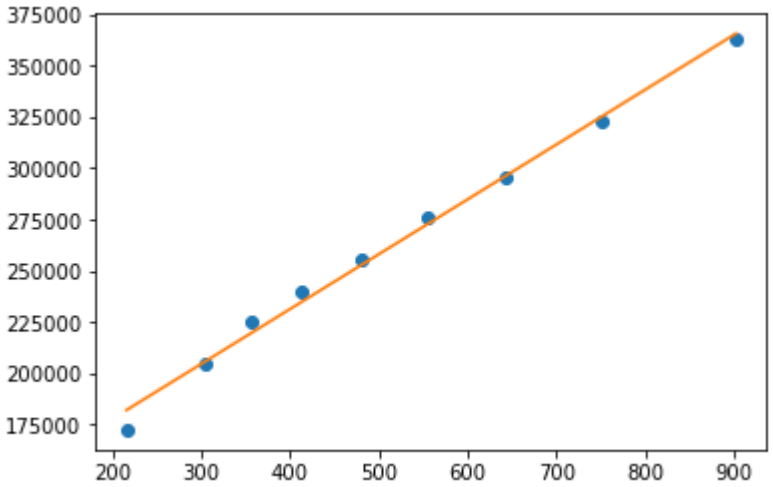
Ref.
https://ponvire.com/2017/07/04/python%E3%81%A7r%E3%81%A8%E5%90%8C%E3%81%98%E3%82%88%E3%81%86%E3%81%AB%E7%B7%9A%E5%BD%A2%E5%9B%9E%E5%B8%B0/

結果の評価：収入と支出の関係を見る

REG_Simple_FamilyIncome

```
import statsmodels.formula.api as smf

url = 'https://sites.google.com/site/datasciencehiro/datasets/FamilyIncome.csv'
df = pd.read_csv(url, comment='#')
result = smf.ols('expenditure ~ income', data=df).fit()
print(result.summary())
```



- y = β_0 + β_1 x を　expenditure = income と表現している。
- “－1”を付けると切片（定数項）を考慮しなくなる。
- 得られたパラメータを用いた、元データ（○印、散布図）に1次の回帰モデルを重ねて描画した。
- このモデルが適切か否かは、見ただけ評価しても良い場合があるが、何らかの定量的評価が必要ならば、それを次のページで述べる。

結果の評価

右の結果を必要な個所だけ説明する。

使ったモデルと数値計算手法を次で示している
Model:　　OLS
Method:　　Least Squares

決定係数と自由度調整済み決定係数が次である
R-squared:　　0.994
Adj. R-squared:　　0.993

coef : 偏回帰係数、求めたパラメータ値
切片(Intercept)　が1.242 e+05,　傾きが267.501
std err : パラメータの標準誤差。ばらつきがあるのかどうか
t : 有意かどうかの際に用いる値
P>|t| : 推定されたパラメータがゼロである確率
いずれも P = 0.000　であるから、両方のパラメータともゼロであるという仮説は棄却される。
[0.025　　0.975]は信頼区間で、95%の確率でこの区間内に値が収まる

OLS Regression Results

Dep. Variable:	expenditure	R-squared:	0.994
Model:	OLS	Adj. R-squared:	0.993
Method:	Least Squares	F-statistic:	1111.
Date:	Fri, 29 Dec 2017	Prob (F-statistic):	5.66e-09
Time:	09:35:20	Log-Likelihood:	-88.364
No. Observations:	9	AIC:	180.7
Df Residuals:	7	BIC:	181.1
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	1.242e+05	4449.025	27.925	0.000	1.14e+05	1.35e+05
income	267.5016	8.024	33.339	0.000	248.528	286.475

Omnibus:	2.156	Durbin-Watson:	0.764
Prob(Omnibus):	0.340	Jarque-Bera (JB):	0.752
Skew:	-0.706	Prob(JB):	0.687
Kurtosis:	2.906	Cond. No.	1.47e+03

`Intercept`の値があるから，比例ではない。
これをどう考える？

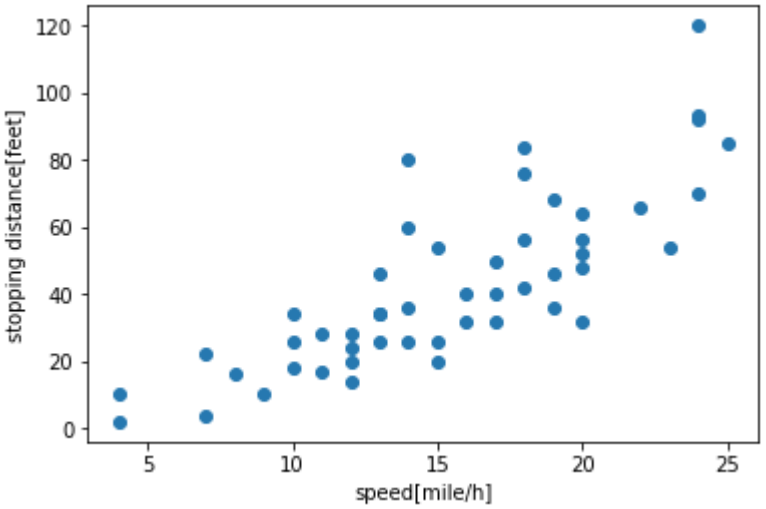
多項式回帰分析
（Polynomial Regression Analysis）

1. 多項式とは
2. シミュレーション例
3. 次数の選定

曲線のカーブフィッティングを行いたい

Rのdatasets cars（この使用は付録参照）

- 統計計算ソフトウェアRが持つdatasetsの中にcarsがある。
- carsは，車の速度と停止距離（1920年代のデータ）
- データフレーム　50×2
- speed　時速（マイル/h）
- dist　停止までの距離（フィート）



右のデータを見て

- 速度と停止距離は比例関係ではないような気がする
  - 比例関係：y = ax　ではなさそう
  - または，一次関数の関係でない，とも言う
  - 工学では，停止距離∝speed^2 と言われる
- 多項式で表現して見たい（次ページ参照）
- 例えば，
  - y = ß_0 + ß_1 x + ß_2 x^2　（2次式）
  - y = ß_0 + ß_1 x + ß_2 x^2 + ß_3 x^3　（3次式）
  - それぞれの多項式のグラフの概形をイメージできるようにする。
  - ß_0　必要？　（物理的，また，現実的な視点から，どう考えますか？）

多項式とは

多項式(polynomial equation)の表現

- 変数の計算が加減乗(+, -, ×)のみの形で表現される。例えば，xを変数としたとき，次の形式をいう。

$$f(x) = a_p x^p + a_{p-1} x^{p-1} + \cdots + a_1 x + a_0 = \sum_{i=0}^{p} a_i x^i$$

中学，高校で習った
y = ax^3 + bx^2 + cx +d
はこの形式

- ここに，a_i (i = 0, 1, …., p)は定数である。
- 数学的な表現としては
  - 上式は**項 (term)**の和で表現されている
  - 次数の最も高い項の次数を，その多項式の**次数 (degree)** という
- 統計の分野で，上式の右辺は，定数項a0を左に置く。これは，統計や経済の分野では定数項を問題にすることが多いためである。

カーブフィッティング（curve fitting）

カーブフィッティングとは

- 得られたデータに（ある意味で）最もよく当てはめる曲線を求めること
- データを必ずしも通るとは限らない
- これに対し，データを必ず通る曲線を当てはめることを補間（interpolation，内挿と言うこともある）といい，スプラインやベジェ曲線が良く用いられる
- なお，多項式のみならず，様々な曲線関数（三角関数，tanh，sinc，シグモイド（ロジスティックともいう）関数など）の当てはめがある。ここでは，多項式のみに注目する

Pythonでカーブフィッティングを行う

- Numpy：polyfit
- Scipy：optimize.leastsq
- Scipy：optimize.curve_fit
- 計算結果はほぼ同じ，使い勝手はpolyfitが一番良い。多項式の関数を定義しなくてもよい。
- 後ほど，statasmodelsを用いた例を示す

numpy.polyfit

- 最小2乗法によるカーブフィッティング（see SciPy.org）
- https://docs.scipy.org/doc/numpy/reference/generated/numpy.polyfit.html

numpy.poly1d

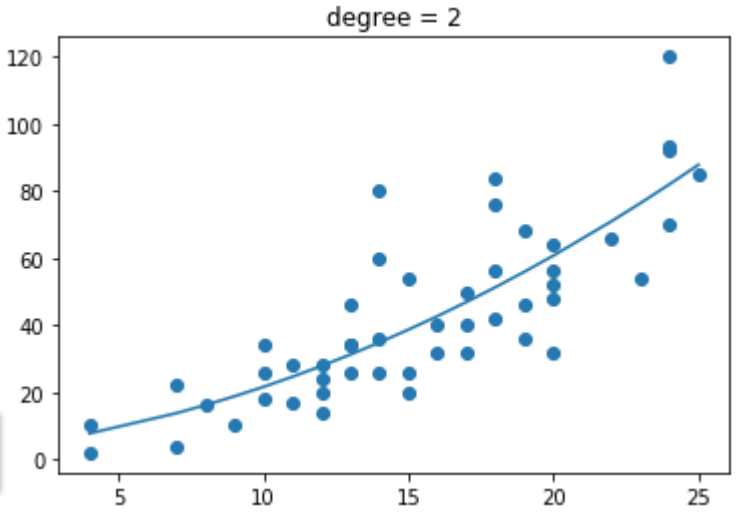
- 多項式の係数を与えて，多項式の関数を生成する
- https://docs.scipy.org/doc/numpy/reference/generated/numpy.poly1d.html

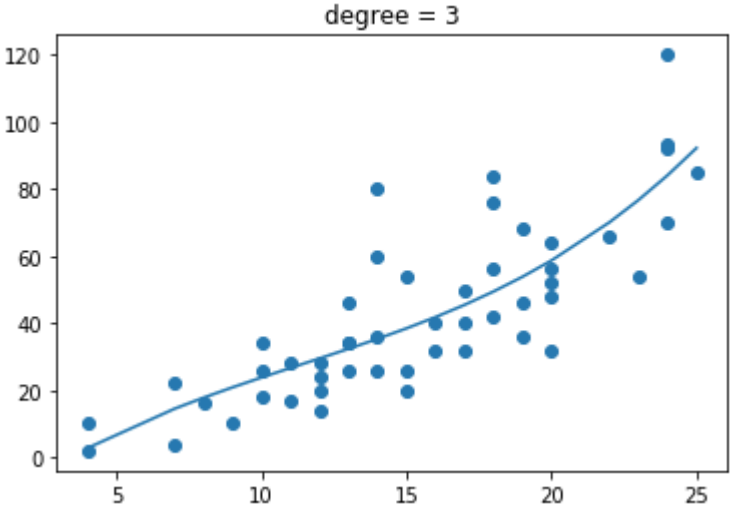
シミュレーション結果

次数が2，3のときの結果

非線形ゆえ，検定での評価は難しい

見た目，または，誤差の量を見る





REG_Poly_cars

次数を変えた実習

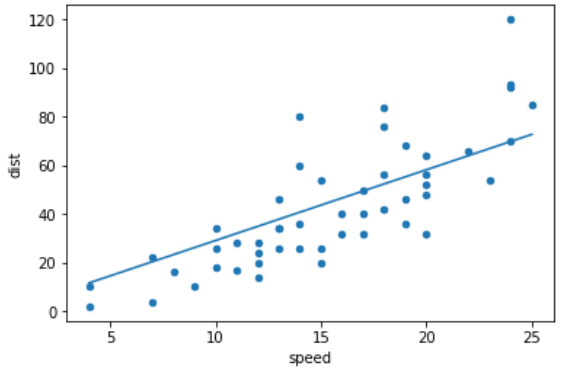
```
# 事前に取得したRデータセットのcarsのデータを用いる
url =
"https://sites.google.com/site/datasciencehiro/datas
ets/cars_R_datasets.csv"
df = pd.read_csv(url)  # read datasets of cars

x = df.speed
#y = df['dist']
result2 = smf.ols('dist ~ np.power(speed,2) + speed',
data=df).fit()
print(result2.summary())
b0, b2, b1 = result2.params
```

次数が9次のとき，オーバーフィッティングが生じる
数値計算では，x^nに比例した項の計算がある。この項は，データの密集した部分の影響を強く表す。
全体としてのある種の最小化問題を解くと，データの疎の部分に，先の影響を吸収しようとする現象が現れる。
これは，補間問題ではあるが，古典的なルンゲの問題と類似している。次数の選定は，次に述べることも考慮する。

statsmodels olsを用いる 1次式の考察

- y = β_1 x のモデルの場合
- 右に回帰モデルのグラフと統計的諸量
  - speedの係数は 2.9091, すなわち、速度が1mile/h上昇する毎に、停止距離は平均的に2.9091フィート増える。ただし、標準偏差が0.141　だけのばらつきがある。
  - p値は0.000 より、speedの係数が0であるという仮説は破棄される。
  - 次のモデルの良さの指標は後で考察される
    - 決定係数　R-squared: 0.896
    - 修正決定係数 Adj. R-squared: 0.894
    - AIC: 421.7
  -



OLS Regression Results

Dep. Variable:	dist	R-squared:	0.896
Model:	OLS	Adj. R-squared:	0.894
Method:	Least Squares	F-statistic:	423.5
Date:	Thu, 28 Dec 2017	Prob (F-statistic):	9.23e-26
Time:	18:03:22	Log-Likelihood:	-209.87
No. Observations:	50	AIC:	421.7
Df Residuals:	49	BIC:	423.7
Df Model:	1		
Covariance Type:	nonrobust		

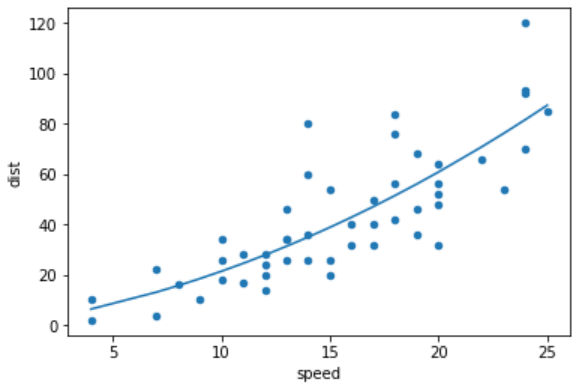
	coef	std err	t	P>|t|	[0.025	0.975]
speed	2.9091	0.141	20.578	0.000	2.625	3.193

Omnibus:	14.345	Durbin-Watson:	1.409
Prob(Omnibus):	0.001	Jarque-Bera (JB):	15.573
Skew:	1.202	Prob(JB):	0.000415
Kurtosis:	4.302	Cond. No.	1.00

statsmodels olsを用いる 2次式の考察

- y = β_1 x + β_2 x^2 のモデルの場合
- 右に回帰モデルのグラフと統計的諸量
  - speedの係数は 1.2390, speed^2 の係数は0.0901、それぞれの標準偏差も示されている。
  - それぞれのp値は0.032, 0.004より、有意水準を5%に取るならば、それぞれの係数が0であるという仮説は破棄される。1%ならば、このことは言えない。
  - 次のモデルの良さの指標を考える
    - 決定係数　R-squared: 0.913
    - 修正決定係数 Adj. R-squared: 0.910
    - AIC: 414.8
  - 1次式の場合と比べて、R-squared, Adj. R-squaredともに上昇、かつ、AICは減少、
  - よって、2次式の方が1次式の場合より、統計的にモデル表現は良い、と示された。

```
result = smf.ols('dist ~ np.power(speed,2) + speed -
1', data=df).fit()
result.summary()
a,b = result.params
print(a,b,c)
print(type(a),type(b), type(c))
df.plot(kind='scatter', x='speed', y='dist')
plt.plot(x, b*x+a*(x**2))
#plt.show()
```



OLS Regression Results

Dep. Variable:	dist	R-squared:	0.913
Model:	OLS	Adj. R-squared:	0.910
Method:	Least Squares	F-statistic:	252.8
Date:	Thu, 28 Dec 2017	Prob (F-statistic):	3.27e-26
Time:	17:57:45	Log-Likelihood:	-205.40
No. Observations:	50	AIC:	414.8
Df Residuals:	48	BIC:	418.6
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
np.power(speed, 2)	0.0901	0.029	3.067	0.004	0.031	0.149
speed	1.2390	0.560	2.213	0.032	0.113	2.365

Omnibus:	10.823	Durbin-Watson:	1.763
Prob(Omnibus):	0.004	Jarque-Bera (JB):	10.587
Skew:	0.971	Prob(JB):	0.00502
Kurtosis:	4.144	Cond. No.	81.9

statsmodels olsを用いる

AIC, R-squaredの説明

- http://wcs.hatenablog.com/entry/2016/11/08/231703
- Log-LikelihoodとAdj. R-squared（いずれも高いほうが良い）は変数を増やすのにつれて単調増加している。
- 一方AIC（低いほど良い）とAdj. R-squared（高いほうが良い）は向上が頭打ちになるポイントがある。AICの場合[complaints, learning]を説明変数に使ったときが最もよく、Adj. R-squaredの場合[complaints, learning, advance]を使ったときが最もよい。

```
In [43]: model = smf.ols('dist ~ speed', data=df)
         result = model.fit()
         result.summary()
```

Out[43]:

OLS Regression Results

Dep. Variable:	dist	R-squared:	0.651
Model:	OLS	Adj. R-squared:	0.644
Method:	Least Squares	F-statistic:	89.57
Date:	Tue, 26 Dec 2017	Prob (F-statistic):	1.49e-12
Time:	13:41:48	Log-Likelihood:	-206.58
No. Observations:	50	AIC:	417.2
Df Residuals:	48	BIC:	421.0
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-17.5791	6.758	-2.601	0.012	-31.168	-3.990
speed	3.9324	0.416	9.464	0.000	3.097	4.768

Omnibus:	8.975	Durbin-Watson:	1.676
Prob(Omnibus):	0.011	Jarque-Bera (JB):	8.189
Skew:	0.885	Prob(JB):	0.0167
Kurtosis:	3.893	Cond. No.	50.7

```
In [50]: model = smf.ols('y ~ np.power(x,2) + x', data=df)
         result = model.fit()
         result.summary()
```

Out[50]:

OLS Regression Results

Dep. Variable:	y	R-squared:	0.667
Model:	OLS	Adj. R-squared:	0.653
Method:	Least Squares	F-statistic:	47.14
Date:	Tue, 26 Dec 2017	Prob (F-statistic):	5.85e-12
Time:	13:45:53	Log-Likelihood:	-205.39
No. Observations:	50	AIC:	416.8
Df Residuals:	47	BIC:	422.5
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	2.4701	14.817	0.167	0.868	-27.338	32.278
np.power(x, 2)	0.1000	0.066	1.515	0.136	-0.033	0.233
x	0.9133	2.034	0.449	0.656	-3.179	5.006

Omnibus:	11.173	Durbin-Watson:	1.762
Prob(Omnibus):	0.004	Jarque-Bera (JB):	11.061
Skew:	0.991	Prob(JB):	0.00396
Kurtosis:	4.173	Cond. No.	2.16e+03

カーブフィッティングの方法

考察

- 物理的観点から，速度0のとき停止距離は0である
- 定数項を不要とする曲線モデルが必要

curve_fit

- 曲線モデルを自作できる

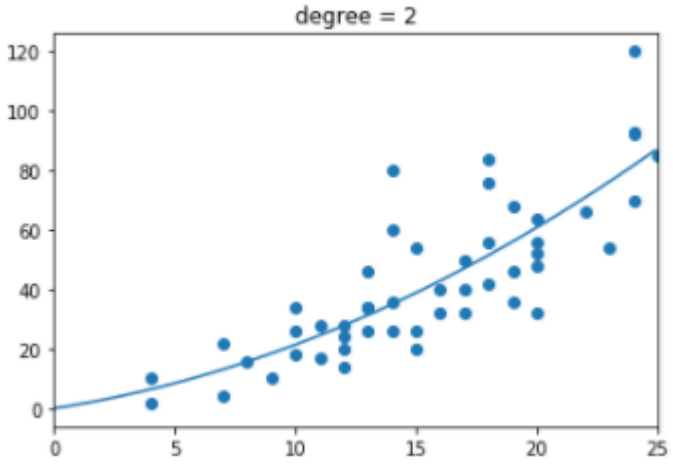
定数項が無い

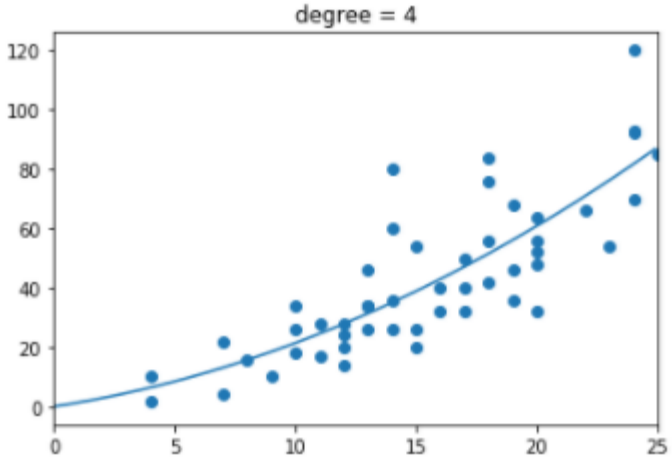
```
# フィッティングする曲線を2次とする
def func(x, b1, b2):
    return b1*x + b2*x**2

parm, cov = scipy.optimize.curve_fit(func, x, y)
```

結果

- 次数2と4の結果を示す
- 視認より次数2で十分であり，
  物理的モデルの妥当性がいえる





scipy.optimize.curve_fit
https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html

次数の決定は

次数をいろいろ変えて、計算結果を比較することになる。この際、次の指標を見ることとなる。

◻ 決定係数からの評価

- R^2, Adj.-R^2 は、見た目とおよそ一致するので使いやすい。ただ、説明変数の種類や次数が上がれば、これらの値は一般に向上する。
- 現実には、あまり次数が高くない場合（1～4程度）は、決定係数から決定することが多い。

◻ AICからの評価

- 詳細は他書に譲るとして、AICは次式で定義される。

  AIC = －2×（モデルの最大対数尤度）＋2×（モデルのパラメータ数）
- 正規分布を仮定する線形回帰モデルのAICは次のように変形して表現されることが知られている

  AIC = N ×log（残差の2乗和/ N）＋2×（モデルのパラメータ数）
- この式を大まかに解釈すると，残差の2乗和とパラメータ数のトレードオフを見ている。
- パラメータ数を多くとると残差の2乗和は減少、逆も然り。このトレードオフがある場合、あるパラメータ数のときにAICは最小を取ることが期待される。この最小値を指標として採用するという考え方である。
- このため、AIC最小が常に最良かどうかを一概にはいえない。そのため、モデル選択指標として、BIC（Bayesian information criterion）、MDL（minimum description length）などが提案されている

次数の決定は

次数をいろいろ変えて、計算結果を比較することになる。この際、次の指標を見ることとなる。

p値

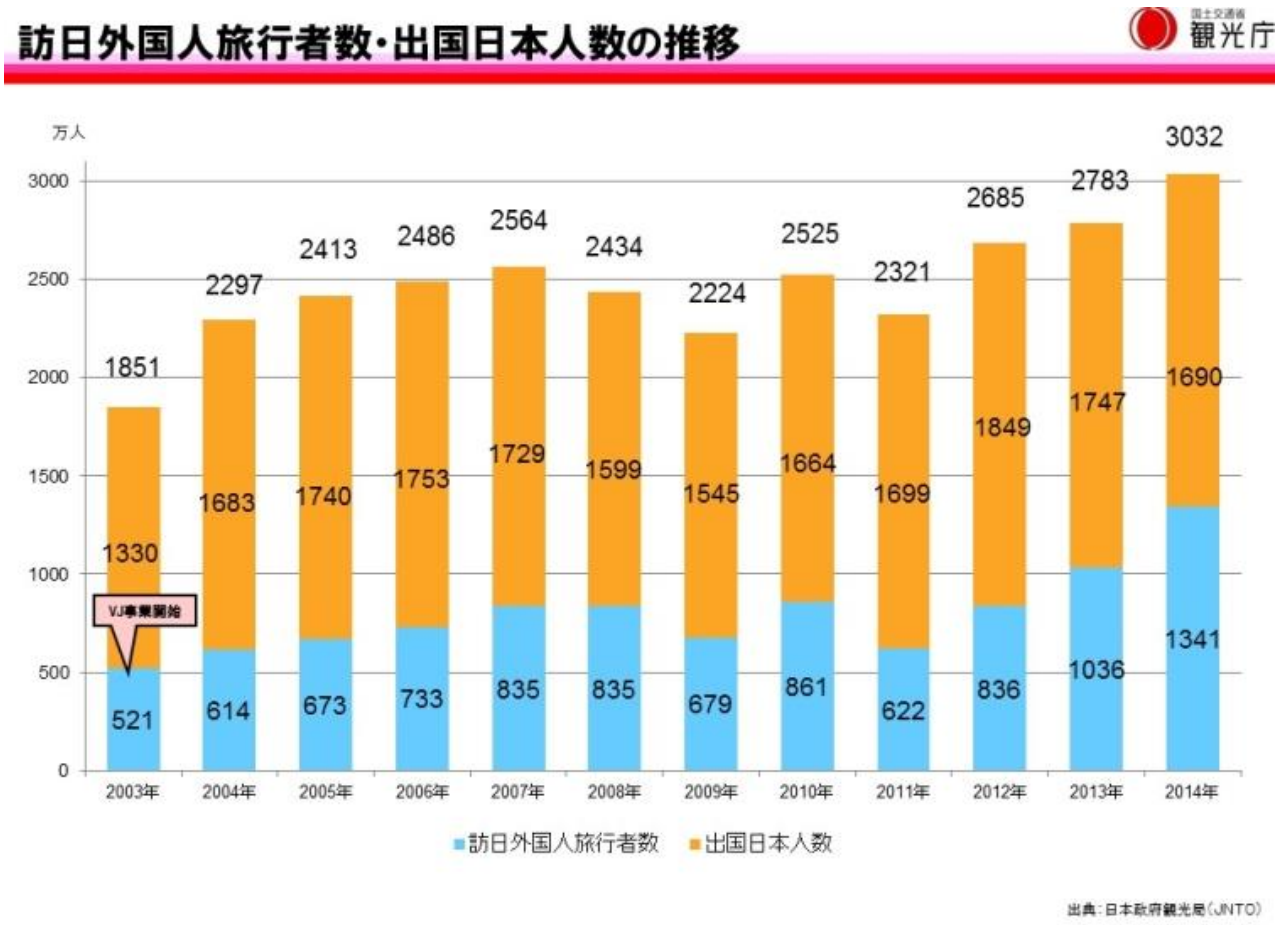
- 複雑なモデル(次数の多いモデル)の方が説明力が高いため、説明力の高いモデルですべてのp値が有意であれば、これを採用することが多い。

現象の物理モデルが既知の場合

- 例えば、電気回路のRLC回路であると分かっている場合には、統計的評価指標とは関係なく、2次の回帰モデルを採用すべき、という考え方もある。先の、スピード、停止距離の例では、物理論から　およそ、停止距離∝速度^2　がわかっているから、2次の回帰モデルを適用した方が結果の説明がしやすい。物理モデルが不明な場合は、上記の指標を多角的に見ざるをえない。

例：外国人旅行者数の推移

- 訪日外国人旅行者数・出国日本人数の推移（観光庁）
  - http://www.mlit.go.jp/kankocho/siryou/toukei/in_out.html
  - 次をどう扱う？　（各自に委ねます）



付録

statsmodels.regression.linear_model.RegressionResults

- http://www.statsmodels.org/dev/generated/statsmodels.regression.linear_model.RegressionResults.html
- **AIC**
  - Akaike's information criteria. For a model with a constant -2llf + 2(df_model + 1). For a model without a constant -2llf + 2(df_model).
- **R-squared**
  - R-squared of a model with an intercept. This is defined here as 1 - ssr/centered_tss if the constant is included in the model and 1 - ssr/uncentered_tss if the constant is omitted.

非線形のカーブフィッティング問題

- **数値計算の分野では，回帰モデルを言わずに，非線形関数を当てはめるカーブフィッティング問題が数多く研究されてきた。この場合**
  - データを産み出す数式モデルを仮定して，そのパラメータ推定の問題と考える
  - カーブフィッティング（必ずしも点は通らない）と補間問題（interpolation，必ず点を通る）とは異なることの注意
  - 補間問題　ルンゲの問題
- **参考**
  - Wikipedia, Curve fitting: https://en.wikipedia.org/wiki/Curve_fitting
  - Wikipedia, Interpolation: https://en.wikipedia.org/wiki/Interpolation
  - Wikipedia, Runge's phenomenon: https://en.wikipedia.org/wiki/Runge%27s_phenomenon

付録
Pythonで
Rのデータセットを使う

rpy2を用いる：
rpy2 is an interface to R running embedded in a Python process, and also includes functionality to deal with pandas DataFrames.
http://pandas.pydata.org/pandas-docs/stable/r_interface.html#updating-your-code-to-use-rpy2-functions

インストール rpy2

インストール手順

- コマンドプロンプトから次のコマンドを実行する。

>conda install rpy2

ここで，次のエラーが出たとき

```
CondaIOError: Missing write permissions in: C:¥ProgramData¥Anaconda3
#
# You don't appear to have the necessary permissions to install packages
# into the install area 'C:¥ProgramData¥Anaconda3'.
# However you can clone this environment into your home directory and
# then make changes to it.
# This may be done using the command:
#
# $ conda create -n my_root --clone="C:¥ProgramData¥Anaconda3"
```

- エラーを見ると” the necessary permissions”と書かれているので、” C:¥Program Files¥Anaconda3″のフォルダに書き込み権限がないことがわかる。
- 対処法
  - 管理者権限でコマンドプロンプトを起動する
  - その方法は，**スタート→Windowsシステムツール→コマンドプロンプト**を選択し、そのまま実行するのではなく、コマンドプロンプトを右ボタンクリックし、**その他→管理者**として実行を行い、コマンドプロンプトを管理者権限で実行する。

rpy2のマニュアル

- http://pandas.pydata.org/pandas-docs/stable/r_interface.html#updating-your-code-to-use-rpy2-functions

データ

UCI　machine learning

- http://archive.ics.uci.edu/ml/index.php
- UCI 機械学習リポジトリのデータセット一覧 https://www.trifields.jp/uci-machine-learning-repository-datasets-956

RのデータセットをPythonから用いる

- https://qiita.com/arc279/items/eceb9510593c223e44c2
- http://pandas.pydata.org/pandas-docs/stable/r_interface.html#updating-your-code-to-use-rpy2-functions
- http://ayaneru.github.io/blog/2015/02/26/20150226/
- rpy2をインストールできない（2017年12月）

エラー対策
パッケージをインストールしようとしたら
CondaIOError: Missing write permissions in: C:¥ProgramData¥Anaconda3
#
You don't appear to have the necessary permissions to install packages
into the install area 'C:¥ProgramData¥Anaconda3'.
However you can clone this environment into your home directory and
then make changes to it.
This may be done using the command:
#
$ conda create -n my_root --clone="C:¥ProgramData¥Anaconda3“

この対処法
http://imagingsolution.net/program/python/anaconda/module_update_error/

Rのパッケージ datasets 情報

インストールが成功すると

- In [1]: from rpy2.robjects import r, pandas2ri
- In [2]: pandas2ri.activate()
- In [3]: r.data('iris')
- In [4]: r['iris'].head()
- Out[4]:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa

パッケージ “datasets”の説明

- 原文：
  - https://www.rdocumentation.org/packages/datasets/
  - https://www.rdocumentation.org/packages/datasets/versions/3.4.3
- 日本語訳（非公認のよう）
  http://www.okadajp.org/RWiki/?%E3%83%91%E3%83%83%E3%82%B1%E3%83%BC%E3%82%B8%20%27datasets%27%20%E3%81%AE%E6%83%85%E5%A0%B1

End