データサイエンス特論
Data Science

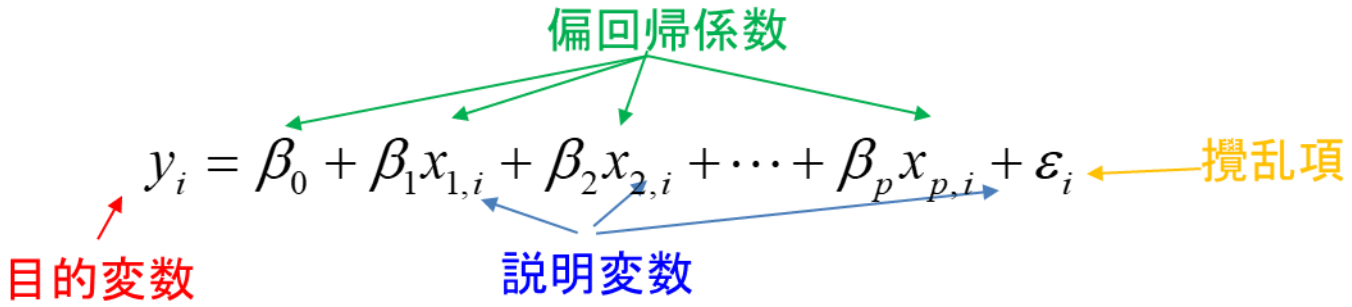
回帰分析
多変数の回帰分析

1. 重回帰分析(Multiple Variate Regression Analysis)
2. 一般化線形モデル

重回帰分析

重回帰モデルとは

- 目的変数yが、複数の説明変数 x_i で表現される
- 係数β_i は偏回帰変数とよばれる。



- ここに、 $\varepsilon_i \sim N(0, \sigma^2)$

利用用途

- 体重は、身長、胸囲、腹囲とどのような関係にあるのか？　例えば、次の関係がわかったとき
  - 体重　＝　0.2×身長＋0.3 ×胸囲＋0.55 ×腹囲－55.2　　←重回帰モデル
  - これは、腹囲が最も体重に影響を与えていると言える。
  - 各目的変数の単位が揃っていないと意味が不明となる（例：mとcm, kgとg）
  - そこで，各変数を平均0、分散1に標準化することも検討に入れる

偏回帰係数の同時検定のためのF検定

偏回帰係数{β_i} (i=0 – p)を同時に検定したい―＞次の仮説を考える

- H0: ß_0 = ß_1 = …. =ß p = 0 （全てがゼロ，求めた重回帰分析に意味が無い）
- H1: {ß_i} (i=0 – p)の少なくとも一つがゼロでない。

F検定

- 右のF値を計算する。第2式を用いることが多い。
- このF値はF(p, N-p-1)分布に従う
- F > F(p, N-p-1)ならば，H0（帰無仮説）を棄却
  すなわち，重回帰分析に意味がある
- statsmodelsでは，F値は自動的に計算され，
  このときの確率Prob(F-statistic)が計算される
- これを見て棄却するか否かを判定すればよい。

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

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

F値の解説

- 第1式を見る。分母 > 0に留意
- 分子を見て，全変動　＞　残差変動より，分子＞0
- よって，F＞0である。
- 次の事実がいえる。
  H0が正しいとき（全変動－残差変動）－＞0に近づく
  H0が棄却のとき（全変動－残差変動）は大きくなる
- よって，F値が十分に大きな値であれば，H0を棄却してH1を採用するのが自然と考える。

例題：2つの説明変数からなる重回帰モデル

次の2つの説明変数からなる重回帰モデルを考える

- $y = \beta\_1 \times x1 + \beta\_2 \times x2$

REG_MultipleRegressoin

データの生成

- yはx3と関係しないが，このx3をyの説明変数として採用した，という誤りがあったとする。

```
df = pd.DataFrame({'y':y, 'x1':x1, 'x2':x2, 'x3':x3})
results = smf.ols('y ~ x1 + x2 + x3 -1', data=df).fit()
results.summary()
```

- x3の生成はかなりトリッキーであるが，あくまで練習用として見てほしい。
- np.random.normal()は呼び出しごとに，異なる値を生成することに注意されたい。
- yは観測雑音として正規分布に従う雑音が重畳していると仮定する。
- 上記の `'y ~ x1 + x2 + x3 -1'` が，$y = \beta_0+\beta_1 \times x1+\beta_2 \times x2+\beta_3 \times x3$を仮定している。
- 結果を次頁に示す

多重共線性(multicollinearity)の影響を見る

```
num = 30
rad = np.linspace(-np.pi, np.pi, num)
x1 = np.sin(rad)
x2 = np.random.normal(-2.0, 3.0, num)

b1, b2 = 3.3, -1.25
noise = 0.001*np.random.normal( 0.0, 1.0, num)
y = b1*x1 + b2*x2 + noise
x3 = 3.35*np.sin((rad+0.001))+ 0.001*np.random.normal( 0.0, 1.0, num)
df = pd.DataFrame({'y':y, 'x1':x1, 'x2':x2, 'x3':x3})
results = smf.ols('y ~ x1 + x2 + x3 -1', data=df).fit()
results.summary()
```

例題：2つの説明変数からなる重回帰モデル

右の結果の考察

- 係数を見ると，x1の値が少しずれている
- x3のp値を見ると，x3の値が0である可能性が少しあり，x3でyを説明できないかもしれないかもしれない。
- R-squaredの値は1でほぼフィッティングされている。
- F値に対する確率Prob(F-statistic)は十分に小さいので良いモデルと示している。
- AICの値は後に比較する

OLS Regression Results

Dep. Variable:	y	R-squared:	1.000
Model:	OLS	Adj. R-squared:	1.000
Method:	Least Squares	F-statistic:	1.948e+08
Date:	Fri, 15 Mar 2019	Prob (F-statistic):	4.00e-99
Time:	07:06:54	Log-Likelihood:	160.31
No. Observations:	30	AIC:	-314.6
Df Residuals:	27	BIC:	-310.4
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	3.3508	0.311	10.771	0.000	2.712	3.989
x2	-1.2500	5.9e-05	-2.12e+04	0.000	-1.250	-1.250
x3	-0.0152	0.093	-0.163	0.872	-0.206	0.175

Omnibus:	0.152	Durbin-Watson:	2.376
Prob(Omnibus):	0.927	Jarque-Bera (JB):	0.362
Skew:	0.076	Prob(JB):	0.834
Kurtosis:	2.483	Cond. No.	5.62e+03

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 5.62e+03. This might indicate that there are strong multicollinearity or other numerical problems.

例題：2つの説明変数からなる重回帰モデル

説明変数を正しく2つにした結果

```
results = smf.ols('y ~ x1 + x2 -1',
data=df).fit()
results.summary()
```

- x1の係数は真値に近づいている
- F値とAICの値は若干良くなっている

考察

- x1とx3は振幅と位相が若干異なる
- しかし，同じ周期性を持つ⇒高い相関性
- これを統計分野では**多重共線性（multicollinearity）**という。数値計算上，連立方程式を解いているため，従属性の高い方程式が2本以上あると，その数値解は不安定となり，結果の信頼度が低下する。
- そのため，相関係数を見て，相関度が高ければ
  どちらかが従属変数を見て，省くということも
  考えられる。
  このことを行うには，単にデータの数値を
  見るのではなく，その物理的背景を知った上で
  省くのか，採用するのかを決めることとなる。

OLS Regression Results

Dep. Variable:	y	R-squared:	1.000
Model:	OLS	Adj. R-squared:	1.000
Method:	Least Squares	F-statistic:	3.027e+08
Date:	Fri, 15 Mar 2019	Prob (F-statistic):	2.05e-103
Time:	07:06:54	Log-Likelihood:	160.30
No. Observations:	30	AIC:	-316.6
Df Residuals:	28	BIC:	-313.8
Df Model:	2		
Covariance Type:	nonrobust		

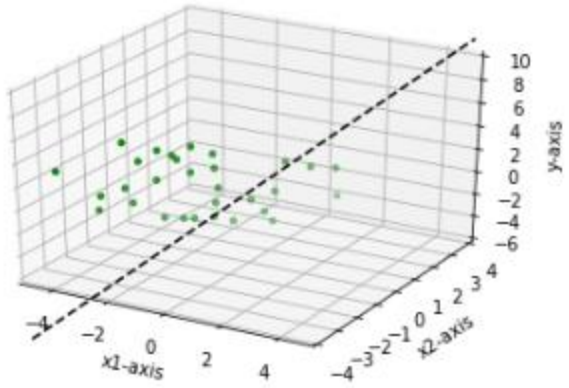
	coef	std err	t	P>|t|	[0.025	0.975]
x1	3.3000	0.000	1.05e+04	0.000	3.299	3.301
x2	-1.2500	5.68e-05	-2.2e+04	0.000	-1.250	-1.250

Omnibus:	0.219	Durbin-Watson:	2.382
Prob(Omnibus):	0.896	Jarque-Bera (JB):	0.420
Skew:	0.076	Prob(JB):	0.811
Kurtosis:	2.440	Cond. No.	5.54

例題：2つの説明変数からなる重回帰モデル

3次元プロットを見る

- 視覚的にもおよそフィッティングしているように見える



相互作用モデル

- 説明変数の乗算項があるものをいう
- 例えば
  - y = ß_0 +ß_1 x_1 + ß_2 x_2 + ß_3 x_1 x_2
- この表現を ols()で実現するには，引数を次のように指定すればよい。
  - ‘y ~ x1 + x2+ x1:x2
- 重回帰分析の初めで，テストデータを作成したように，この相互作用モデルのデータを各自で作成して，確かめてみよ。

ワインの品質分析

背景

- ワイン好きの経済学者オーリー・アッシェンフェルター（Orley Ashenfelter, プリンストン大学）がワインの将来の価値を予測できるか、という命題で、ワインの質の定量的評価式を提案し、物議を醸した。（出典：Ian Ayres,その数学が戦略を決める，文春文庫,(2010)）
- 現在から見れば，少なからず論の欠点はあるが，問題を提起したことに意味があると言える。
- Orley Ashenfelter, see Wikipedia https://en.wikipedia.org/wiki/Orley_Ashenfelter
- この定式化やデータは用いないが、ワインの品質評価を定式化できるか否かを，Ashnfelterが用いたのと異なる次のデータから見出すことを試みる。

データの内容

- P. Cortezらが調査したポルトガルワインの成分を基に、赤ワインと白ワインの品質を検証したデータを用いる。
- ワインごとに測定された11種類の成分データとそのワインの味を評価したグレード（数値）からなっている。グレードは3人以上のワイン査定士が評価した結果の中間値である（グレードは0（とてもまずい）から10（絶品）まで）。これは次から取得できる。
- 赤ワインを対象とする。白ワインの分析は各自に任せる。

ワインの品質分析

データの取得

- UCI Wine Quality Data Set http://archive.ics.uci.edu/ml/datasets/Wine+Quality
- -> Data Folder -> winequality-red.csvをダウンロード
- セパレータ(数字区切り)がセミコロン(;)であるため、".csv"であってもセル毎に分離されずに表示されるが、これは、pandasで対処する。データサイズはN=1599である。
- データのラベル名に，空白があり，これも含めて読めるはずであるが，実際には読めなかった。そのため，CSVファイルをエディタで開き，空白をアンダーライン(_)に変換した。このファイルをwinequality-red_mode.csvとする。

データの読込み

REG_Multi_WineQuality

```
url='https://sites.google.com/site/datasciencehiro/datasets/winequality-red_mod.csv'
wine_set = pd.read_csv(url, sep=";")
wine_set.head()
```

ワインの品質分析

データの説明

fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
酒石酸濃度	酢酸酸度	クエン酸濃度	残留糖分濃度	塩化ナトリウム濃度	遊離亜硫酸濃度	亜硫酸濃度	密度	pH	硫酸塩濃度	アルコール度数(%)	0(very bad)-10(excellent)の品質スコア

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
5	7.4	0.66	0.00	1.8	0.075	13.0	40.0	0.9978	3.51	0.56	9.4	5

目的変数と説明変数の構成

- 意味は特にないが，次のように置いてみる。

```
ols_model = "quality ~  volatile_acidity + chlorides + total_sulfur_dioxide +
sulphates + alcohol"
results = smf.ols(formula = ols_model, data=wine_set).fit()
results.summary()
```

演習：ワインの品質分析

初級編

- 1599のデータセットに対して，幾つかのformulaを考えて，それに対する重回帰分析を行い，その結果について考察せよ。
- 特に，どの要素がqualityに大きく影響するかを考察すること。

上級編

- 全部で1599セットある。
- 例えば，分析用のトレーニングデータを最初の1000セット，テストデータを残りの599セットに分ける。
- 説明変数はより多くの要素を組み入れること。
- トレーニングデータを用いて重回帰分析を行い，偏回帰係数を求める。
- 各種結果に対する考察を行うこと。
- この偏回帰係数を用いて重回帰モデルを作成し，この説明変数にテストデータのそれを代入して，目的変数がテストデータのqualityとどれだけ予測できているかを調べ，これを表で表し，重回帰分析の有効性について考察すること

1．トレーニングデータを用いる 偏回帰係数を求めて重回帰モデルを定める
2．この重回帰モデルの有効性を調べるため、テストデータ（要素のみ）を本モデルに与え、モデルの計算値（qualityの推定）とテストデータにある目的変数（真のquality）の差をとり検討する。例えば、この差の二乗の総和の平均を取るなどして考察する。

余話　ワインの味は予測できるか

- 次の本での論争　→　イアン・エアーズ：その数学が戦略を決める，文藝春秋，2010（原本：Ian Ayres: Super Crunchers: Why Thinking-By-Numbers is the New Way To Be Smart, Bantam），データサイエンスの先駆けともいえる書籍。
- この中で，Prof. Orley Ashenfelter（プリンストン大学）がワインの品質に関する回帰分析を示していると述べている。しかし，これは，原文において著者のミスであり，実際は，価格に関する回帰分析であると筆者自身も認めている，と書かれている。このことは，上記の改定を経た文庫本で訳者が章末に記している。また，このことを裏付ける AshenfelterのHPを次に示す。
- http://www.liquidasset.com/winedata.html
- http://www.liquidasset.com/orley.htm
  - 次のサイトはその指摘がある（真偽は不明）　http://cruel.hatenablog.com/entry/20150121/1421802947
  - Ashenfelter原論文を掲載している（真偽は不明）　http://www.liquidasset.com/orley.htm
  - 原著論文で問題になっているTable 2 https://www.researchgate.net/publication/5091269_Bordeaux_Wine_Vintage_Quality_and_Weather
  - Orley Ashenfelter https://en.wikipedia.org/wiki/Orley_Ashenfelter

考え方

- ワイン専門家からアッシェンフェルターのワイン方程式は当初，嘲笑されたというが，旧来の観念に縛られるのではなく，なんでも分析してみよう，という考え方は学ぶに値する
- 関連サイト
  - 下記は，統計学の観点からワインやワイン産業の分析を行った論文が報告されている。ワインに関する各種データもある。
  - American Association of Wine Economics http://www.wine-economics.org/
  - European Association of Wine Economics http://www.euawe.org/

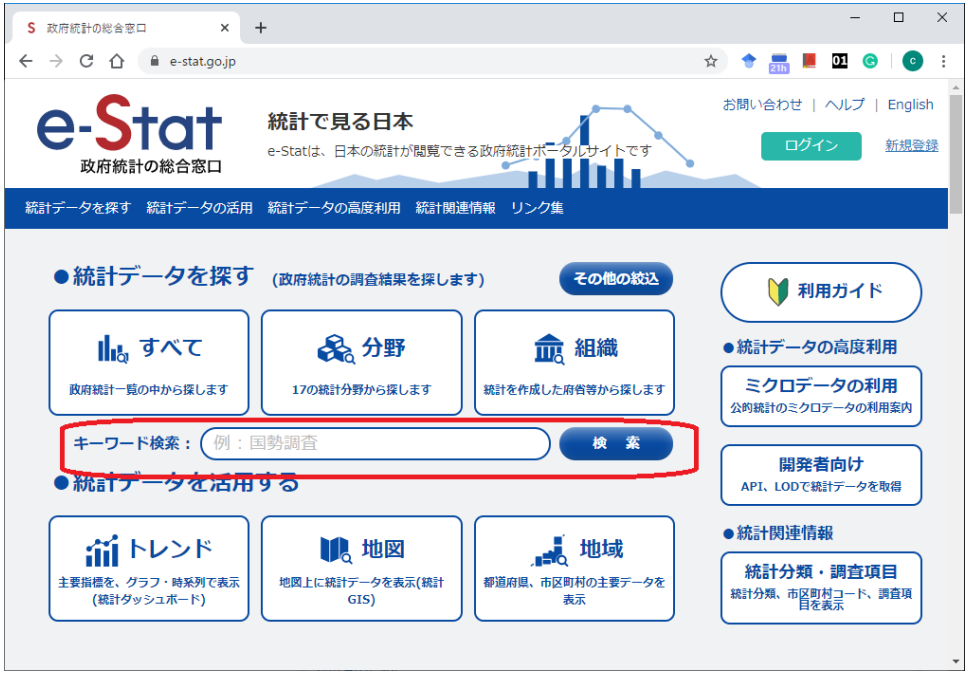
演習　保育所入所待機児童数

問題

各都道府県の**保育所入所待機児童数**が社会・人口に関するどのような要因に影響されるか否かを調べる

データの取得

- e-stat https://www.e-stat.go.jp/を開く
- 赤枠の「キーワード検索」の欄で
  “保育所入所待機児童数”を検索
- 検索結果のうち，なるべく新しい年度の“EXCEL”をダウンロード
  - ダウンロードできない人のために、「社会生活統計指標－都道府県の指標－2017 / 基礎データ」,2017年のデータである”kiso-j.xls”をValuable_Kitフォルダに入れておく。
- このEXCELファイルには、タブ“J1”とタブ“J2”がある



特に、根拠があるわけではありませんが、右のように置く。

目的変数
タブ“J2”:「保育所入所待機児童数」

説明変数
タブ“J2”:「地域子育て支援拠点事業実施箇所数」
タブ“J1”:「介護老人福祉施設数」
タブ“J1”:「児 童 福 祉 施 設 数」

演習　保育所入所待機児童数

データの作成

- 日本語の問題
  - このデータの日本語は，ローマ字または英語に変換した方が扱いやすい
  - 日本語データを扱うには（お勧めしません）
    - pandasの扱い df= pd.read_csv(“filename.csv",encoding="SHIFT-JIS")
    - matplotlibの扱い　日本語フォントをmatplotlibに与える操作が必要（説明省略）
- 都道府県は，ファイルのローマ字を用いる
- 都道府県コード（最後の行）をIDとして用い，この数字を用いると良い
- 年度は最新のものだけを用いるものとする
- 1行目にラベル名として次を与える：　‘ID’, ‘Prefecture’, ‘Y’（目的変数），‘X1’（説明変数1番目），‘X2’（説明変数2番目），‘X3’（説明変数3番目）とする（名称は，自由に与えられます）。

演習　保育所入所待機児童数

重回帰分析の実行

- 上記のデータをCSV形式で保存（“filename.csv”としたとする），次と同じフォルダに入れる
- REG_Multi_WineQuality.ipynbを適当な名前でコピー（”aaa.ipynb”とする）
- 例えば，この中の

```
url='https://sites.google.com/site/datasciencehiro/datasets/winequality-red_mod.csv'
wine_set = pd.read_csv(url, sep=";")
ols_model = "quality ~  volatile_acidity + chlorides + total_sulfur_dioxide + sulphates + alcohol"
```

を次に変更（’#’はコメント）

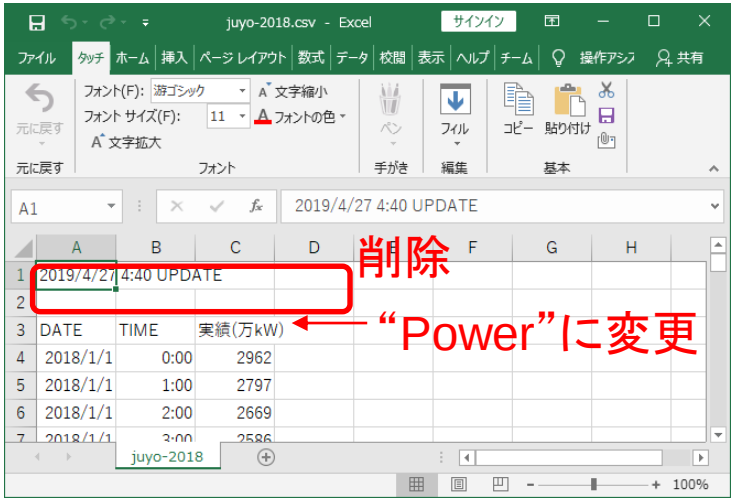
```
#url='https://sites.google.com/site/datasciencehiro/datasets/winequality-red_mod.csv'
wine_set = pd.read_csv(‘filename.csv’)
ols_model = “Y ~  X1 + X2 + X3"
```

演習　電力消費量と気温

電力消費量は気温と関係性があるか？　を重回帰分析で調べる

データの取得

- 地点は**東京**とする
- 年度(2016, 2017, 2018, ・・・・のいずれか)を決めて



電力消費量(東京電力)過去の使用量

- http://www.tepco.co.jp/forecast/index-j.html
- 「過去の電力使用実績データ」から年度を定める
- マウス右クリックで，CSVファイルをダウンロード
- 右図のように編集
- 24hour → 1日の最大値を，その日の代表値とする
- 1日のデータ24点を1点に集約，これをダウンサンプリングという
- ファイル名を”power_temp.csv”とする

気温(気象庁)

- http://www.data.jma.go.jp/gmd/risk/obsdl/　→「東京」を選択
- 「項目を選ぶ」を選ぶ
  - 「データの種類」 →“日別値”，「気温」→“日最高気温”と“日最低気温”をチェック
- 「期間を選ぶ」を選ぶ
  - 各自が定めた西暦年の1月1日から12月31日を指定(電力データがこの期間だから)
- 「CSVファイルをダウンロード」（注意:このファイルのエンコードはSHIFT-JISである。
- 1日単位のデータである。

データファイル(CSV)の結合

- power_temp.csvファイルに日に合わせて気温のデータを結合する。

”REG_Multi_PowerTemp.ipynb”の説明

演習　電力消費量と気温

電力消費量は気温と関係性があるか？　を重回帰分析で調べる

仮説

- 電力消費量は，その日の最大値を用いる
- 気温は，その日の最高気温と最低気温を用いる
- モデル1：　電力消費量　＝　a ×　最高気温　＋　const（定数）
- モデル2：　電力消費量　＝　a ×　最高気温　＋　b×最低気温＋const（定数）
- どちらのモデルが有効か？　それとも，両方とも有効でないか？

次の評価項目を基に考察

- 決定係数（R-squared）
- F検定（F-stat, p-value）
- t検定（t-stat, p-value）
- 係数，a, b, const

予測する考え方は？

- 各自で原理を考えてみよう
- 注意：http://www.statsmodels.org/dev/sandbox.html　や https://github.com/statsmodels/statsmodels/blob/master/statsmodels/sandbox/regression/predstd.py　までは用いなくてよい。

余話　検定は万能か？

- 赤池弘次：情報量基準 AIC とは何か－その意味と将来への展望，数理科学 14(3), pp.5-11, 1976
  - あるサイコロの正しさを検定するという問題も全く同様で，現実のサイコロで完全に対称なものが存在しえないことは明らかである。このように仮説(帰無仮説）は常に否定される立場にあり，データによる検定結果を待つまでもなく結論は見えている。

  ⇒「有意差無し」は「サンプル数（または反復数）が少なすぎた」という「実験の不備」を示しており，この場合の検定に意味があるのか？

- Clive W.J. Granger (原著), 細谷 雄三 (翻訳)：経済モデルは何の役に立つのか―経済経験モデルの特定化とその評価，牧野書店
  - 非常に大規模なデータセットに対しては，正確に規定されたあらゆる帰無仮説は，事実上その仮説が厳密に正しい場合を除いて，標準的な有意水準では実質的に棄却されてしまうであろう。結果を解釈するとき，われわれは統計的有意性よりも経済的有意性をもっと強調するようになるであろう。

  ⇒　データが多いと，意味あるモデルを見出せない。この経済的有意性は物理的背景に基づき考えられる物理モデル，と読み替えても同じことが言える。

- 検定だけでなく，あらゆる特徴を多角的に見よう！

一般化線形モデル
(Generalized linear model, GLM)

1. ポアソン回帰モデル
2. ロジスティック回帰モデル

本講義では範囲外とします。

GLMの定式化

構成

- 線形予測子（Linear predictor）
  - 説明変数 x_i から成る回帰モデルである。攪乱項がなく，また，z_iは目的変数でなく，単に式変形に用いる媒介するための変数（媒介変数）であることに注意されたい。

$$z_i = \beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i} + \cdots + \beta_p x_{p,i}$$

- リンク関数（Link function）
  - あるパラメータ(parameter)が説明変数と非線形の関係にあるが，関数L[]で変換すると線形予測子（線形である回帰モデル）に等しくなるような関数をリンク関数と呼ぶ。

$$L[\text{parameter}] = z_i = \beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i} + \cdots + \beta_p x_{p,i}$$

- 何を予測するか
  - ポアソン回帰モデルでは，ポアソン分布のλ
  - ロジスティック回帰モデルでは，確率P
  - 両者とも、離散確率分布で表される
  - ここで説明するGLMは、離散確率分布のパラメータ（ポアソンのλ、ロジスティックのP）を求めることにある。

ポアソン回帰モデル

1. ポアソン分布とは
2. 定式化
3. 例

ポアソン分布に従うデータ

ポアソン分布の性質

- 客がてんでばらばらに店に到着する　⇒　定常性，独立性，希少性が成り立つと考える

$$P(X = y_i) = \frac{\lambda^{y_i} \exp(-\lambda)}{y_i!}$$

- Xは自然数である。
- ある単位時間（一定時間）ごとに客数をカウントし，その数を y_i とおく
- 十分長い時間を過去にとって，単位時間に到着する客数の平均値をλとおく
- 例えば，λ＝６人とわかっているときに，ある時間に8人の客（y_i＝8）が到着する確率は次となる

$$P(8) = \frac{6^8 e^{-6}}{8!} \simeq 0.103$$

- これをポアソン到着という

ポアソン到着する客の様子は

- ポアソン到着のとき，客の到着時間の間隔Tは指数分布に従うことが知られている。

$$T = -\frac{1}{\lambda} \log_e uniform(0,1)$$

uniform(0,1)は一様乱数(0,1）を示す

- このシミュレーション例を右図に示す。
- 横軸は3時間ごとに目盛りを与えている
- 初めの3時間ごとに，到着数は，2, 1, 1, 1, 0, 1, 0, 3, 2, 0人
- この状況を統計分野では「カウントデータ」と言うことがある。
- カウントデータならば，直ちにポアソン分布とは言えない。
- （規則正しくカウントアップされることもあるため）

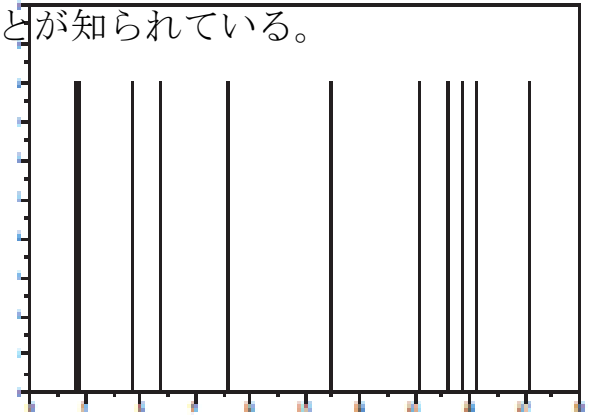


図 6.21　T の発生分布

リンク関数

$$\lambda_i = \exp\left(\beta_0 + \beta_1 x_{1,i}\right)$$

$$\log_e\left(\lambda_i\right) = \beta_0 + \beta_1 x_1$$

線形の関係を求めることができる

ポアソン回帰分析（β1=0の場合）

λを予測するのが目的
次に指数法則を適用し，
βを求める

何を予測？

- 右のようにおけるとする。 $\lambda = \exp(z) = \exp(\beta_0 + \beta_1 x_1)$

$a^m a^n = a^{m+n}$

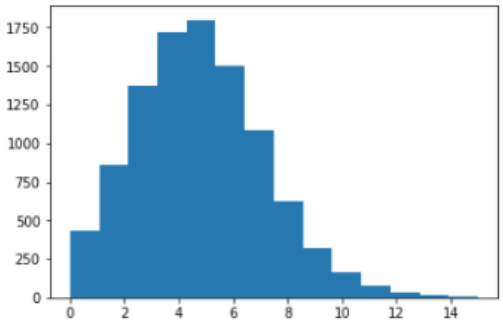
- ここで，ポアソン分布には，右の関係がある。 $E[y] = \lambda$
- ここで，β_1 = 0 ならば，λは一定値となる。 wiki https://en.wikipedia.org/wiki/Poisson_distribution
- λ一定であるから確率論でいう定常性（Stationarity）があり，期待値も一定値となる。
- 実際に，平均値がλになるのかを確認する。

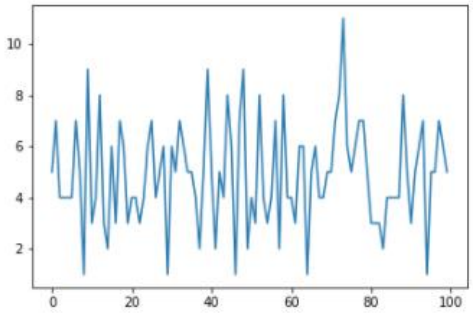
GLM_Poisson

```
lam = 5
y = np.random.poisson(lam,10000)
count, bins, ignored = plt.hist(y, 14, normed=False)
num = 100
plt.plot(range(num), y[0:num])
```

データを見る

- λ=5とした分布は，ポアソン分布を近似している。
- 観測値は右図のようになった。





ポアソン回帰分析（ β1=0 の場合）

結果の考察

- モデルは‘y ~ x’ とおいて， y = β_0 + β_1 x を仮定した。
- ポアソン回帰分析はfamily=sm.families.Poisson(link=sm.families.links.log)を指定する。
- 結果で係数を見ると，β_1 はほぼ 0 であり，切片β_0のみ値が生じ，予想通りとなった。
- この分析では，λ=exp(β_0)であるから，これを計算すると，ほぼ5であり，真値に近い値を得た。
- E[y] = λとなるかを確かめるために，y_i の平均値（df.y.mean()）を計算すると，近い値を得て，理論と本シミュレーションの適合性が言える。

```
x = range(len(y))
df = pd.DataFrame({'x':x, 'y':y})
glm_model = 'y ~ x'
result = smf.glm(formula=glm_model, data=df,
family=sm.families.Poisson(link=sm.families.links.log)).fit()
print(result.summary())
b0, b1 = result.params
print(np.exp(b0))
df.y.mean()
```

```
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:                      y   No. Observations:                10000
Model:                            GLM   Df Residuals:                    9998
Model Family:                 Poisson   Df Model:                           1
Link Function:                    log   Scale:                          1.0
Method:                          IRLS   Log-Likelihood:                -21961.
Date:                Fri, 12 Jan 2018   Deviance:                       10408.
Time:                        07:56:02   Pearson chi2:                 9.82e+03
No. Iterations:                     4
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      1.6068      0.009    179.188      0.000       1.589       1.624
x          -1.069e-06   1.56e-06     -0.687      0.492   -4.12e-06    1.98e-06
==============================================================================
```

```
b0, b1 = result.params
print(np.exp(b0))
```

```
4.98694808914
```

```
df.y.mean()
```

```
4.9604
```

ポアソン回帰分析（ β1≠0 ）

何を予測？

- 右のようにおけるとする。$\lambda_i = \exp\left(\beta_0 + \beta_1 x_{1,i}\right)$
- λ_iはx_1, iの関数となり一定値ではない。よって，確率論でいう非定常性（non Stationarity）といえる。
- 下の関係について，期待値や分散は1つの標本に対する演算ではなく，iを固定したときに得られる集合に対する期待値と分散である。

$$E\left[y_i\right] = V\left[y_i\right] = \lambda_i$$

- このポアソン分布は非定常であり，E[y_i]も一定とならない。
- このような非定常過程を扱いたい。
- 次のようなプログラムで発生したデータを用いる

GLM_Poisson

```
b0 , b1 = 0.5, 3.5
for i in range(num):
    x[i] = i
    lam = np.exp( b0 + (b1/float(num)) * (float(i)))
    y[i] = np.random.poisson(lam,1)
```

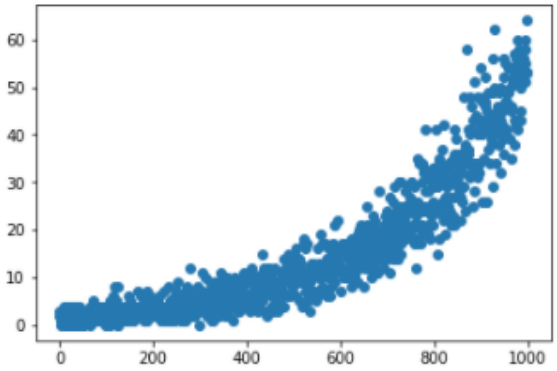
説明

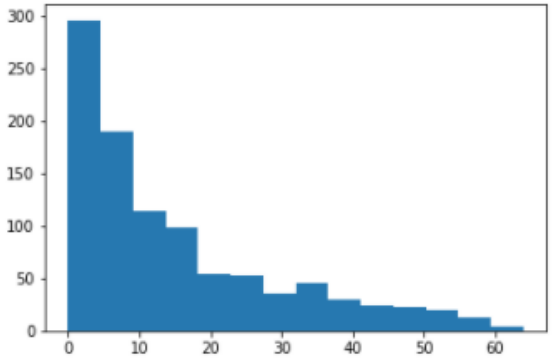
- x[i]は，説明変数であり，順序数（0, 1, 2 , ..）を入れる。
- lam（＝λ）は，初期値が b0 （β_0), 最終値がb1（β_1） となるようにした。ただし，b1を大きくとれないことと，データ数numを大きくとりたかったため，上記のスクリプトでは，(b1/float(num)) が見かけ上の係数となる。
- y[i]には，このlamに基づくポアソン分布が1つずつ格納される。

ポアソン回帰分析（ β1≠0 ）

グラフ表現

- 頻度と(x,y)散布図を示す。
- 非定常であるから，左の頻度はすでに見慣れたポアソン分布ではない。
- 散布図を見て，平均値が指数的に上昇していることと，上昇に伴い分散（ばらつき）が大きくなっていることがわかる。





左のようなデータが観測され，そのヒストグラム（分布）が右のようならば，ポアソン回帰モデルをもちいたGLMの適用を検討する

ポアソン回帰分析（ β1≠0 ）


```
df = pd.DataFrame({'x':x, 'y':y})
glm_model = 'y ~ x'
result = smf.glm(formula=glm_model, data=df,
family=sm.families.Poisson(link=sm.families.links.log)).fit()
print(result.summary())
b0, b1 = result.params
b1 = b1 * num
print(b0,b1)
e_b0 = np.exp(b0)
e_b1 = np.exp(b1)
print(e_b0, e_b1)
```

(b1/float(num)) が見かけ上の係数となるので，この分母を払っている

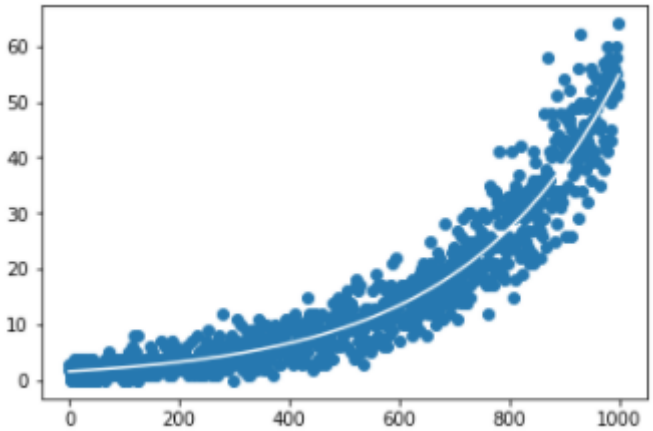
Generalized Linear Model Regression Results

Dep. Variable:	y	No. Observations:	1000
Model:	GLM	Df Residuals:	998
Model Family:	Poisson	Df Model:	1
Link Function:	log	Scale:	1.0
Method:	IRLS	Log-Likelihood:	-2530.3
Date:	Fri, 12 Jan 2018	Deviance:	1048.0
Time:	07:44:35	Pearson chi2:	1.02e+03
No. Iterations:	5		

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	0.4883	0.028	17.139	0.000	0.432	0.544
x	0.0035	3.66e-05	96.134	0.000	0.003	0.004

結果

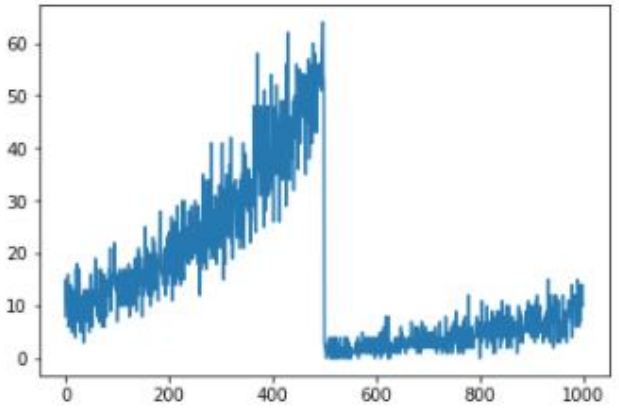
- b0 = 0.488270 b1 = 3.519202 ←近い値を示している
  （注意：シミュレーション毎に値は変わる）
- exp(b0) = 1.629494 exp(b1) = 33.757466
- この値を用いたグラフを右に示す。
- フィッティングカーブは E[y_i]を表す。
- すなわち，i番目における集合平均であると言える。



ポアソン回帰分析（ β1≠0 ）

考察

- 散布図を見ると，指数的曲線が視覚として見えるから，このようなカーブを示すデータのみにポアソン回帰分布を用いると考えることは勘違いである。
- データは時系列（順序が重要）でなく，数値解法（最尤法を大抵使っている）もデータの順序に依存しない。
- 各データ（x_i, y_i）がポアソン分布に従っていることが要求されているだけである。
- このことをシミュレーションで検証するために，num個のデータ(x_i, y_i)の前半と後半をひっくり返したデータを作成した。yだけをプロットしたのが右図である。x_iも用いて散布図にすると，初めの散布図と全く同じになるので，このようにした（プログラムを見て下さい）。
- このデータに対して，同じポアソン回帰分析を行ったところ，全く同じ結果を得た。データをシャフルしても同じ結果を得るであろう。
- すなわち，データの順序は関係しないと言える。
- これらのことから，次のことが指摘できる
  - 見た目で平均値が指数的に上昇するデータにはポアソン回帰分析
  - の適用を試みる（他の回帰分析や検定も試みる）
  - 見た目が指数的でない場合でも，ポアソン到着と思う場合には
  - 適用を試みる
  - ポアソン到着のことを統計分野では，カウントデータ（count data）と称することもあるが，これまで述べてきたように，データはポアソン分布の性質を有している場合にポアソン回帰分析は有効
  - ポアソン分布と見なした適用例：交通事故発生件数，1日に受け取る電子メール数，単位時間あたりに店やATM などに訪れる客の数，などがある。



- 今回，順序数で表した説明変数x_iは一つのみであったが，
  目的変数は　β_2 x2 + β_3 x3 + ・・・　と拡張してもかまわない。

ロジスティック回帰モデル

1. ロジスティック回帰
2. 定式化
3. 例

ロジスティック回帰

対象データ

- 目的変数が0, 1という二値の場合のデータによく用いられている。例えば，次のようなものがあげられる。
  - ある植物種子に肥料をどれだけ投与（説明変数）したかで種子が発芽（y_i=1)したか否か（y_i=0）を示すデータ
  - 被験者iの疾病の発生のあり（y_i=1），無し(y_i=0）と，血圧や体重（説明変数）との関連性を示すデータ
- このため，目的変数y_iはベルヌーイ分布に従うとする。

$$y_i \sim P_i^{y_i}\left(1-P_i\right)^{1-y_i} \quad , \quad y_i \in \{0,1\}$$

  - ベルヌーイ分布は，全体の人数（またはモノ）が一人の二項分布である。今回のように，一人一人別々の分布に従うときにはベルヌーイ分布を用いる。

定式化

- 上記の確率P_iが，0,1 の値を取る。これと説明変数との関係モデルを見出したい。
- このモデルが不連続では数学的に不都合であるから，連続関数で表したい。このため，ロジスティック関数を導入する。

$$P_i = \frac{1}{1+\exp\left(-z_i\right)}$$

- ここに，z_iは線形予測子であり，話の見通しを良くするために，z_i = ß_0 + ß_1 x_1,iとおく。もちろん，目的変数がx_{1, i} ～x_{p, i}となっても議論は変わらない。

リンク関数

- 観測できるのは，0 か 1 であった。

$$P_i = \frac{1}{1+\exp\left(-z_i\right)}$$

- リンク関数を次で与える

$$L[P_i] = \log\left(\frac{P_i}{1-P_i}\right) = \beta_0 + \beta_1 x_i$$

この形のリンク関数を
ロジット関数（logit funtcion）という。

  - 右辺が線形結合であることが重要
  - この関数を用いることで，β0とβ1を求めることができる。

- 備考
  - 土居先生を参照　http://www012.upp.so-net.ne.jp/doi/biostat/CT39/glm.pdf

例：カブトムシ問題

薬品の投薬量とカブトムシの生存率

- Annette J. Dobson and Adrian G. Barnett, An Introduction to Generalized Linear Models, 3rd ed. , CRCPress 2008
- x_i：ある薬品の投薬量
- n_i：薬品を与えたカブトムシの数
- y_i：そのうち死んだ数

Table 7.2 *Beetle mortality data.*

Dose, x_i ($\log_{10}CS_2mgl^{-1}$)	Number of beetles, n_i	Number killed, y_i
1.6907	59	6
1.7242	60	13
1.7552	62	18
1.7842	56	28
1.8113	63	52
1.8369	59	53
1.8610	62	61
1.8839	60	60

準備

- 生き残った（0）か死滅したか（1）ゆえ，ロジスティック回帰モデルを適用する。
- データを見ると，このような0/1表現ではないが，次のように考える
  - 生存：n－y
  - 死滅：y

例：カブトムシ問題

準備

- logit (P_i) = β_0 + β_1 x_i
- 確率はほぼ P = y / n と予想される
- この例のように，死亡(1）と生存（n_i – y_i）で表されるような場合には，下記のように‘y + I (N-y)’と記述する
- ここに，I()はカッコ内の記号を算術演算であると宣言するためのものである。もし，これが無いと ‘N – y‘ の’ －’はPatsy の規則によりyを引くのでなく除去することになる。

GLM_Logistic_Beetle

```
df = pd.DataFrame({'x':[1.6907, 1.7242, 1.7552, 1.7842, 1.8113, 1.8369, 1.861, 1.8839],
                   'n':[59, 60, 62, 56, 63, 59, 62, 60],
                   'y':[ 6, 13, 18, 28, 52, 53, 61, 60]})
glm_model = 'y + I(n-y) ~ x'
fit = smf.glm(formula=glm_model, data=df, family=sm.families.Binomial(link=sm.families.links.logit))
print(result.summary())
```

例：カブトムシ問題

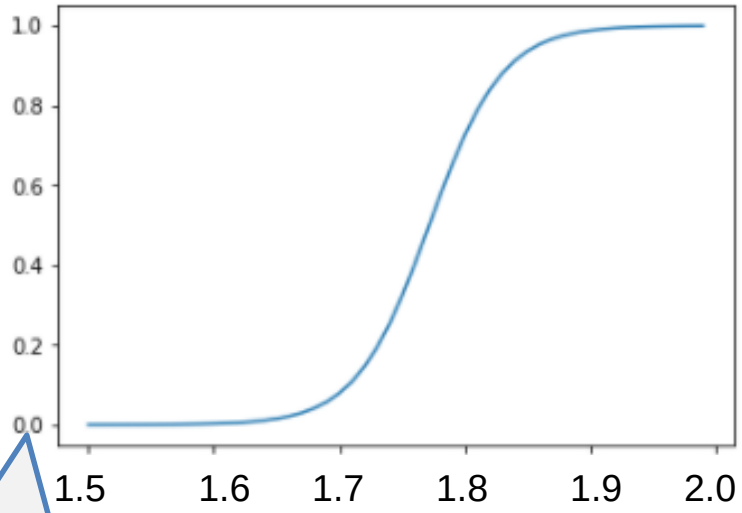
結果

- 係数が0となる確率は0である。
- P_i = 1/ (1 + exp(－(β_0 + β_1 x_i)) のグラフを右に
- 縦軸が確率p_i，横軸が投薬量である。
- x の範囲（1.6907 ≦x≦ 1.8839 ）では，ほぼ p_i = y_i / N_i である

Generalized Linear Model Regression Results

Dep. Variable:	['y', 'I(N - y)']	No. Observations:	8
Model:	GLM	Df Residuals:	6
Model Family:	Binomial	Df Model:	1
Link Function:	logit	Scale:	1.0
Method:	IRLS	Log-Likelihood:	-18.715
Date:	Wed, 10 Jan 2018	Deviance:	11.232
Time:	21:24:17	Pearson chi2:	0.169
No. Iterations:	6		

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-60.7175	5.181	-11.720	0.000	-70.871	-50.563
x	34.2703	2.912	11.768	0.000	28.563	39.978



縦軸は確率 P

なぜ，観測量を目的変数においていいのか？

- ロジスティック回帰では，ベルヌーイ分布の確率を推定するのが目的なのに，なぜ，観測量を目的変数においていいのか？
- ざっくりと

$$y_i \sim Be(p_i)$$

$$f(y_i \mid p_i) = p_i^{y_i}\left(1 - p_i^{y_i}\right)^{1-p_i^{y_i}} \qquad (i = 1, \cdots, n)$$

尤度関数

$$L\left(\beta_0, \beta_1 \mid y_1, \cdots, y_n\right) = \prod_{i=1}^{n} f(y_i \mid p_i)$$

この尤度関数に，観測量を代入することになるため。

成績アップの要因

背景

- Lee C. Spector and Michael Mazzeo, “Probit Analysis and Economic Education”, Journal of Economic Education, Vol. 11, Issue 2, pp. 37-44, 1980
- この中で，教育プログラムであるPSI (personalized system of instruction) が成績向上に有効であるかの検証を行ったデータがある。このデータは，次のWilliam H. Greene　による著書 Econometric Analysis（http://pages.stern.nyu.edu/~wgreene/Text/econometricanalysis.htm）から取得できるが，StatsModelsのDatasetsにあり，次のようにして取得できる。

GLM_Logistic_PSI

```
data = sm.datasets.spector.load().data
df = pd.DataFrame(data)
df.head()
```

	GPA	TUCE	PSI	GRADE
0	2.66	20.0	0.0	0.0
1	2.89	22.0	0.0	0.0
2	3.28	24.0	0.0	0.0
3	2.92	12.0	0.0	0.0
4	4.00	21.0	0.0	1.0

データの説明

- 説明変数：
  - GPA (Grade Point Average，この場合は前期の成績），
  - TUCE (Test of Understanding in College Economics, この場合は統一テスト結果(Wikipediaより））），
  - PSI (個人教育プログラムに参加(1)か否か（0））,
- 目的変数：
  - GRADE（成績が上がった=1, 否＝0)
- GRADEに寄与するのは，どの説明変数化を調べる

成績アップの要因

結果の考察

```
glm_model = 'GRADE ~ GPA + TUCE + PSI'
fit = smf.glm(formula=glm_model, data=df, family=sm.families.Binomial(link=sm.families.links.logit))
result = fit.fit()
print(result.summary())
```

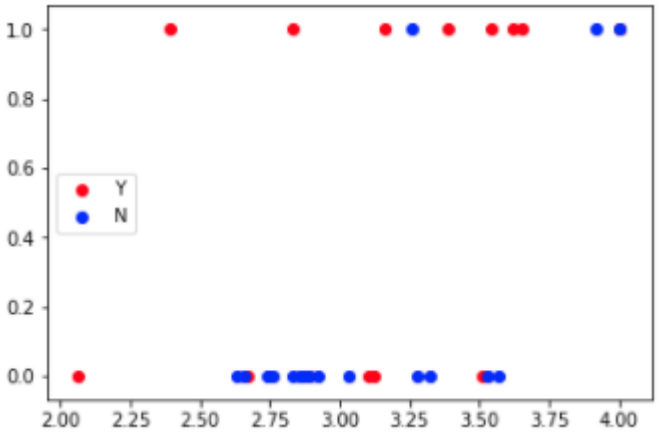
Generalized Linear Model Regression Results

Dep. Variable:	GRADE	No. Observations:	32
Model:	GLM	Df Residuals:	28
Model Family:	Binomial	Df Model:	3
Link Function:	logit	Scale:	1.0
Method:	IRLS	Log-Likelihood:	-12.890
Date:	Fri, 12 Jan 2018	Deviance:	25.779
Time:	13:26:44	Pearson chi2:	27.3
No. Iterations:	5		

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-13.0213	4.931	-2.641	0.008	-22.686	-3.356
GPA	2.8261	1.263	2.238	0.025	0.351	5.301
TUCE	0.0952	0.142	0.672	0.501	-0.182	0.373
PSI	2.3787	1.065	2.234	0.025	0.292	4.465

```
df.corr()
```

	GPA	TUCE	PSI	GRADE
GPA	1.000000	0.386986	0.039683	0.497147
TUCE	0.386986	1.000000	0.112780	0.303055
PSI	0.039683	0.112780	1.000000	0.422760
GRADE	0.497147	0.303055	0.422760	1.000000



- GPAとPSIの係数が高いので，こららの影響が大きく，TUCEの影響は小さいと読み取れる。
- この確認のために相関係数df.corr()を見ると，確かに先の二つとの相関は高そうであるが，いずれも比較的小さい値である。
- 横軸にGPA，縦軸にGRADEのプロット図を示す。ここに，赤印はPSI=1, 青印はPSI=0である。これを見て，PSIが有効か否かは判断できない。