統計学における
相関分析と仮説検定の
基本的な考え方とその実践

自己紹介

Yasuhiro Nakayama

http://id774.net

はじめに

今日は統計学の中でも基礎となる
「相関分析」と「仮説検定」の
基本的な考え方を話します。

はじめに

また Python を利用して実際に

データを分析しその結果を可視化します。

統計学

とりあえず発表の時間は限られているので

高速で統計学やデータ分析の

基礎的な部分を説明します。

KPI とは

KPI（key performance indicator）とは

「目標を達成するために何が必要かを定量的に表す数値」です。

整形して美人になりたいというのは KPI ではありません。

体重を 3 ヶ月後までに 10 キロ減らすとか鼻を 1.5 センチ高くするといったものは KPI です。

データの種類

データにはどのような種類があり

KPI として利用しようとしている指標は

どんなデータなのか正しく理解していないと

しばしば誤った KPI を導き

無意味なデータ分析へとつながります。

変数とは

たとえばアンケートやカルテに、あなたは性別は何ですか、年齢はいくつですか、と質問があったとします。

このとき取る値は大きく

・離散変数（discrete variable）

・連続変数（continuous variable）

に分けられます。

変数とは

このうち離散変数は測定値の境界線を明らかにできるものを言います。

たとえば性別などがそうです。

この離散変数は順序をつけられるかどうかでさらに 2 つに分けられます。

変数の種類

・順序付け不能な離散変数

（例　性別、国籍、所属している会社）

・順序付け可能な離散変数

（例　成績：1．優　2．良　3．可　4．不可）

尺度とは

変数はその性質について次のようにその尺度水準を分類されます。

1）名義尺度

2）順序尺度

3）間隔尺度

4）比例尺度

尺度の種類

1）名義尺度

単なるカテゴリとして与えられ、順序を付けられない変数は名義尺度です。出身地、所属する会社などはこれにあたります。名義尺度は当然のことながら平均を求めたりすることはできません。ただし、最頻値を取ることはできます。

2）順序尺度

順序付け可能な離散変数を評価する尺度です。順位などがこれに相当します。あくまで順序でしかないので、 1 位だから 2 位の人の 2 倍だというような計算をすることはできません。なぜなら 1 位の人は 2 位の人と僅差で 3 位以下の人とは大差、というふうに、目盛間隔が一定では無いからです。平均や分散を求めることはできません。

尺度の種類

3）間隔尺度

ゼロを起点としない連続変数です。たとえば時刻、気温などが相当します。時刻が 20 時だから 10 時の 2 倍だとか、温度が 30 度だから 15 度の 2 倍暑いというような評価はできません。ただし、その間隔自体は意味がある点が順序尺度との違いです。平均や分散などの要約統計量を求めることができますが、比率を求めることはできません。

4）比例尺度

ゼロを起点とする連続変数です。たとえば売上、価格、ユーザー数、ある日からの経過日数、 100 点満点のテストの得点などです。

とにかくデータを集めよう

データの分析に際してはまずデータを集めます。

どんなデータを集めるかはもちろん金融や医療、教育、社会調査など分野ごとにさまざまですが

生のデータは「数字の羅列」です。

生のデータの例

```
95,-10,110
5,-40,108
60,5,100
100,-5,101
33,0,93
5,-10,91
0,0,88
```

元ネタ

http://president.jp/articles/-/12416

朝食や出社時間と営業成績

朝食を食べてきた度合い (朝食率)	出社時間	営業成績
95	-10	110
5	-40	108
60	5	100
100	-5	101
33	0	93
5	-10	91
0	0	88

X, Y と Z に相関はあるか

X	Y	Z
95	-10	110
5	-40	108
60	5	100
100	-5	101
33	0	93
5	-10	91
0	0	88

重要な統計量

- 平均
- 分散
- 標準偏差
- 四分位数
- 中央値

代表値とは

要約統計量とはデータを要約するために統計的操作を加えて求めた数値のことです。

代表値とはその要約統計量でもとくに使われるポピュラーなものです。平均といえば馴染みがあるでしょう。みんなで飲み会をしたときの一人当たりのお金の計算をはじめとして日常的によく利用されます。

1）平均値

2）中央値

3）最頻値

ることに気を付けるべきです。

代表値とは

1）平均値

観測された値をすべて足して個数で除算したもので算術平均とも言われます。すべてのデータから求まるため、全体の変動を表すという利点があります。欠点は外れ値の影響を受けることです。このため上位または下位の数パーセントを除いて平均を求めるといったトリム平均が利用されることもあります。

2）中央値

観測された値を並び替えた時にちょうど真ん中に位置する値です。分布形状が不明である場合や外れ値を多く含むと予想される場合に有効です。

3）最頻値

その名の通りもっとも観測された値です。

いずれにせよ注意すべき点として、あくまで要約であるため何らかの情報が欠落していることに気を付けるべきです。

計算してみよう

さっそくこれらを計算してみます。

分析のためには Python という言語を使います。

なぜ Python なのかという話をする前に統計の世界でよく使われている R 言語の話をします。

R 言語の特長

R 言語はオープンソースソフトウェアであり無料で利用できる統計用言語。

R 言語の登場以前は、学術論文など社会的信頼性を要求される統計データの処理環境といえば高額なプロプライエタリソフトウェアばかりが前提とされた。

「フリーソフトウェアの精神に則り永続的で世界規模な集合知に支えられ、無償でありながら高い信頼に値する。」

このような統計環境というのは、統計家の長期的な生産性に大きく寄与する「持続可能な統計環境」と言える。

R の代わりに Python を使う

Python は科学計算に強い

豊富なライブラリ

- NumPy/SciPy … 多次元配列計算など数値演算
- matplotlib … データ可視化
- scikit-learn … 機械学習
- statsmodels … 計量経済学
- IPython … 対話的なデータ分析実行環境
- pandas … データフレームを扱う

Python は汎用言語

Python はいわゆる一般のプログラミング言語。

インタプリタ言語でありすぐに書き始めることができる。

テキスト処理や Web プログラミングもできる。

一般的なファイル入出力や OS の処理とも親和性が高い。

R はデータフレームという形式でデータを扱うが pandas を使うと Python でもこれと似たことができる。つまり Python は R と同等のことができ一般的な用途にも使える。

2008 年に pandas が誕生。

2010 年代以降、データ分析の主力言語として人気に。

Python は計算が速い

Python 自体はインタプリタ言語だが、内部で

C/C++ や FORTRAN で書かれた古くから使われているライブラリ（*1）が呼ばれている。

このため数値演算の大半を低レベルの最適化されたコードで実行することになるため計算が速い。

*1 線形演算ライブラリの BLAS や LAPACK など

とにかく使ってみようぜ

IPython Notebook

IPython Notebook

ブラウザから Python の対話環境を使える。

プロット（描画）した図も中に表示される。

めちゃ便利。

欠点はたまにちょっとだけ不安定。

人によっては操作しづらいかも。

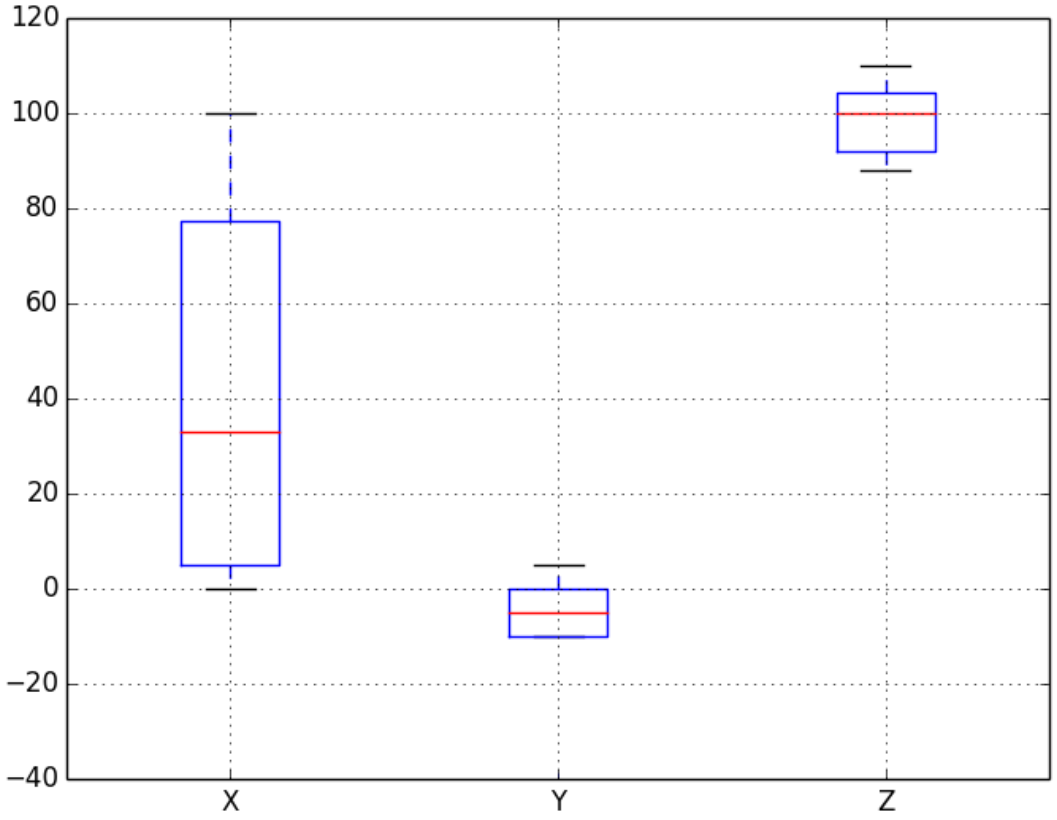
要約統計量の算出

```
>> import pandas as pd
>> df = pd.read_csv('data.csv',
                    Names = ['X', 'Y', 'Z'])

>> df.describe() # 要約統計量を表示する

>> df.boxplot() # 箱ひげ図を描く
```

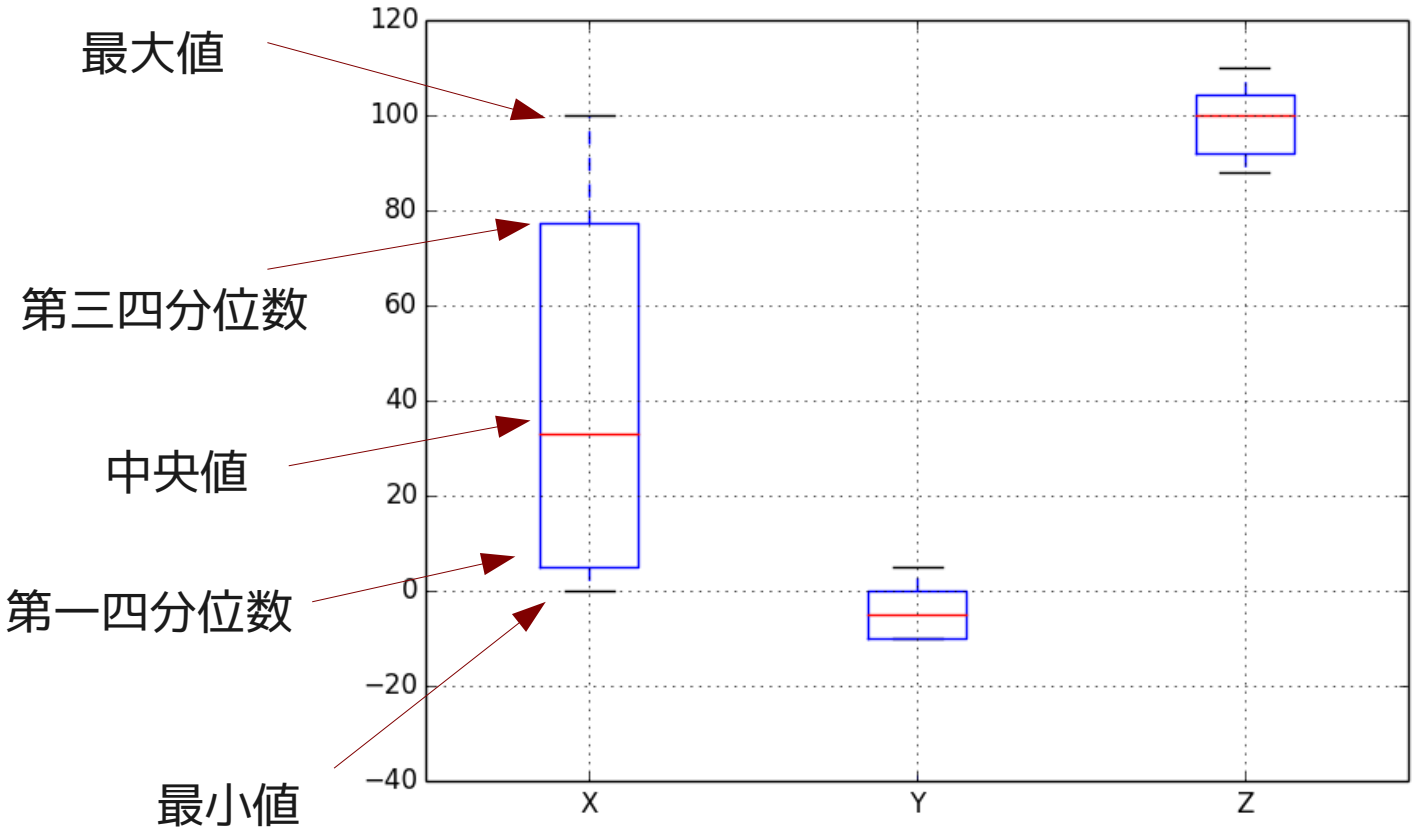
要約統計量の算出

```
>> df.describe()
                X           Y           Z
count    7.000000   7.000000    7.000000
mean    42.571429  -8.571429   98.714286
std     42.968427  14.920424    8.440266
min      0.000000 -40.000000   88.000000
25%      5.000000 -10.000000   92.000000
50%     33.000000  -5.000000  100.000000
75%     77.500000   0.000000  104.500000
max    100.000000   5.000000  110.000000
```

箱ひげ図を描く



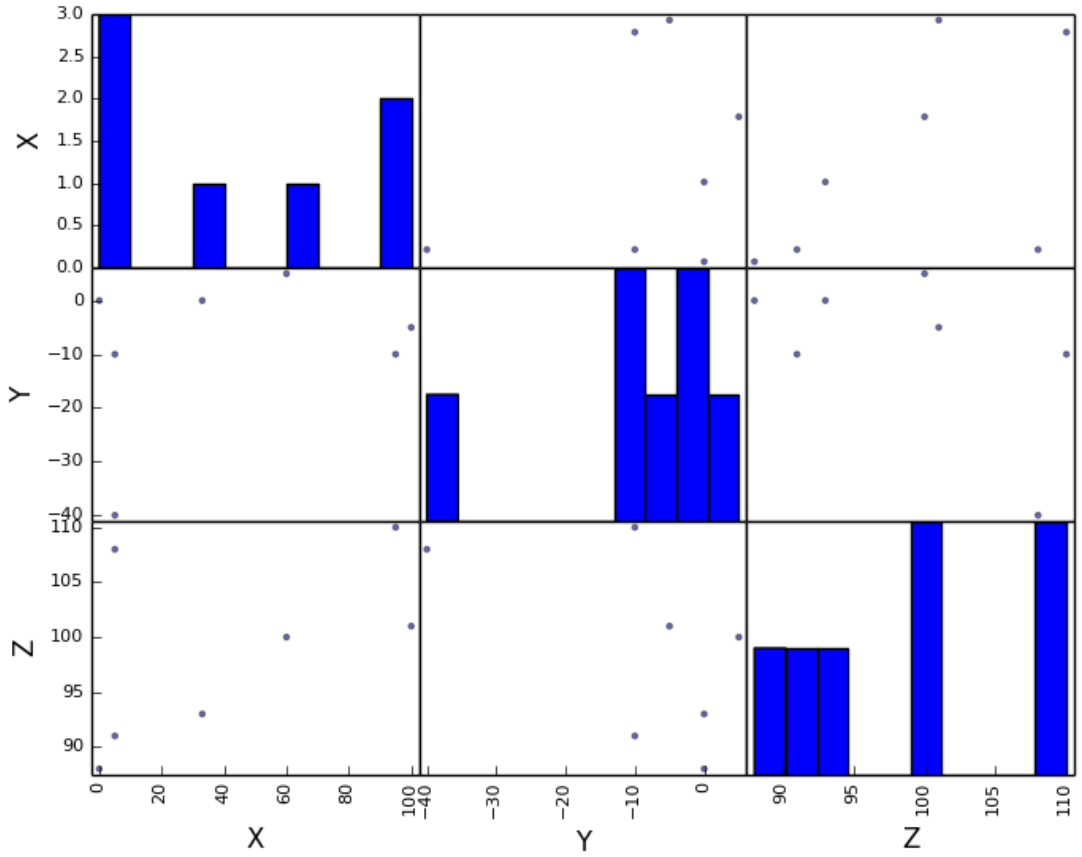
箱ひげ図を描く



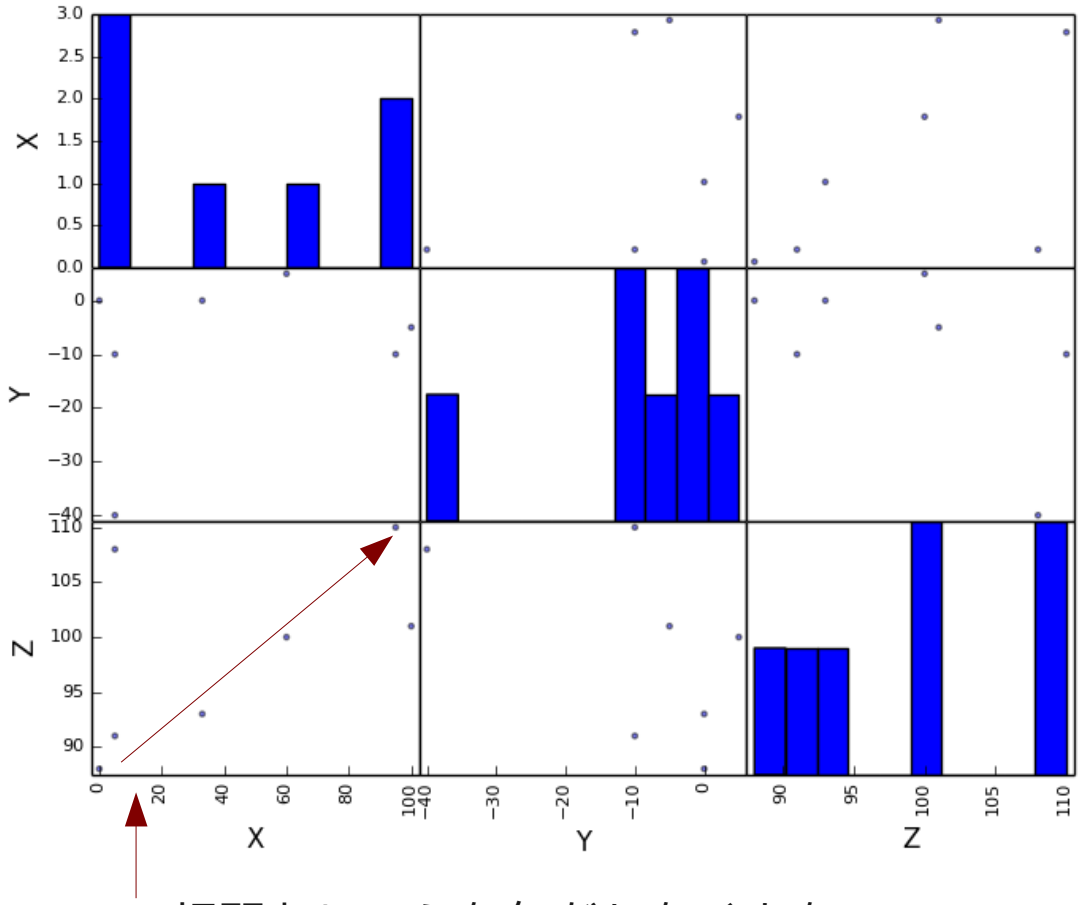
散布図行列を描く

```
>> from pandas.tools.plotting import
   scatter_matrix
>> plt.figure()
>> scatter_matrix(df)
```

散布図行列を描く


X
Y
Z
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
−10
−20
−30
−40
110
105
100
95
90
0
20
40
60
80
100
−40
−30
−20
−10
0
90
95
100
105
110

散布図行列を描く



相関ありそうな気がしなくもない

相関係数を求める

```
>> df.corr()
```

相関係数を求める

```
>> df.corr()
          X         Y         Z
X  1.000000  0.300076  0.550160
Y  0.300076  1.000000 -0.545455
Z  0.550160 -0.545455  1.000000
```

相関係数を求める

```
>> df.corr()
          X         Y         Z
X  1.000000  0.300076  0.550160
Y  0.300076  1.000000 -0.545455
Z  0.550160 -0.545455  1.000000
```

相関係数（絶対値）

0.7 以上 … 強い相関有り

結論

微妙な相関関係ですね…。

結論

ちなみに元記事では

もしかしたら朝食を食べたかどうかではなく、朝食を食べながら業界紙を読む A さんも、出社時間の早い B さんも、朝一で情報収集しているから営業成績が良いんだろうという結論になっています。

回帰式を求める

```
>> from scipy import stats
>> stats.linregress(df['X'], df['Z'])
```

[戻り値]

傾き、切片、相関係数、 P 値、標準誤差

[回帰式]

Y = 傾き * X + 切片

回帰式を求める

```
>> from scipy import stats
>> stats.linregress(df['X'], df['Z'])


(0.1080676770556071,
 94.11369029196326,
 0.5501601429388922 6,
 0.2006940322725799 5,
 0.07335655742148877 7)
```

回帰式を求める

```
(0.1080676770556071,
 94.11369029196326,
 0.5501601429388922 6,
 0.2006940322725799 5,
 0.0733565574214887 77)
```

求める回帰式は　$Y = 0.11x + 94.11$

回帰分析と多変量解析

このように 1 つの目的変数を 1 つの説明変数で予測することです、その 2 変量の間の関係性を y ＝ ax＋b という一次方程式の形で表します。a は傾き、 b は切片です。

これを単回帰分析と言います。

実際のデータ分析ではたくさんの変量を扱いこれを分類していきます。これを多変量解析と言います。

多変量解析の例

- 重回帰分析
- 主成分分析
- 独立成分分析
- 因子分析
- クラスター分析など

仮説検定

ここから仮説検定の話です

仮説検定とは

検定とはある命題が妥当か否かについて一定の確率的根拠に基づいて統計学的に判定することです。

統計学的検定とも言います。

帰無仮説と対立仮説

- 帰無仮説
- 対立仮説

帰無仮説とは、それが棄却（= 否定の意味）されたときに意味を持つ仮定を設定します

国語と数学の成績

出席番号	国語	数学
1	68	86
2	75	83
3	80	76
4	71	81
5	73	75
6	79	82
7	69	87
8	65	75

帰無仮説と対立仮説

- 帰無仮説

  国語と数学には有意な差があるとは言えない
- 対立仮説

  国語と数学の成績には有意な差がある

帰無仮説とは、それが棄却（= 否定の意味）されたときに意味を持つ仮定を設定します

有意性とは

理論比からのズレが誤差の範囲内であるか、あるいはそれ以上の何らかの意味のあるものかを調べることを指します。

統計学では仮説からのズレを有意と言います。統計学的検定とはすなわちこの有意性を検定することになります。

第一種の誤りと第二種の誤り

帰無仮説を棄却することはすなわち対立仮説を採択することになります。

帰無仮説が正しいのにそれを棄却してしまうことを第一種の誤りと言います。

帰無仮説が誤っているのにそれを棄却しないことを第二種の誤りと言います。

有意性検定と論理

有意性検定は、帰無仮説で期待する結果が「生じなかったこと」を根拠として、仮説を棄却するかどうかを決めます。

これは論理学では背理法と言われています。棄却されなかったからといってそれが「積極的に支持されたわけではない」わけではありません。

あくまで帰無仮説が矛盾しないであろうことが言えるわけであり、仮説が真であることを証明したわけではありません。

データフレームにする

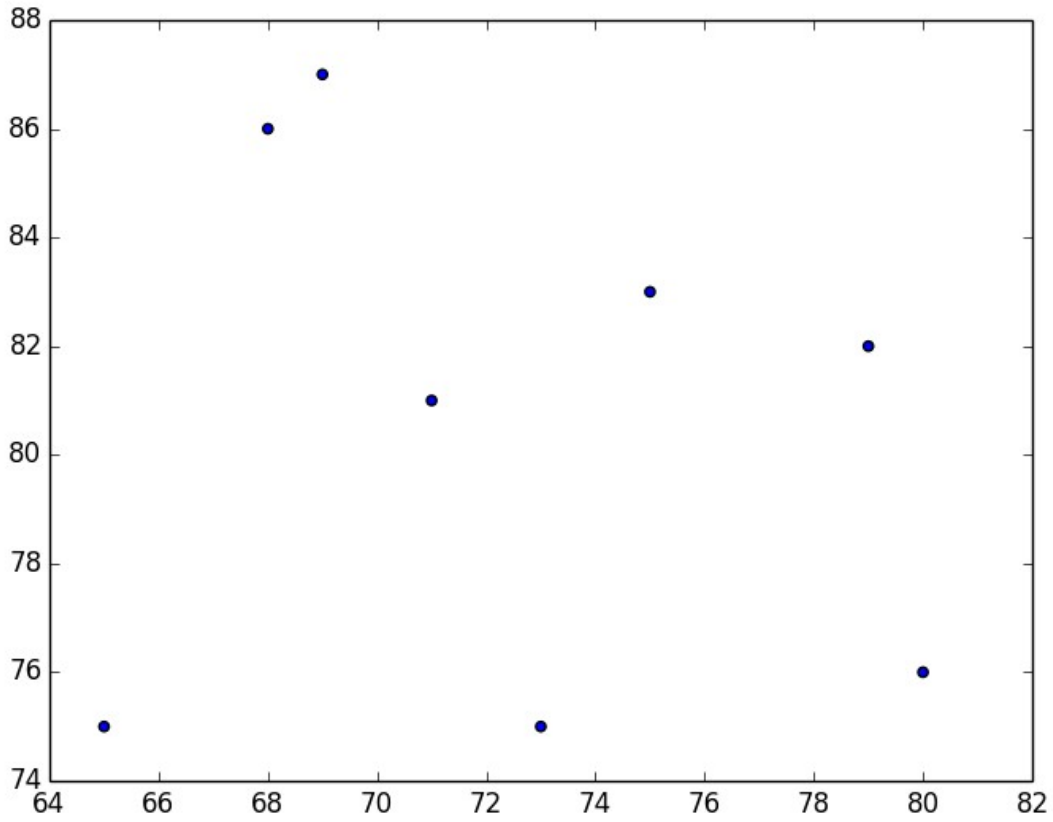
```
>> X = [68, 75, 80, 71, 73, 79, 69, 65]
>> Y = [86, 83, 76, 81, 75, 82, 87, 75]
>> df = pd.DataFrame({'X': X, 'Y': Y})
```

とりあえず散布図

二変量の関係を可視化するにはまず
散布図を描くのが基本です

```
>> plt.figure()
>> plt.scatter(X, Y)
```

散布图



相関係数を求める

```
>> df.corr()

          X         Y
X  1.000000 -0.154388
Y -0.154388  1.000000
```

スチューデントの t 検定

母集団の分散に替えて標本分散を利用するのがスチューデントの t 検定です。

母集団から標本抽出したサンプルの分散を利用します。

たくさんのサンプルは集められないが手元に収集した小規模なサンプルから検定をおこないたいというニーズに応えることができます。

t 検定の一般的な式

標本平均から母集団の平均値が特定の値である数値を調べます。

標本の標準偏差を標本サイズのルートで割り上記をこれで除算すれば t 値が求まります。

$$t = \frac{\overline{X} - \mu_0}{\frac{s}{\sqrt{n-1}}}$$

※自由度 n-1 の場合

※現実的にはウェルチの t 検定の式が使われます

相関係数を求める

```
>> from scipy import stats
>> stats.ttest_rel(X, Y)

(array(-2.9923203754253302),
 0.020160016173681 61)
```

戻り値は t 値、 p 値

有意確率とは

有意確率は P 値とも言います。

ある実験中に群間差が偶然生じる可能性を示す尺度を指します。

例えば P 値が 0.05 というのは、偶然生じることが 100 回に 5 回あることを意味します。

95 % の確率で有意であるというわけです。

統計学は絶対的なものではないのです。

有意水準と結論

P 値が 0.02 だったということは

有意水準を 0.05 に設定しているとすると帰無仮説が棄却されます。

つまり

「国語と数学の成績に有意な差があるとは言えないとは言えない」つまり差があると 95% の確率で言えるのです。

ハンバーガー統計学の問題

有名なハンバーガー統計学の問題を例に挙げてみます。

ワクワク店店長「うちの店では、ポテトやハンバーガーの売り上げは上々で、かなりいいんだ。でも、それに比べると、フライドチキンの売り上げはイマイチなような気がするんだよね」

「それでね、うちのフライドチキンは、ライバル店のモグモグバーガーと比べて、本当に売れてないのか調べたいんだよ。お願い、どうか力を貸してくれないか」

この場合は次の帰無仮説を設定して検定をします。

「チキンとポテトの売り上げの割合に関して、モグモグとワクワクの間に差はない」

http://kogolab.chillout.jp/elearn/hamburger/chap3/sec1.html

調査対象のデータ

店舗	ポテト	チキン
ワクワク	435	165
モグモグ	265	135

ピアソンのカイ二乗検定

ピアソンのカイ二乗検定とは、観察された事象の相対的頻度がある頻度分布にしたがうという帰無仮説を検定します。

$$X^2 = \sum \frac{(O - E)^2}{E}$$

O = 頻度の期待値、 E = 帰無仮説から導かれる頻度の期待値（理論値）

カイ二乗検定をする

```
>> import scipy.stats as stats
>> crossed = sp.array([[435, 265],
                       [165, 135]])
>> stats.chi2_contingency(crossed)
```

戻り値はカイ二乗値、 P 値、自由度、頻度の期待値（理論値）です。

カイ二乗検定と結果

```
>> stats.chi2_contingency(crossed)
(4.1716269841269842,
0.04110630826596564,
1,
Array([[ 420.,  280.],
       [ 180.,  120.]]))
```

有意水準を 0.05 とするとそれより P 値が下回っています。

カイ二乗検定

有意水準は 0.05 が一般的です。

このことからワクワクとモグモグのポテトとチキンの売上には有意な差が無いとは言えないとは言えないことが 95% の確率で言えることになります。

おそらくワクワクのほうがチキンの売上が小さいでしょう。

まとめ

駆け足で統計学の肝である「回帰分析」と「仮説検定」について説明しました。

Python の科学計算ライブラリを使うとこのような計算が簡単におこなえます。

IPython notebook はブラウザさえあれば Python を使えるのでべんり。

ブログ読んでね。

おしまい

ご清聴ありがとうございました