

## 17日目：相関係数（2）

昨日まで、因子分析から平均値の比較へと進めました。本日から方向性を変えて、相関係数の分析をやってみたいと思います。

データも本日から変えてみたいと思います。大学生を対象に、読書習慣に関する調査を行ったとします。読書習慣に関連すると考えられる諸変数も測定し、読書習慣形成との関わりから整理することを狙ってみます。このプロセスで、相関係数、偏相関係数、回帰分析をやってみたいと思います。もちろん、でっち上げのダミーデータですが。

本日は相関係数をやります。しかし、この段階を端折る人も多いのではないかと思います（あくまでも私の想像）、相関係数を出す前には、散布図を眺めておくことが必要だと思います。やはりデータの状況をきちんと眺めておいてから相関係数を算出し、その意味を把握することが大事だと思います。

散布図を作るのは簡単ですから、必ずやっておきましょう。データは **x** という名前を読み込んだとします。

```
plot(x$知識欲, x$不可欠)
```

`plot(x$知識欲, x$不可欠)` のように2変数を併記すればOK。3変数以上を一気にやりたいなら、`plot(x[2:8])` などと複数列を指定するか、変数を別のファイルに入れておいて（たとえば **xx**）、`plot(xx)` というやり方もあります。

複数列を指定すると、興味深い結果を返してきます。散布図マトリックスともいえるような出力が得られます。指定した変数が多すぎると、一つひとつが小さくなりすぎて見にくくなりますが、ある程度までであれば多少小さくても十分ではないでしょうか。なお、右上と左下の部分は、同じ図ではないことがわかると思います。X軸とY軸にとる変数を入れ替えてあります。

これでざっと分布状況を確認したら、相関係数を求めてみます。

10日目に、`cor()` という命令を紹介しました。これは単に相関係数を計算してくれるだけで、ざっと眺めるにはよいですが、論文などに記載するには情報不足です。今日は無相関検定もやってくれる2つのコマンドを取り上げます。

まず一つ目は、

```
cor.test(x$知識欲, x$不可欠, method="pearson")
```

結果は図のように表示されます。なお、`cor()`と同じで、`method=`で違う方法を指定できますし、指定しなければ (`cor.test(x$知識欲, x$不可欠)`) ピアソンの積率相関係数を求めます。

相関係数とともに、無相関検定の `t` 値、自由度、`p` 値を表示しています。この出力の興味深いところは、95 percent confidence

```
> cor.test(x$知識欲, x$不可欠, method="pearson")

Pearson's product-moment correlation

data: x$知識欲 and x$不可欠
t = 8.3055, df = 160, p-value = 3.975e-14
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.4311574 0.6482000
sample estimates:
      cor
0.5488631
```

interval, つまりこ

の相関係数の95%信頼区間も出してくれるところです。今回のデータから、母集団における相関係数を推定した場合、95%の確率で.431から.648の間にあるという推定です。有意だけでも弱い相関が得られた場合、これを見て、落ち着いて解釈をしてください。

これはかなり必要な値をカバーしてくれる出力をしてくれる (`n` が出てきませんが…) のですが、面倒なのは複数列に渡る変数指定ができないところです。`cor.test(x)`や、`cor.test(x[2:9])`とするとエラーになります…。

もうひとつは、`psych` パッケージにある、`corr.test()`です (`r` がひとつ多い!)。こちらは複数列の指定も受け付けてくれます。なお、`method=`は `cor()`や `cor.test()`と同じです。

```
corr.test(x[2:9], method="pearson")
```

この場合の出力は、相関係数、データ数、有意確率の3つの指標になります。なかなか便利なのですが、相関係数も有意確率も小数点以下2桁で表示されます。ここは好みの分かれるところでしょう。4桁という論文にはまず出会いませんが、3桁表示は多くあります。

`round` でコントロールできるかやってみたのですが、うまくいきませんでした。そこで、`corr.test` の結果をいったん保存し、相関係数 (`r`)、データ数 (`n`)、有意確率 (`p`) をそれぞれに呼び出すという形だとうまくいきます。

この仕組みは、いろいろと応用が効きますので覚えておいて損はないと思います。`R` が返してくる結果は、一つのオブジェクトであり、いくつかの内容が含まれていることが多いです。たとえば因子分析の時に、結果を一度保存して、`print` でコントロールしました。このようなやり方は他にもありました。それは、求めた結果が複数の内容を含んでいるからできることです。通常の結果表示は、それをまとめたデフォルト表示なのです。

どのような内容が計算結果のオブジェクトに含まれるかは、それをいったん何かに代入し、

`names` で中を見ることができます。たとえば、`corr.test` の結果なら…

```
bb <- corr.test(x[2:9])
names(bb)
```

これを実行すると、以下のような結果が返ってきます。

```
> bb <- corr.test(x[2:9])
> names(bb)
[1] "r"      "n"      "t"      "p"      "adjust" "sym"    "Call"
```

この結果は、`bb` の中に、`"r"`、`"n"`、`"t"`、`"p"`、`"adjust"`、`"sym"`、`"Call"` の7つの結果が含まれていることを示しています。そして、それぞれの中身を確認するには、`bb$r` などとすれば見ることができます（`r` とか `n` などは、`bb` というデータの変数名（列名）と考えればよいでしょう。`x$知識欲`で、`知識欲`のデータが見られるのと同じイメージです）。もちろん `help` でも確認できますが。

それぞれを確認すると、`r` が相関係数、`n` が人数、`p` が有意確率であることがわかります。デフォルトの表示は、`r` や `p` の値を小数点以下2桁に丸めてあるようです。そこで、これを小数点以下3桁に丸めて表示するように、`round` でコントロールしてやります。つまり…

```
bb <- corr.test(x[2:9])
round(bb$r, 3)
bb$n
round(bb$p, 3)
```

これで、大体デフォルトの出力と同じで、かつ小数点以下3桁表示になります。

さて、相関係数を求める場合には、行と列に同じ変数を入れたい場合もありますが、別の変数を指定したい場合も多いでしょう。そのような場合には、`corr.test(x[8:9], x[2:7])` というような指定もできます。指定の前半部が行（つまり縦）になり、後半部が列（つまり横）になります。なお、10日目には説明しませんでした。が、`cor` も同様です。`cor(x[8:9], x[2:7])` という指定ができます。

```
> corr.test(x[8:9], x[2:7])
Call:corr.test(x = x[8:9], y = x[2:7])
Correlation matrix
      親の様子 家にある本 知識欲 不可欠さ インターネット マンガ
本好き    0.48      0.32    0.22    0.25      0.24    0.02
読書習慣    0.49      0.42    0.39    0.23     -0.03    0.01
Sample Size
      親の様子 家にある本 知識欲 不可欠さ インターネット マンガ
本好き    162      162    162    162      162    162
読書習慣    162      162    162    162      162    162
Probability values adjusted for multiple tests.
      親の様子 家にある本 知識欲 不可欠さ インターネット マンガ
本好き      0        0    0.02    0.01     0.01    1
読書習慣      0        0    0.00    0.01     1.00    1
> |
```

cor, cor.test, corr.test の結果を比較してみると、corr.test が一番見やすいのかなと思います。

しかし、論文に記載する表に加工しようとする、いずれもちょっと手間がかかりそうで、また転記ミスをする可能性もありそうです。

本日はここまでにします。明日は論文に掲載する表の形式に近いものを出力する自作関数を紹介しようと思います。