Correlation & Distance

Rui Zhou

2026-05-26

目录

1 这章在讲什么

「相关」是统计入门绕不开的概念，但实际上有三种相关常被混用：

- **零阶相关** $r(X, Y)$：直接算 X 和 Y 的相关系数，不控制任何其他变量。
- **偏相关**（partial correlation）$r(X, Y \mid Z)$：先把 X 和 Y 各自对 Z 回归取残差，再算两个残差的相关——这是「控制 Z 之后」X 和 Y 的

纯净相关。

- **半偏相关 / 部分相关**（part / semi-partial correlation）$r(X, Y \cdot Z)$：只把 Y 对 Z 回归取残差，X 不动——这是「Z 已经解释了 Y 那部分，剩下的部分跟 X 的相关」。多元回归里每个解释变量对 R^2 的「独特贡献」用的就是它。

三种相关的关系

设回归 $Y = \beta_0 + \beta_1 X + \beta_2 Z + \varepsilon$。则：

$$\text{偏相关 } r(X, Y \mid Z) = \frac{r_{XY} - r_{XZ} r_{YZ}}{\sqrt{(1 - r_{XZ}^2)(1 - r_{YZ}^2)}}$$

$$\text{半偏相关}^2 = R^2_{\text{full}} - R^2_{\text{without X}}$$

实际意义：**偏相关**回答「X 单独和 Y 的关系有多强（剔除 Z 的所有影响）」；**半偏相关**回答「在已有 Z 的模型里，X 能额外解释多少 Y」。心理学论文常说「控制人口学变量后……」，里头数学是偏相关。

第二个主题是**距离**——这是聚类、PCA、判别分析的基础。最常用的几种：

- **欧氏距离** $d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_i (x_i - y_i)^2}$：直觉对应「直线距离」，对量纲敏感。
- **马氏距离**（Mahalanobis）：欧氏距离对变量的协方差矩阵做白化校正，自动消除量纲和相关性。
- **曼哈顿距离** $\sum_i |x_i - y_i|$：对离群值更稳健，适合高维。

```
library(bruceR)
library(rstatix)
library(tidyverse)
library(stargazer)
```

2 Correlation

2.1 Bivariate Correlation

Here we're to discuss more about bivariate correlation when taking groups into account. As you can see, the pooled data may interact and neutralize the effect or relationship of two variables. However, after we inspect that correlation under each group, surprising outcomes may spring up!

Take dataset "car_sales.sav" as an example, which you have been quite similar to "mtcars". We're interested in whether or not people value fuel-efficient cars more.

```
carsales = import("car_sales.sav")
```

```
##   Successfully imported: 157 obs. of 26 variables
```

```
attach(carsales)
```

```
ggplot(data = carsales, mapping = aes(x = mpg,y = sales))+
  geom_point()
```

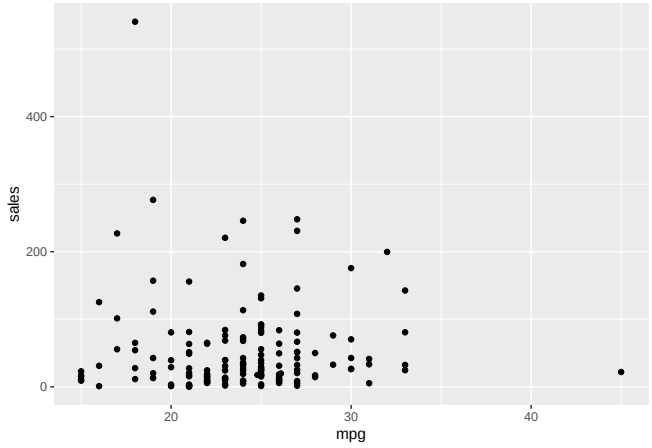


图 1: Scatter Plot of mpg and sales with ALL Observations

At first glance, there are apparently some outliers. Moreover, the scatter

points seem to cluster in the bottom-left, possibly a skewed distribution instead of normal! Nevertheless, those don't bother us to check the overall correlation first.

```
cor(sales,mpg,use = 'c')
```

```
## [1] -0.01668058
```

The correlation is not high in magnitude. Here, use "`cor()`" for simplicity. Note that for data with missing values, specify parameter "`use = 'complete.obs'`" (or "`use = 'c'`" as abbreviation) to compute only with complete cases.

Admittedly, "`cor.test()`" will also return the correlation coefficient, along with outcomes with hypothesis testing. However, the latter makes it to be complex unnecessarily! (Good thing is that, "`cor.test()`" will automatically wipe out cases with missing values.)

```
cor.test(mpg,sales)
```

Then, we're to deal with outliers and skewness, then to recompute correlation.

Good news is that, log transformation helps when it comes to skewed distribution, and outliers will get contained under log transformation at the same time.

```
cor(log(mpg),log(sales),use = 'c')
```

```
## [1] 0.1141838
```

The correlation coefficient is improved in magnitude, but not significant after all.

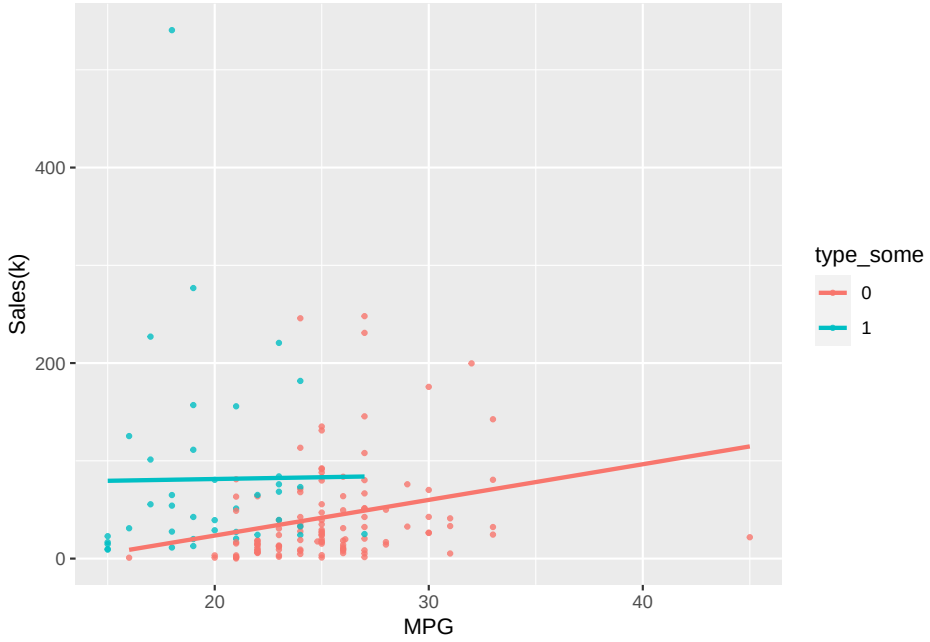
Then, here comes what's the most important point. When faced with a brand-new data, before conducting any computation and hypothesis testing, getting an overall picture of the data is the most important thing to do!

After observation, it's likely that cars whose model are "Metro" or "F-Series"

are special compared to others. Accordingly, we're to group the data and then recompute correlation.

```
sales_some = sales[model!="Metro"&model!="F-Series"]
mpg_some = mpg[model!="Metro"&model!="F-Series"]
type_some = as.factor(type[model!="Metro"&model!="F-Series"])

df = data.frame(sales_some,mpg_some,type_some)
df %>%
  ggplot(aes(mpg_some, sales_some,color = type_some)) +
  geom_point(alpha=0.8, size=0.8) +
  geom_smooth(method="lm",se=FALSE) +
  ylab('Sales(k)') +
  xlab('MPG')
```



Use graphs to think! The categorized data plays an important role here, and unfolds the underlying correlation. Based on the prior observation, then check correlation quantitatively.

```
cor.test(sales_some[type_some=="1"],mpg_some[type_some=="1"])
```

```
##
##  Pearson's product-moment correlation
##
## data:  sales_some[type_some == "1"] and mpg_some[type_some == "1"]
## t = 0.070377, df = 38, p-value = 0.9443
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.3011645  0.3217809
## sample estimates:
##        cor
## 0.01141584
```

```
cor.test(sales_some[type_some=="0"],mpg_some[type_some=="0"])
```

```
##
##  Pearson's product-moment correlation
##
## data:  sales_some[type_some == "0"] and mpg_some[type_some == "0"]
## t = 2.8695, df = 112, p-value = 0.004915
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.0817139 0.4251484
## sample estimates:
##       cor
## 0.2616958
```

Finally, don't forget to

```
detach(carsales)
```

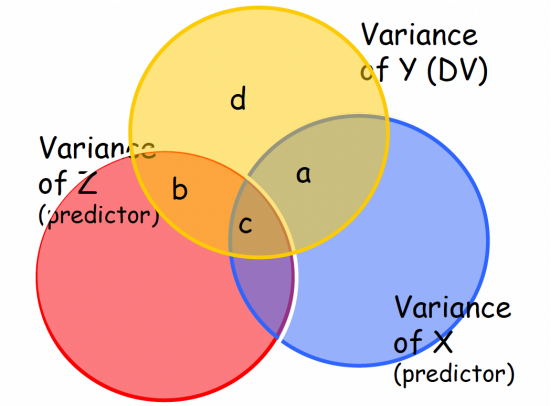
2.2 Partial & Part Correlation

2.2.1 Background Knowledge

Above all, I'd like to clarify one concept that we've been familiar with but without mentioning this concept. Zero-order correlation refers to the correlation between X and Y without controlling other covariates. Traditionally, Pearson and Spearman correlations are typical of zero-order correlation.

We've discussed partial and part (or, semi-partial) correlation in multilinear regression. Take a step further, we can consider those under the most general cases, i.e., without the setting of MLR but some variables.

Suppose we have three variables, called X, Y and Z, where X is the predictor, Y is the dependent variable, and Z is the covariate. Noteworthy that the assumption of the roles of X, Y, Z don't matter much, only for accommodating the setting.



Note that, in the general case, Y is not represented as the whole area in the picture, but just an equal role as a circle.

If the correlation of X and Y is of our great interest,

$$r_x = \frac{a+c}{a+b+c+d}$$
$$pr_x = \frac{a}{a+d}$$
$$spr_x = \frac{a}{a+b+c+d}$$

- Partial: Holding Z from BOTH X and Y, i.e., removing the effect of Z.

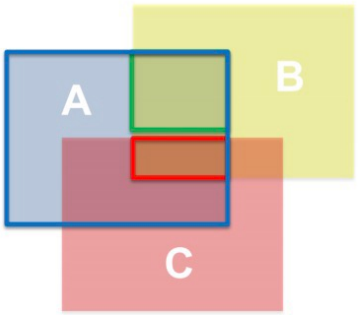
  Good for knowing the **variance uncounted by other control variable(s)**.

- Semi-Partial: Holding Z from X only.

  Good for knowing the **unique contribution of** X.

- Zero-Order: No extra constraint.

One more enlightening way to think about partial correlation is through multilinear regression.



If we're interested in the partial correlation between A and B, with C as the covariate. Then, what we need to do is partial out the effect of C on BOTH A and B, and this goal can be accomplished through step-by-step linear regression.

Specifically, first we conduct a simple linear regression of A on C, and the residual part is that of A which doesn't intersect with C. Similarly, then followed by a SLR of B on C, same with the meaning of residual part. For

the two residual parts, they share it in common that they're clean of C. Lastly, compute the correlation between A and B directly, and that's what we want!

Even with more covariates, conduct A and B on all the covariates and then conpute the correlation with their residual parts. Done!

2.2.2 Examples

We first man-make a data that vividly shows the differences between partial and zero-order correlation.

```
DV = c(1:8, 8:11)
IV = c(4:1,8:5,12:9)
covariate = c(rep(0,4), rep(4,4), rep(7,4))
studiedData = data.frame(DV,IV,covariate)

studiedData |> ggplot(aes(x = IV,y = DV))+
  geom_point()
```

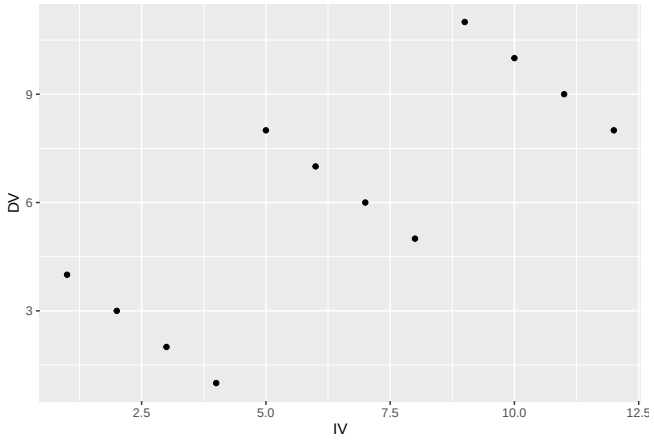


图 2: Scatter Plot of X and Y

Clearly, considering zero-order correlation, X is strictly negatively-correlated with Y, the covariate is strictly positively-correlated with Y.

However, X shows partially positive correlation with Y overall.

```
cor(IV,DV)
```

```
## [1] 0.7608292
```

If you're to study the partial correlation, things will be different.

```
library(ppcor)
pcor(studiedData)
```

表 1: Partial Correlation between Variables

	DV	IV	covariate
DV	1	-0.972	0.991
IV	-0.972	1	0.993
covariate	0.991	0.993	1

One more example. A popular radio talk show host has just received the latest government study on public health care funding and has uncovered a startling fact: As health care funding increases, disease rates also increase! Cities that spend more actually seem to be worse off than cities that spend less. Then, shall we just cut the funding and save people's health?

```
# example: a covariate can "make" a correlation that should not exist
currentData <- import("health_funding.sav")
```

```
##   Successfully imported: 50 obs. of 4 variables
```

```
attach(currentData)
studiedData = data.frame(disease,funding,visits)
```

First of all, check overall zero-order correlation.

```
cor(studiedData)
```

It seems astonishing that "`funding`" is greatly correlated with "`disease`"!

表 2: Zero-Order Correlation

	disease	funding	visits
disease	1	0.737	0.762
funding	0.737	1	0.964
visits	0.762	0.964	1

May it offer the plausible reason to cut the funding? Remember, we still have the covariate "`visits`". One alternative explanation might be that, offered with more funding, people may be more likely to pay visits to health care, and thus more diseases are detected, which should have been detected but not due to wealth state.

```
pcor(studiedData)$estimate
```

表 3: Partial Correlation

	disease	funding	visits
disease	1	0.013	0.286
funding	0.013	1	0.920
visits	0.286	0.920	1

It's astonishing that, considering the partial correlation, "`funding`" almost has no correlation with "`disease`" in essence!

Additionally, we can consider the part correlation.

```
spcor(studiedData)$estimate
```

And finally,

```
detach(currentData)
```

表 4: Part Correlation

	disease	funding	visits
disease	1	0.009	0.193
funding	0.004	1	0.622
visits	0.076	0.596	1

3 Distance

We need to distinguish between two kinds of distance. One is distance between cases (or observations), the other is distance between variables.

There are traditionally three ways to measure distance, either from the perspective of dissimilarity or similarity.

- Dissimilarity
  - Euclidean distance

$$d(\vec{p}, \vec{q}) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

  - Chebychev distance

  A vector space where the distance between two vectors is the greatest of their differences along any coordinate dimension.

$$D(\vec{x}, \vec{y}) = \max_i |x_i - y_i|$$

- Similarity
  - Cosine similarity
    - A measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them.
    - It is thus a judgment of orientation and not magnitude.

```
currentData = import("data10-04.sav", sheet = 1)
```

```
##   Successfully imported: 12 obs. of 6 variables
```

```
attach(currentData)
# only use the data that should be considered or analyzed
studiedData = data.frame(hgrow,temp,rain,hsun,humi)
```

3.1 Correlation

Note that correlation itself is a measure of distance. However, correlation only applies for **distance between variables**, not for distance between cases. (If you insist to test the correlation between cases, easy, just transpose your data!)

```
cor(studiedData)
```

表 5: Correlation between Variables

	hgrow	temp	rain	hsun	humi
hgrow	1	0.983	0.709	0.704	0.374
temp	0.983	1	0.715	0.690	0.292
rain	0.709	0.715	1	0.702	0.384
hsun	0.704	0.690	0.702	1	-0.051
humi	0.374	0.292	0.384	-0.051	1

3.2 Euclidean Distance

Then, we're interested in Euclidean distance **between cases** by default. Use "`dist(x, method = "euclidean")`". Mind that the distance makes sense only if the data is scaled!

```
# studiedData %>% dist()
studiedData  %>%
  scale() %>% dist()
```

表 6: Euclidean Distance between Cases

	1	2	3	4	5	6	7	8	9	10	11
1	0	0.925	2.142	3.692	4.790	3.904	4.078	5.793	3.764	2.536	1.782
2	0.925	0	1.288	2.856	4.179	3.338	3.756	5.505	3.576	2.232	1.433
3	2.142	1.288	0	2.149	3.774	3.151	3.962	5.710	3.870	2.675	2.054
4	3.692	2.856	2.149	0	2.558	2.001	3.108	4.535	3.518	2.627	2.509
5	4.790	4.179	3.774	2.558	0	2.307	2.871	2.895	2.724	3.151	3.662
6	3.904	3.338	3.151	2.001	2.307	0	1.265	3.338	1.937	1.636	2.236
7	4.078	3.756	3.962	3.108	2.871	1.265	0	2.696	1.292	1.566	2.448
8	5.793	5.505	5.710	4.535	2.895	3.338	2.696	0	2.761	3.721	4.519
9	3.764	3.576	3.870	3.518	2.724	1.937	1.292	2.761	0	1.654	2.541
10	2.536	2.232	2.675	2.627	3.151	1.636	1.566	3.721	1.654	0	0.967
11	1.782	1.433	2.054	2.509	3.662	2.236	2.448	4.519	2.541	0.967	0
12	0.591	0.923	2.192	3.447	4.501	3.529	3.629	5.333	3.402	2.093	1.374

If the Euclidean distance between **variables** is of interest, then *transpose* your data.

```
dist = studiedData  %>%
  scale() %>% t() %>% dist()
```

It's noteworthy that, the Euclidean distance result returned by "`dist()`" is a unique type of "`dist`", which is incompatible with some other types. Luckily, use "`as.matrix()`" to transform that into a matrix (with the diagonal and upper right triangle filled automatically).

表 7: Euclidean Distance between Variables

	hgrow	temp	rain	hsun	humi
hgrow	0	0.605	2.529	2.550	3.712
temp	0.605	0	2.505	2.609	3.947
rain	2.529	2.505	0	2.561	3.680
hsun	2.550	2.609	2.561	0	4.808
humi	3.712	3.947	3.680	4.808	0

3.3 Cosine Similarity

Also, we can see the cosine similarity **between cases**.

Remember, if you're to study the distance between cases, transpose your matrix of data; if you're to study the distance between variables, throw the matrix of data into "`lsa::cosine()`" directly.

```
studiedData %>%
  as.matrix.data.frame() %>% t() %>%
  lsa::cosine() %>%
  stargazer(title = 'Cosine Similarity between Cases')
```

And finally,

```
detach(currentData)
```

表 8: Cosine Similarity between Cases

1	0.985	0.968	0.886	0.626	0.886	0.851	0.632	0.643	0.934	0.970	0.994
0.985	1	0.994	0.948	0.609	0.901	0.846	0.616	0.597	0.931	0.995	0.997
0.968	0.994	1	0.974	0.664	0.934	0.881	0.671	0.639	0.950	0.990	0.986
0.886	0.948	0.974	1	0.662	0.927	0.860	0.669	0.600	0.913	0.954	0.925
0.626	0.609	0.664	0.662	1	0.876	0.929	1.000	0.984	0.847	0.571	0.609
0.886	0.901	0.934	0.927	0.876	1	0.989	0.882	0.852	0.991	0.890	0.894
0.851	0.846	0.881	0.860	0.929	0.989	1	0.934	0.920	0.982	0.828	0.847
0.632	0.616	0.671	0.669	1.000	0.882	0.934	1	0.984	0.853	0.580	0.616
0.643	0.597	0.639	0.600	0.984	0.852	0.920	0.984	1	0.843	0.556	0.610
0.934	0.931	0.950	0.913	0.847	0.991	0.982	0.853	0.843	1	0.916	0.932
0.970	0.995	0.990	0.954	0.571	0.890	0.828	0.580	0.556	0.916	1	0.990
0.994	0.997	0.986	0.925	0.609	0.894	0.847	0.616	0.610	0.932	0.990	1

4 容易踩的坑

相关与距离的常见误区

1. **把相关当因果**：$r(X,Y) = 0.8$ 不代表 $X \to Y$，可能 $Y \to X$、$Z \to X, Y$、或纯巧合。教科书例子：「冰淇淋销量」和「溺水率」高相关——其实是「天气」同时影响两者。
2. **Pearson r 用在非线性关系上**：r 只衡量**线性**相关。两个变量是 $Y = X^2$ 的关系，r 能算到 0。先画散点图再算 r。
3. **偏相关解释成「Z 不存在时的相关」**：偏相关是「Z 取固定值时」X 和 Y 的关系，不是「假如 Z 不存在」。这两件事很不一样——前者前提是 Z 仍然影响 X 和 Y，只是在每个 Z 水平上分别看。
4. **欧氏距离不标准化就跑**：跟聚类一节同理，量纲差异会让距离被某一两个大量级变量主导。用 `scale()` 标准化是最简方案；用马氏距离能自动校正。