治理技术专题

定量政治分析方法

Quantitative Analysis II

苏 毓 淞

清华大学社会科学院政治学系

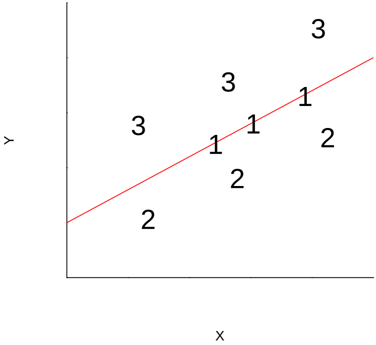
第十一讲　时间序列分析

复习：余数项独立假定

- $y_i = \alpha + \beta x_i + \epsilon_i, \qquad i = 1, 2, \ldots, n$
- $\mathsf{E}(\epsilon_i) = 0, \mathsf{var}(\epsilon_i) = \sigma^2$
- 我们无法从任一个单元余数去预测另一单元的余数
- 如果余数项独立假定不成立，则：
  - 标准误、假设检定、信用区间都是错的。
  - 最小二阶方程不适合用来估计 β。

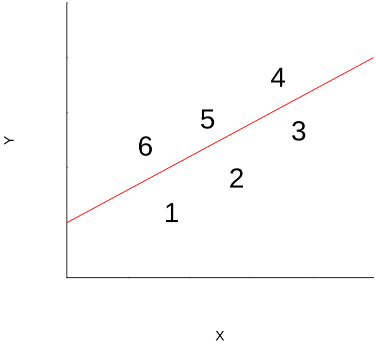
常见违反独立假定的情况

- 簇效应 (cluster effect)
- 同类相聚



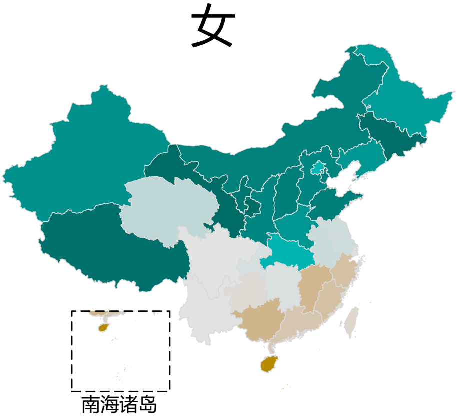
常见违反独立假定的情况

- 序列效应 (serial effect)
- 常见于依时间或依空间搜集来的数据



空间自相关 (Spatial Autocorrelation)

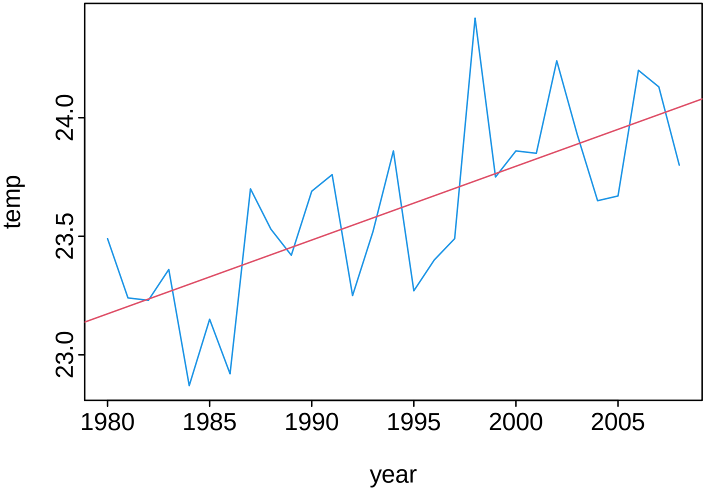
- 粽子口味地图，对女性来说，口味南咸北甜。



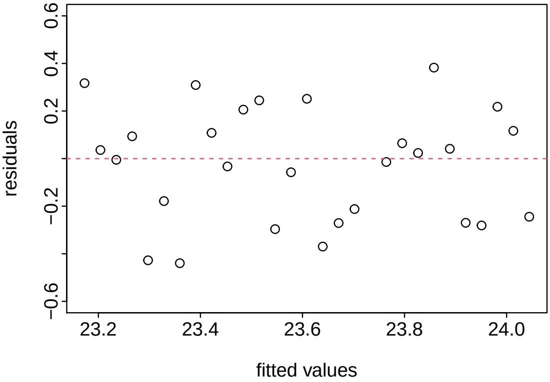
时间序列分析 (Time Series Analysis)

- 数据在时间上自相关，或称为序相关 (serial correlation)
  - 某个时间点的余数 (ϵ_t) 与其他时间点的余数 ($\epsilon_{t-1}, \epsilon_{t-2}$) 相关
  - 例如关税率、汇率、在位者投票优势。
- STATA 中，时间序列分析基本命令：
  - 向 STATA 宣告你的数据是时间序列数据：
    `tsset var` 其中 `var` 是时间变量
  - 使用字首 `L.` 表示滞后变量 (lagged variable)：
    `L.var` 表示滞后一个时间单元，`L2.var` 表示滞后两个时间单元
  - 使用字首 `F.` 表示前向变量 (forward variable)：
    `F.var` 表示前向一个时间单元，`F2.var` 表示前向两个时间单元

以可视化的方式查验序相关的问题

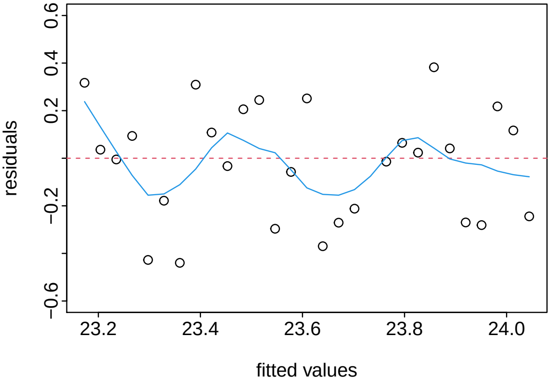


以可视化的方式查验序相关的问题



看起来没太大的问题...

以可视化的方式查验序相关的问题



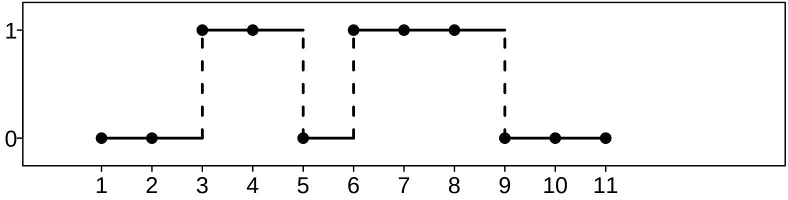
加入 lowess 线后，发现余数随着时间呈现跌宕起伏的循环....

游程检验 (runtest)

- 将两独立样本混合后，排序后得到秩（rank），来自总体 1 的样本秩为 0，总体 2 则为 1。

秩	1	2	3	4	5	6	7	8	9	10	11
	0	0	1	1	0	1	1	1	0	0	0

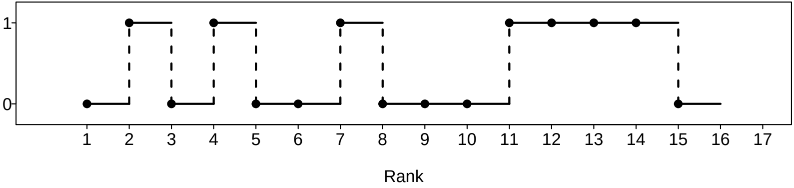
- 来自 1 的有 2 个游程，来自 0 的有 3 个，共有 5 个游程。
- 如果两独立样本相等，秩是交错的，游程是应该是交错相等的。



游程检验案例

A	9	22	64	34	17	4	31	28
B	58	53	26	11	52	51	8	

4	8	9	11	17	22	26	28	31	34	51	52	53	58	64
0	1	0	1	0	0	1	0	0	0	1	1	1	1	0



游程检验案例

```
. runtest run
 N(run <= 0) = 8
 N(run >  0) = 7
         obs = 15
     N(runs) = 9
          z  = .29
    Prob>|z| = .77
```

游程检验 (runtest)(Normal Approximation)

当 $n_1 \geq 20$,$n_2 \geq 20$，可用正态分布近似法求之：

$$\mu = \frac{2n_1n_2}{n_1+n_2} + 1$$

$$\sigma^2 = \frac{2n_1n_2(2n_1n_2 - n_1 - n_2)}{(n_1+n_2)^2(n_1+n_2-1)}$$

$$Z = \frac{r-\mu+c}{\sigma} \sim N(0,1)$$

- $r =$ 游程数量

以游程检验查验序相关的问题

```
> chk <- res>0
> n <- length(chk)
> r <- 1+ sum(chk[-1] !=chk[-n])
> n1 <- sum(chk)
> n2 <- n-sum(chk)
> mu <- (2*n1*n2/(n1+n2))+1
> s <- sqrt((2*n1*n2*(2*n1*n2-n1-n2))/((n1+n2)^2*(n1+n2-1)))
> z <- (r - mu)/s
> z
[1] 0.1958652
> 2*pnorm(-abs(z))
[1] 0.8447157
```

以游程检验查验序相关的问题

```
> cbind(year, res, res>0)
   year          res
1  1980  0.317102945 1
2  1981  0.035979804 1
3  1982 -0.005143567 0
4  1983  0.093734360 1
5  1984 -0.427388553 0
6  1985 -0.178512915 0
7  1986 -0.439635599 0
8  1987  0.309241946 1
9  1988  0.108118728 1
10 1989 -0.033005024 0
11 1990  0.205872292 1
12 1991  0.244748846 1
13 1992 -0.296374525 0
14 1993 -0.057497209 0
15 1994  0.251379802 1
16 1995 -0.369743492 0
17 1996 -0.270867473 0
18 1997 -0.211990462 0
19 1998  0.686886702 1
20 1999 -0.014236516 0
21 2000  0.064640953 1
22 2001  0.023517582 1
23 2002  0.382393830 1
24 2003  0.041271223 1
25 2004 -0.269852605 0
26 2005 -0.280975289 0
27 2006  0.217902256 1
28 2007  0.116777512 1
29 2008 -0.244345553 0
```

```
> table(res>0)

FALSE  TRUE
   14    15
```

以游程检验查验序相关的问题

```
> library(tseries)
> runs.test(as.factor(sign(res)))

        Runs Test

data:  as.factor(sign(res))
Standard Normal = 0.19587, p-value = 0.8447
alternative hypothesis: two.sided
```

- H_0：游程是随机产生的。
- Z 是统计不显著接受 H_0，游程是随机产生的。

自相关系数 (Autocorrelation Coefficients)

- 估计与滞后 k 阶的自相关系数即：余数项与滞后 k 阶余数项的相关系数
- 自相关系数 r_k:

$$c_k = \frac{1}{n}\sum_{t=1}^{n-k}(x_t - \bar{x})(x_{t+k} - \bar{x})$$

$$r_k = \frac{c_k}{c_0} = \text{cor}(x_t, x_{t+k})$$

- r_k 的 95% 信用区间：

$$-\frac{1}{n} \pm \frac{2}{\sqrt{n}}$$

n 是计算 r_k 时的样本量。

部分自相关系数 (Partial Autocorrelation Coefficients)

- 估计与滞后 k 阶的部分自相关系数即：在控制了滞后 1 阶,. . .,滞后 $k-1$ 阶余数项后，余数项与滞后 k 阶余数项的相关系数
- 部分自相关系数 f_k;

$$f_k = \begin{cases} \text{cor}(x_1, x_0) = r_1 & \text{if } k = 1; \\ \text{cor}(x_k - x_k^{k-1}, x_0 - x_0^{k-1}) & \text{if } k \geq 1; \end{cases}$$

- f_k 的 95% 信用区间：

$$-\frac{1}{n} \pm \frac{2}{\sqrt{n}}$$

n 是计算 r_k 时的样本量。

交互相关系数 (Cross-correlation Coefficients)

- 两个时间序列之间的交互的相关系数。
- 自相关系数 r_k:

$$g_k^{xy} = \frac{1}{n}\sum_{t=1}^{n-k}(y_t - \bar{y})(x_{t+k} - \bar{x})$$

$$r_k^{xy} = \frac{g_k^{xy}}{\sqrt{\sigma_x \sigma_y}}$$

- r_k 的 95% 信用区间：

$$-\frac{1}{n} \pm \frac{2}{\sqrt{n}}$$

  n 是计算 r_k 时的样本量。
- 注意：$r_k^{xy} \neq r_{-k'}^{xy}$，但是 $r_k^{xy} \neq r_{-k'}^{yx}$，所以哪个变量是 x 哪个是 y 影响 r_k^{xy} 值

时间序列相关系数显著性问题

- 95% 信用区间：

$$-\frac{1}{n} \pm \frac{2}{\sqrt{n}}$$

- 一般来说，n 会随着阶数增加而减少，致使区间变大，但是多数软件使用数据原样本量来绘制两条水平线，亦即 n 不随阶数增加而减少。
- 虽然 95% 信用区间可以用来表述相关系数的显著性与否，但是 don't oversell it，因为时间序列的相关系数显著性有很大的概率是随机的。

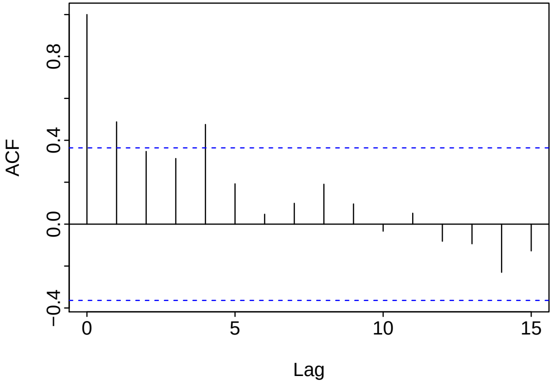
判定数据的 AR(?) 模型

- 判断数据是几阶 (order) 自回归 (Autoregressive, AR) 模型，必须查验部分自相关系数。
- 自相关系数非必要！
- 部分自相关系数为查验 AR 模型的必要条件，而非自相关系数。

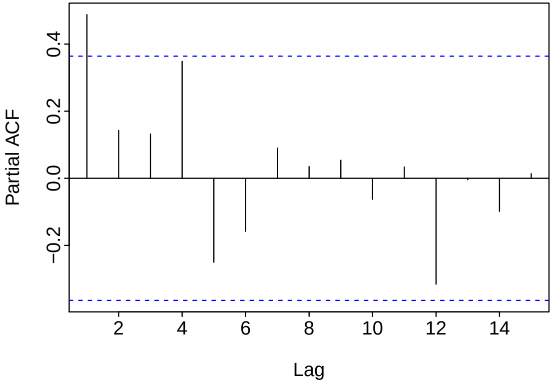
查验自相关

```
> print(acf(temp, lag.max=15))

Autocorrelations of series ‘temp’, by lag

     0      1      2      3      4      5      6      7      8      9     10
 1.000  0.488  0.347  0.313  0.476  0.192  0.047  0.099  0.190  0.096 -0.033
    11     12     13     14     15
 0.052 -0.082 -0.094 -0.230 -0.128
> print(pacf(temp, lag.max=15))

Partial autocorrelations of series ‘temp’, by lag

     1      2      3      4      5      6      7      8      9     10     11
 0.488  0.143  0.132  0.349 -0.250 -0.158  0.090  0.035  0.054 -0.062  0.034
    12     13     14     15
-0.316 -0.004 -0.099  0.014
```

可视化查验自相关 (ac)



可视化查验部分自相关 (pac)



滞后一阶自相关模型 AR(1)

- 令各个时间点的随机余数项为 $\epsilon_1, \epsilon_2, \ldots, \epsilon_T$，则 $\mathsf{E}(\epsilon_t) = 0$
- 假设是 AR(1) 模型，所以 ϵ_t 彼此不独立，但仅仅与滞后一阶相关，不与其他余数项相关。
- $\mathsf{E}(\epsilon_t | \epsilon_1, \epsilon_2, \ldots, \epsilon_{t-1}) = \alpha\epsilon_{t-1}$
- $\mathsf{E}(\epsilon_t | \epsilon_1, \epsilon_2, \ldots, \epsilon_{t-1})$ 即 ϵ_t 与其他项回归
- α 是自相关系数。

滞后一阶自相关模型 AR(1)

```
> M1 <- lm(temp ~ year, data=dat)
> res <- residuals(M1)
> alpha <- cor(res[2:n], res[1:(n-1)])
> alpha
[1] -0.03545959
```

-0.0355 即为 α 的估计值。

滞后一阶自相关模型 AR(1) 的分析步骤

- 将 Y_t 与 X_t 进行 OLS 回归
- 从余数估计一阶序相关系数 r_1
- 判定是否有序相关
- 判定是否序相关是否为滞后一阶 AR(1)
- 如果以上皆是，则进行变量滤波调整 (filtering transformation)
  - $Y_t^* = Y_t - r_1 Y_{t-1}$
  - $X_t^* = X_t - r_1 X_{t-1}$
- 将 Y_t^* 与 X_t^* 进行 OLS 回归得到 AR(1) 调整后回归估计值

滤波调整数学原理

- 使用 OLS 回归 AR(1) 数据

$$Y_t = \beta_0 + \beta_1 X_t + \epsilon_t, \qquad \mathsf{E}(\epsilon_t|\epsilon_1, \epsilon_2, \ldots, \epsilon_{t-1}) = \alpha\epsilon_{t-1}$$

- 进行滤波调整

$$\begin{aligned} Y_t^* &= Y_t - r_1 Y_{t-1} \\ X_t^* &= X_t - r_1 X_{t-1} \end{aligned}$$

- 数学证明

$$\begin{aligned} Y_t^* &= Y_t - r_1 Y_{t-1} = (\beta_0 + \beta_1 X_t + \epsilon_t) - \alpha(\beta_0 + \beta_1 X_{t-1} + \epsilon_{t-1}) \\ &= (\beta_0 - \alpha\beta_0) + \beta_1(X_t - \alpha X_{t-1}) + (\epsilon_t - \alpha\epsilon_{t-1}) \\ &= \gamma_0 + \beta_1 X_t^* + \epsilon_t^* \end{aligned}$$

滤波调整数学原理

- AR(1) 的序相关被过滤掉了：

$$\mathsf{E}(\epsilon_t^*|\epsilon_1^*, \epsilon_2^*, \ldots, \epsilon_{t-1}^*) = \mathsf{E}(\epsilon_t - \alpha\epsilon_{t-1}|\epsilon_1, \epsilon_2, \ldots, \epsilon_{t-1}) = 0$$

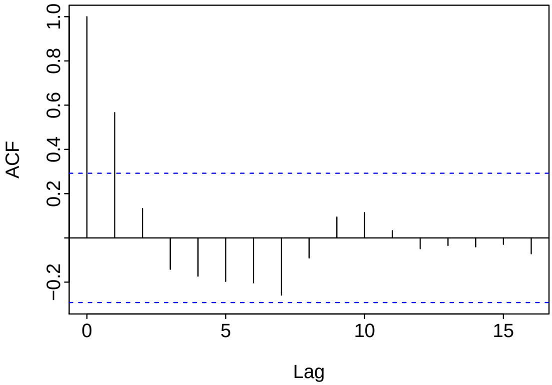
- 所以 OLS 的回归系数仍然是正确的。
- 因为 α 为未知数，所以用 r_1 代替。

```
> yearF <- with(dat, year - alpha*c(NA, year[1:(n-1)]))
> tempF <- with(dat, temp - alpha*c(NA, temp[1:(n-1)]))
> M2 <- lm(tempF ~ yearF)
> M2

Call:
lm(formula = tempF ~ yearF)

Coefficients:
(Intercept)        yearF
  -44.98370      0.03362

> M1

Call:
lm(formula = temp ~ year, data = dat)

Coefficients:
(Intercept)         year
  -38.45092      0.03112
```

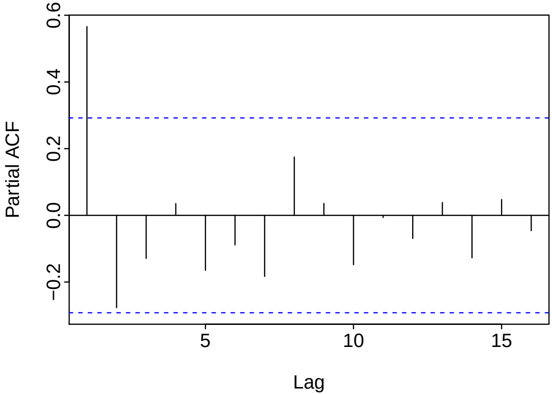
案例：民主得分

- 土耳其，1955-2000 数据
  - 因变量：民主得分 polity（-10 至 10）
  - 自变量：人均 GDP （gdp），开放程度 (open)
- 理论：人均 GDP 越高，贸易开放程度越大，越可能导致更高的民主得分。
- 某年民主得分可能影响接续一年的民主得分

```
> M3 <- lm(polity ~ gdp + open, data=turkey)
> summary(M3)

Call:
lm(formula = polity ~ gdp + open, data = turkey)

Residuals:
     Min       1Q   Median       3Q      Max
-11.4502  -0.2079   1.0731   2.1013   3.9164

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.9644526  3.2959305   0.899    0.374
gdp          0.0009986  0.0011713   0.853    0.399
open        -0.0482560  0.1030952  -0.468    0.642

Residual standard error: 3.988 on 42 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.03223,   Adjusted R-squared:  -0.01385
F-statistic: 0.6994 on 2 and 42 DF,  p-value: 0.5026

> res <- residuals(M3)
> cor(res[2:n], res[1:(n-1)])
[1] 0.5519347
```

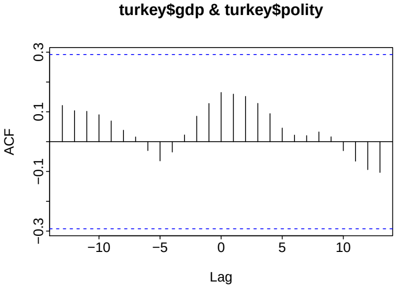


一阶似乎很显著



PAC 确认了数据是一阶自相关 AR(1)

CCF




```
> print(ccf(y=polity, x=gdp))

Autocorrelations of series 'X', by lag

   -13    -12    -11    -10     -9     -8     -7     -6     -5     -4     -3
 0.121  0.104  0.102  0.090  0.069  0.038  0.016 -0.030 -0.064 -0.035  0.023
    -2     -1      0      1      2      3      4      5      6      7      8
 0.085  0.128  0.165  0.159  0.152  0.128  0.094  0.046  0.022  0.020  0.033
     9     10     11     12     13
 0.016 -0.030 -0.065 -0.094 -0.103
```

```
> polF <- polity - alpha*c(NA, polity[1:(n-1)])
> gdpF <- gdp - alpha*c(NA, gdp[1:(n-1)])
> openF <- open - alpha*c(NA, open[1:(n-1)])
> M4 <- lm(polF ~ gdpF + openF)
> M4

Call:
lm(formula = polF ~ gdpF + openF)

Coefficients:
(Intercept)         gdpF        openF
   1.103907     0.001247    -0.073579

> M3

Call:
lm(formula = polity ~ gdp + open, data = turkey)

Coefficients:
(Intercept)          gdp         open
  2.9644526    0.0009986   -0.0482560
```