STATS 500 HW5

Minxuan Chen

2023-10-12

Table of contents

- Problem 1 . . . 1
  - (a) . . . 1
  - (b) . . . 2
- Problem 2 . . . 4
  - (1) . . . 4
  - (2) . . . 7
  - (3) . . . 7
  - (4) . . . 8
- Problem 3 . . . 8

Github repo: https://github.com/PKUniiiiice/STATS_500

Problem 1

(a)

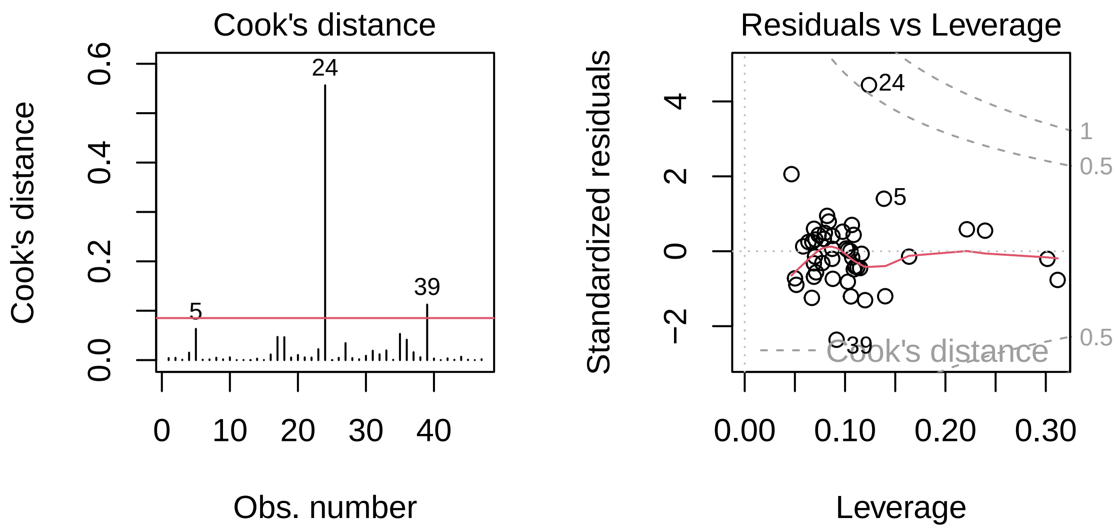
```r
library(faraway)
data(teengamb)

#teengamb$sex <- as.factor(teengamb$sex)
m1 <- lm(gamble ~ ., data=teengamb)

#we use cook's distance to check for influential points
round(cooks.distance(m1), digits=4)
```

```
     1      2      3      4      5      6      7      8      9     10     11
0.0042 0.0047 0.0008 0.0152 0.0633 0.0008 0.0015 0.0048 0.0015 0.0057 0.0001
    12     13     14     15     16     17     18     19     20     21     22
0.0003 0.0000 0.0030 0.0002 0.0115 0.0469 0.0465 0.0053 0.0104 0.0055 0.0052
    23     24     25     26     27     28     29     30     31     32     33
0.0222 0.5565 0.0000 0.0047 0.0344 0.0041 0.0017 0.0087 0.0190 0.0118 0.0196
    34     35     36     37     38     39     40     41     42     43     44
0.0007 0.0530 0.0414 0.0160 0.0059 0.1124 0.0032 0.0001 0.0035 0.0008 0.0069
    45     46     47
0.0009 0.0001 0.0019
```

```r
par(mfrow=c(1,2))
plot(m1, which=4)
abline(h=4/nrow(teengamb), col=2)
plot(m1, which=5)
```




```r
par(mfrow=c(1,1))
```

From the plots, we conclude that case No.24 and No.39 are influential points.

(b)

We use partial regression and residual plots to check the structure of the model.

Partial regression plots

```r
library(car)
```

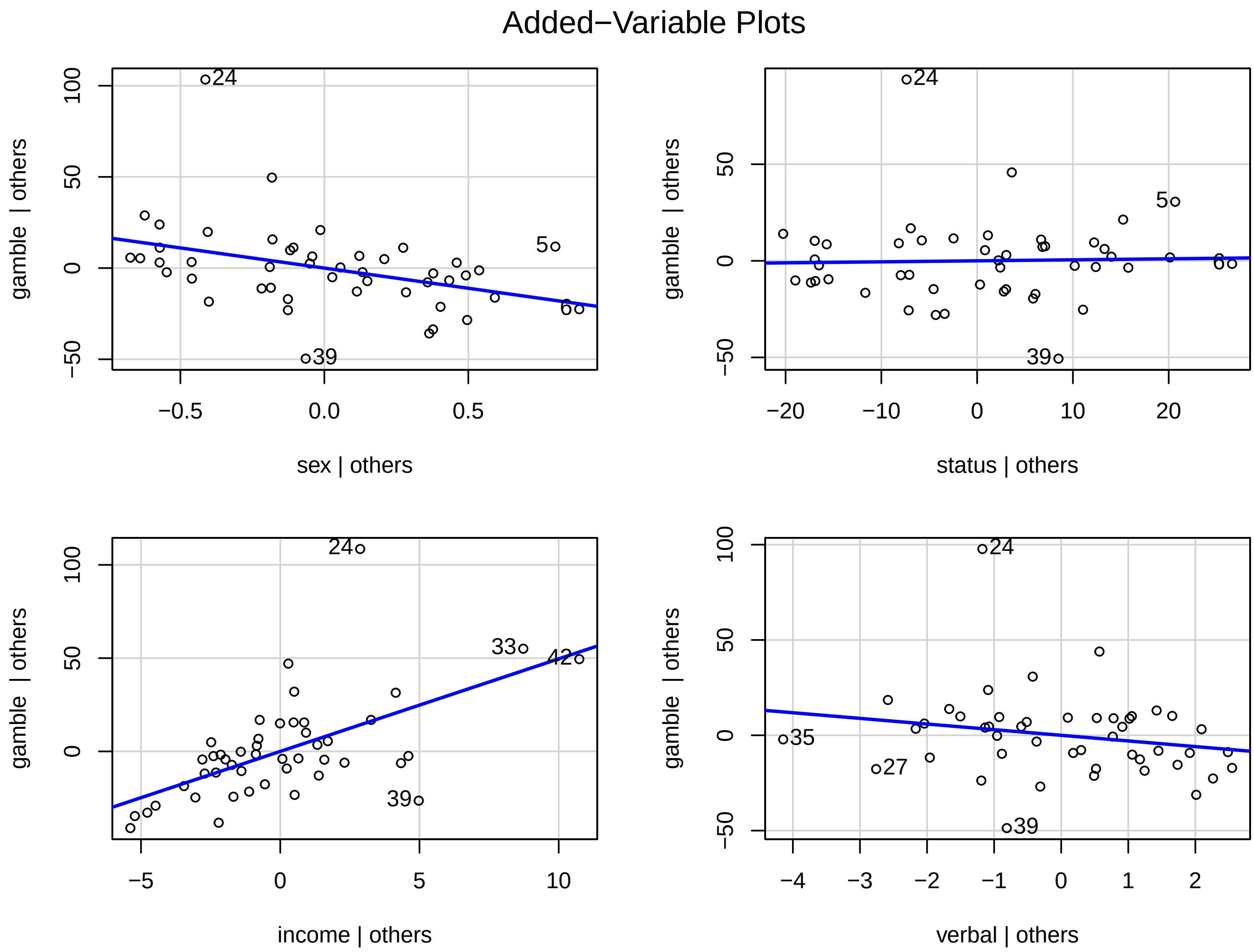
```
Loading required package: carData

Attaching package: 'car'

The following objects are masked from 'package:faraway':

    logit, vif
```

```r
avPlots(m1)
```



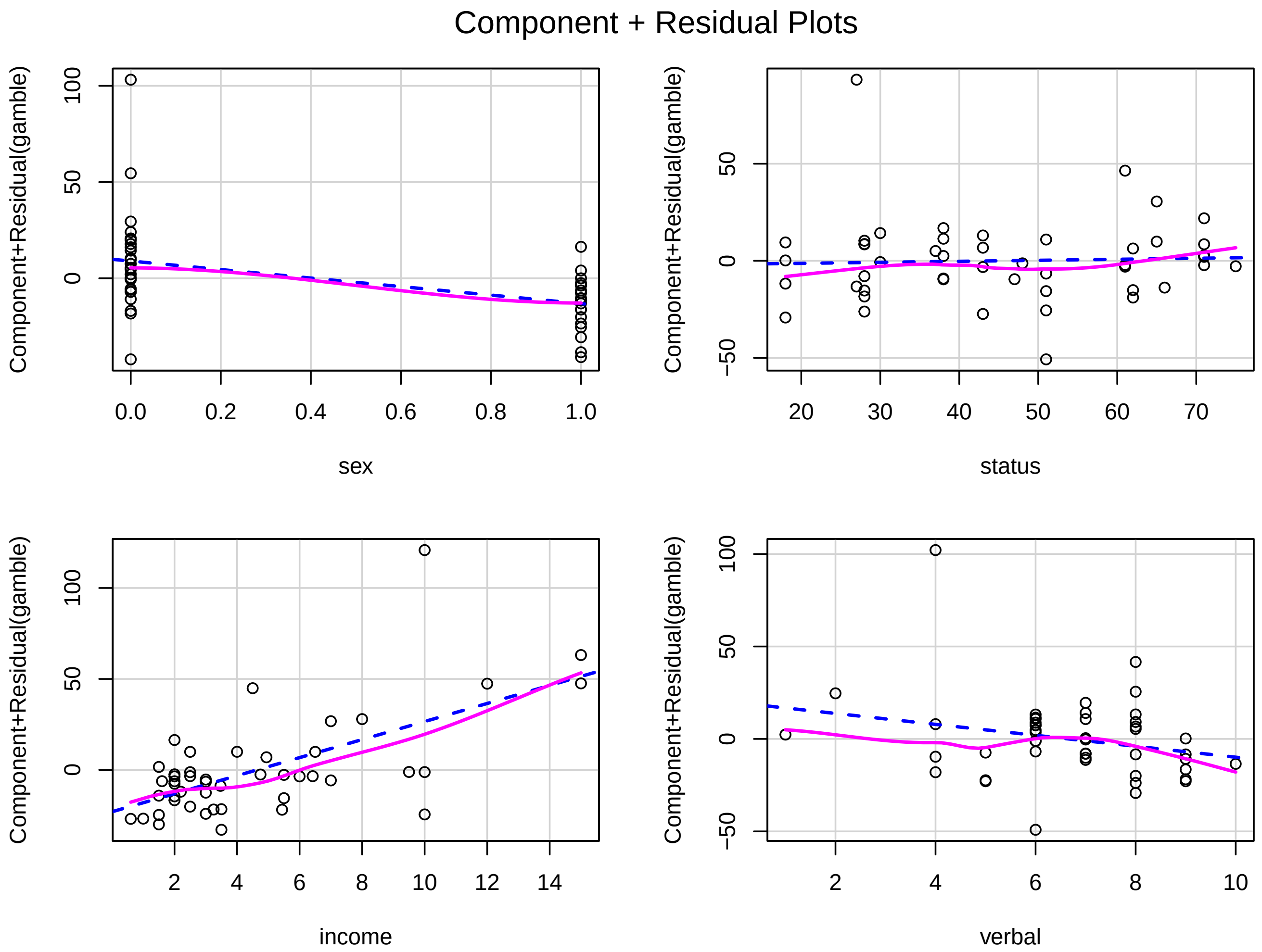
These four partial regression plots do not reveal any significant issues related to non linearity.

For outliers, each of these plots has identified two points with the largest residuals. For instance, in the `gamble-verbal` plot, cases No. 24 and No. 39 exhibit notably large-residual points and can be considered outliers.

For influential points, the plots have also identified two points with the most extreme horizontal values, signifying the large partial leverage. Some of them are merely high-leverage points (e.g., No. 34 and No. 42 in `gamble-income`), but others can be categorized as influential (e.g., No. 24 in both `gamble-sex` and `gamble-status`).

Partial residual plots

```
crPlots(m1)
```



The pink lines represent a smoother of the (component+residual) vs x_j. Our observations reveal that for the variables `sex` and `status`, there is no significant non linearity.

However, in `income` and `verbal`, the smoothers exhibit slight curvature. This suggests that it might be beneficial to consider adding squared terms for these variables to the model.

Problem 2

(1)

```
data("longley")

#original
m_ori <- lm(Employed~., data=longley)
summary(m_ori)
```

```
Call:
lm(formula = Employed ~ ., data = longley)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.41011 -0.15767 -0.02816  0.10155  0.45539 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -3.482e+03  8.904e+02  -3.911 0.003560 ** 
GNP.deflator  1.506e-02  8.492e-02   0.177 0.863141    
GNP          -3.582e-02  3.349e-02  -1.070 0.312681    
Unemployed   -2.020e-02  4.884e-03  -4.136 0.002535 ** 
Armed.Forces -1.033e-02  2.143e-03  -4.822 0.000944 ***
Population   -5.110e-02  2.261e-01  -0.226 0.826212    
Year          1.829e+00  4.555e-01   4.016 0.003037 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3049 on 9 degrees of freedom
Multiple R-squared:  0.9955,    Adjusted R-squared:  0.9925 
F-statistic: 330.3 on 6 and 9 DF,  p-value: 4.984e-10
```

```r
#normalized
m_nor <- lm(Employed~., data=data.frame(scale(longley)))
summary(m_nor)
```

```
Call:
lm(formula = Employed ~ ., data = data.frame(scale(longley)))

Residuals:
      Min        1Q    Median        3Q       Max 
-0.116776 -0.044896 -0.008019  0.028916  0.129669 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -1.752e-15  2.170e-02   0.000 1.000000    
GNP.deflator  4.628e-02  2.609e-01   0.177 0.863141    
GNP          -1.014e+00  9.479e-01  -1.070 0.312681    
Unemployed   -5.375e-01  1.300e-01  -4.136 0.002535 ** 
Armed.Forces -2.047e-01  4.246e-02  -4.822 0.000944 ***
Population   -1.012e-01  4.478e-01  -0.226 0.826212    
Year          2.480e+00  6.175e-01   4.016 0.003037 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.0868 on 9 degrees of freedom
Multiple R-squared:  0.9955,    Adjusted R-squared:  0.9925
F-statistic: 330.3 on 6 and 9 DF,  p-value: 4.984e-10
```

We rescale both x and y, consequently, t-statistic (except for the intercept), F-statistic, and R^2 remain unchanged, but

$$\hat{\sigma} \to \hat{\sigma}/sd(y)$$
$$\hat{\beta}_j \to sd(x) \cdot \hat{\beta}_j/sd(y)$$
$$\hat{\beta}_0 \to (\hat{\beta}_0 - \bar{y} + \Sigma\hat{\beta}_j\bar{x}_j)/sd(y)(= 0)$$

We can verify this

```r
temp <- scale(longley)
men <- attr(temp, "scaled:center")
sdd <- attr(temp, "scaled:scale")

#sigma^2
all.equal(summary(m_nor)$sigma,
          summary(m_ori)$sigma/sdd[7],
          check.attributes = FALSE)
```

```
[1] TRUE
```

```r
#beta_j
all.equal(coef(m_nor)[2:7],
          coef(m_ori)[2:7]*sdd[1:6]/sdd[7],
          check.attributes = FALSE)
```

```
[1] TRUE
```

```r
#beta_0
all.equal(coef(m_nor)[1],
          (coef(m_ori)[1]-men[7]+sum(coef(m_ori)[2:7]*men[1:6]))/sdd[7],
          check.attributes = FALSE)
```

```
[1] TRUE
```

Pros:
1. We can compare coefficients directly (removing magnitude difference between predictors) 2. It helps numerical stability (numerical problems in computing $(X^TX)^{-1}$) can be avoided or mitigated).

Cons:
1. Interpretation of coefficients is harder, since they are not in original unit.

(2)

We calculate condition number of $X^T X$.

```r
kappa(crossprod(model.matrix(m_ori)))
```

```
[1] 5.44871e+14
```

It's a really large value, so there will be large collinearity in this linear model.

(3)

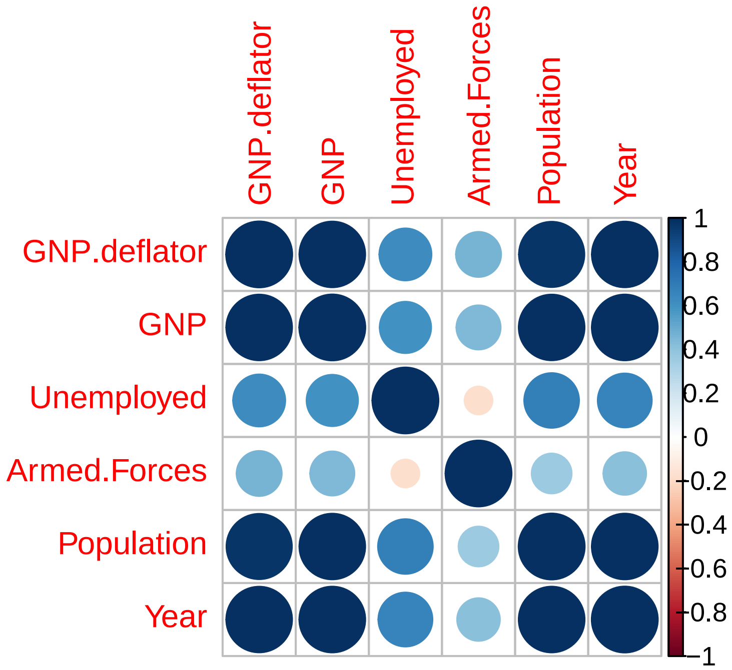
```r
cor(longley[1:6])
```

```
             GNP.deflator       GNP Unemployed Armed.Forces Population
GNP.deflator    1.0000000 0.9915892  0.6206334    0.4647442  0.9791634
GNP             0.9915892 1.0000000  0.6042609    0.4464368  0.9910901
Unemployed      0.6206334 0.6042609  1.0000000   -0.1774206  0.6865515
Armed.Forces    0.4647442 0.4464368 -0.1774206    1.0000000  0.3644163
Population      0.9791634 0.9910901  0.6865515    0.3644163  1.0000000
Year            0.9911492 0.9952735  0.6682566    0.4172451  0.9939528
                  Year
GNP.deflator 0.9911492
GNP          0.9952735
Unemployed   0.6682566
Armed.Forces 0.4172451
Population   0.9939528
Year         1.0000000
```

```r
library(corrplot)
```

```
corrplot 0.92 loaded
```

```r
corrplot(cor(longley[1:6]))
```



We observe strong correlation among `GNP.deflator`, `GNP`, `Population` and `Year`. This provides direct evidence of collinearity.

(4)

```
vif(m_ori)
```

```
GNP.deflator          GNP    Unemployed Armed.Forces    Population         Year
   135.53244   1788.51348      33.61889      3.58893     399.15102    758.98060
```

We observe that the VIFs of `GNP.deflator`, `GNP`, `Unemployed`, `Population` and `Year` are notably high. It shows that there exists obvious collinearity.

Problem 3

True relationship is

$$y_i^A = \beta_0 + \beta_1 x_i^A$$

There are errors in y_i and x_i, but here we only regard errors in y_i as random variable, that is

$$y_i^O = \beta_0 + \beta_1 x_i^O - \beta_1 \delta_i + \epsilon_i$$

We can regard $-\beta_1 \delta_i + \epsilon_i$ as the new error term,

$$y_i^O = \beta_0 + \beta_1 x_i^O + \tilde{\epsilon}_i$$

Despite certain assumptions of $\tilde{\epsilon}_i$ being violated compared to the classical linear regression model, we can still address this problem from a least squares perspective. In other words, we can apply the standard formula to estimate $\hat{\beta}_1$.

$$\begin{aligned}
\hat{\beta}_1 &= \frac{\sum(x_i^O - \bar{x}^O)(y_i^O - \bar{y}^O)}{\sum(x_i^O - \bar{x}^O)^2} \\
&= \frac{\sum(x_i^A - \bar{x}^A)(y_i^A - \bar{y}^A) + (x_i^A - \bar{x}^A)(\epsilon_i - \bar{\epsilon}) + (\delta_i - \bar{\delta})(y_i^A - \bar{y}^A) + (\delta_i - \bar{\delta})(\epsilon_i - \bar{\epsilon})}{\sum(x_i^A - \bar{x}^A)^2 + (\delta_i - \bar{\delta})^2 + 2(x_i^A - \bar{x}^A)(\delta_i - \bar{\delta})} \\
&= \frac{\text{numerator}}{n(\sigma_x^2 + \sigma_\delta^2 + 2\sigma_{x\delta})}
\end{aligned}$$

Then, We consider the numerator. Since the errors ϵ_i are the only random variables, using $y_i^O = y_i^A + \epsilon_i$ again and $E(\epsilon_i) = 0$,

$$\begin{aligned}
\mathbb{E}(\text{numerator}) &= \sum(x_i^A - \bar{x}^A)\mathbb{E}(y_i^A - \bar{y}^A) + (x_i^A - \bar{x}^A)\mathbb{E}(\epsilon_i - \bar{\epsilon}) + (\delta_i - \bar{\delta})\mathbb{E}(y_i^A - \bar{y}^A) + (\delta_i - \bar{\delta})\mathbb{E}(\epsilon_i - \bar{\epsilon}) \\
&= \sum(x_i^A - \bar{x}^A)\mathbb{E}(y_i^A - \bar{y}^A) + (\delta_i - \bar{\delta})\mathbb{E}(y_i^A - \bar{y}^A) \\
&= \sum(x_i^A - \bar{x}^A)\beta_1(x_i^A - \bar{x}^A) + (\delta_i - \bar{\delta})\beta_1(x_i^A - \bar{x}^A) \\
&= \beta_1 \sum(x_i^A - \bar{x}^A)^2 + (\delta_i - \bar{\delta})(x_i^A - \bar{x}^A) \\
&= \beta_1 \cdot n(\sigma_x^2 + \sigma_{x\delta})
\end{aligned}$$

Therefore

$$\mathbb{E}(\hat{\beta}_1) = \frac{\mathbb{E}(\text{numerator})}{n(\sigma_x^2 + \sigma_\delta^2 + 2\sigma_{x\delta})} = \beta_1 \frac{\sigma_x^2 + \sigma_{x\delta}}{\sigma_x^2 + \sigma_\delta^2 + 2\sigma_{x\delta}}$$