Lecture 20

Math 6040/7260 Linear Models

Dr. Xiang Ji @ Tulane University

March 5, 2021

Contents

Acknowledgement

Dr. Hua Zhou's slides

GA 2000 US Presidential Election Data

The `gavote` data contains the voting data of Georgia (GA) in the 2000 presidential election. It is available as a dataframe.

```
# equivalent to head(gavote, 10)
gavote %>% head(10)
```

```
##          equip   econ perAA rural    atlanta gore  bush other votes ballots
## APPLING  LEVER   poor 0.182 rural notAtlanta 2093  3940    66  6099    6617
## ATKINSON LEVER   poor 0.230 rural notAtlanta  821  1228    22  2071    2149
## BACON    LEVER   poor 0.131 rural notAtlanta  956  2010    29  2995    3347
## BAKER    OS-CC   poor 0.476 rural notAtlanta  893   615    11  1519    1607
## BALDWIN  LEVER middle 0.359 rural notAtlanta 5893  6041   192 12126   12785
## BANKS    LEVER middle 0.024 rural notAtlanta 1220  3202   111  4533    4773
## BARROW   OS-CC middle 0.079 urban notAtlanta 3657  7925   520 12102   12522
## BARTOW   OS-PC middle 0.079 urban    Atlanta 7508 14720   552 22780   23735
## BEN.HILL OS-PC   poor 0.282 rural notAtlanta 2234  2381    46  4661    5741
## BERRIEN  OS-CC   poor 0.107 rural notAtlanta 1640  2718    52  4410    4475
```

We convert it into a tibble for easy handling by tidyverse.

```
gavote <- gavote %>%
  as_tibble(rownames = "county") %>%
  print(width = Inf)
```

```
## # A tibble: 159 x 11
##    county   equip econ   perAA rural atlanta    gore  bush other votes ballots
##    <chr>    <fct> <fct>  <dbl> <fct> <fct>     <int> <int> <int> <int>   <int>
##  1 APPLING  LEVER poor   0.182 rural notAtlanta  2093  3940    66  6099    6617
##  2 ATKINSON LEVER poor   0.23  rural notAtlanta   821  1228    22  2071    2149
##  3 BACON    LEVER poor   0.131 rural notAtlanta   956  2010    29  2995    3347
##  4 BAKER    OS-CC poor   0.476 rural notAtlanta   893   615    11  1519    1607
##  5 BALDWIN  LEVER middle 0.359 rural notAtlanta  5893  6041   192 12126   12785
##  6 BANKS    LEVER middle 0.024 rural notAtlanta  1220  3202   111  4533    4773
##  7 BARROW   OS-CC middle 0.079 urban notAtlanta  3657  7925   520 12102   12522
##  8 BARTOW   OS-PC middle 0.079 urban Atlanta     7508 14720   552 22780   23735
##  9 BEN.HILL OS-PC poor   0.282 rural notAtlanta  2234  2381    46  4661    5741
## 10 BERRIEN  OS-CC poor   0.107 rural notAtlanta  1640  2718    52  4410    4475
## # ... with 149 more rows
```

- Each row is a county in GA.
- The number of votes, `votes`, can be smaller than the number of ballots, `ballots`, because a vote is not recorded if (1) the person fails to vote for President, (2) votes for more than one candidate, or (3) the equipment fails to record the vote.
- We are interested in the `undercount`, which is defined as `(ballots - votes) / ballots`. Does it depend on the type of voting machine `equip`, economy `econ`, percentage of African Americans `perAA`, whether the county is rural or urban `rural`, or whether the county is part of Atlanta metropolitan area `atlanta`.

  Let's create a new variable `undercount`

```
gavote <- gavote %>%
  mutate(undercount = (ballots - votes) / ballots) %>%
  print(width = Inf)
```

```
## # A tibble: 159 x 12
##    county   equip econ   perAA rural atlanta    gore  bush other votes ballots
##    <chr>    <fct> <fct>  <dbl> <fct> <fct>     <int> <int> <int> <int>   <int>
##  1 APPLING  LEVER poor   0.182 rural notAtlanta  2093  3940    66  6099    6617
##  2 ATKINSON LEVER poor   0.23  rural notAtlanta   821  1228    22  2071    2149
##  3 BACON    LEVER poor   0.131 rural notAtlanta   956  2010    29  2995    3347
##  4 BAKER    OS-CC poor   0.476 rural notAtlanta   893   615    11  1519    1607
##  5 BALDWIN  LEVER middle 0.359 rural notAtlanta  5893  6041   192 12126   12785
##  6 BANKS    LEVER middle 0.024 rural notAtlanta  1220  3202   111  4533    4773
##  7 BARROW   OS-CC middle 0.079 urban notAtlanta  3657  7925   520 12102   12522
##  8 BARTOW   OS-PC middle 0.079 urban Atlanta     7508 14720   552 22780   23735
##  9 BEN.HILL OS-PC poor   0.282 rural notAtlanta  2234  2381    46  4661    5741
## 10 BERRIEN  OS-CC poor   0.107 rural notAtlanta  1640  2718    52  4410    4475
##    undercount
##         <dbl>
##  1     0.0783
##  2     0.0363
##  3     0.105
##  4     0.0548
##  5     0.0515
##  6     0.0503
##  7     0.0335
##  8     0.0402
##  9     0.188
## 10     0.0145
```

```
## # ... with 149 more rows
```

- For factor `rural`, we found the variable name is same as one level in this factor. To avoid confusion, we rename it to `usage`.

  We also want to standardize the counts `gore` and `bush` according to the total `votes`.

```r
(gavote <- gavote %>%
  rename(usage = rural) %>%
  mutate(pergore = gore / votes, perbush = bush / votes)) %>%
  print(width = Inf)
```

```
## # A tibble: 159 x 14
##    county   equip econ   perAA usage atlanta    gore  bush other votes ballots
##    <chr>    <fct> <fct>  <dbl> <fct> <fct>     <int> <int> <int> <int>   <int>
##  1 APPLING  LEVER poor   0.182 rural notAtlanta  2093  3940    66  6099    6617
##  2 ATKINSON LEVER poor   0.23  rural notAtlanta   821  1228    22  2071    2149
##  3 BACON    LEVER poor   0.131 rural notAtlanta   956  2010    29  2995    3347
##  4 BAKER    OS-CC poor   0.476 rural notAtlanta   893   615    11  1519    1607
##  5 BALDWIN  LEVER middle 0.359 rural notAtlanta  5893  6041   192 12126   12785
##  6 BANKS    LEVER middle 0.024 rural notAtlanta  1220  3202   111  4533    4773
##  7 BARROW   OS-CC middle 0.079 urban notAtlanta  3657  7925   520 12102   12522
##  8 BARTOW   OS-PC middle 0.079 urban Atlanta     7508 14720   552 22780   23735
##  9 BEN.HILL OS-PC poor   0.282 rural notAtlanta  2234  2381    46  4661    5741
## 10 BERRIEN  OS-CC poor   0.107 rural notAtlanta  1640  2718    52  4410    4475
##    undercount pergore perbush
##         <dbl>   <dbl>   <dbl>
##  1     0.0783   0.343   0.646
##  2     0.0363   0.396   0.593
##  3     0.105    0.319   0.671
##  4     0.0548   0.588   0.405
##  5     0.0515   0.486   0.498
##  6     0.0503   0.269   0.706
##  7     0.0335   0.302   0.655
##  8     0.0402   0.330   0.646
##  9     0.188    0.479   0.511
## 10     0.0145   0.372   0.616
## # ... with 149 more rows
```

A model with two quantitative predictors

- We start with a linear model with just two predictors: percentage of Gore votes, `pergore`, and percentage of African Americans, `perAA`.

$$\text{undercount} = \beta_0 + \beta_1 \cdot \text{pergore} + \beta_2 \cdot \text{perAA} + \epsilon.$$

`lm`

```r
(lmod <- lm(undercount ~ pergore + perAA, gavote))
```

```
##
## Call:
## lm(formula = undercount ~ pergore + perAA, data = gavote)
##
## Coefficients:
## (Intercept)      pergore        perAA
```

```
##     0.03238       0.01098       0.02853
```

- The **regression coefficient** $\widehat{\boldsymbol{\beta}}$ can be retrieved by

```r
# same lmod$coefficients
coef(lmod)
```

```
## (Intercept)     pergore       perAA
##  0.03237600  0.01097872  0.02853314
```

- interpreting the partial regression coefficients:
  - **Geometric interpretation**: The partial regression coefficient β_j associated with the predictor x_j is the slope of the regression plane in the x_j direction. Imagine taking a "slice" of the regression plane. In the terminology of calculus, β_j is also the partial derivative of the regression plane with respect to the predictor x_j.
  - **Verbal interpretation**: The partial regression coefficient β_j associated with predictor x_j is the slope of the linear association between y and x_j *while controlling for the other predictors in the model* (i.e., holding them constant).
- The **fitted values** or **predicted values** are

$$\widehat{\mathbf{y}} = \mathbf{X}\widehat{\boldsymbol{\beta}}$$

```r
# same as lmod$fitted.values
predict(lmod) %>% head()
```

```
##          1          2          3          4          5          6
## 0.04133661 0.04329088 0.03961823 0.05241202 0.04795484 0.03601558
```

and the **residuals** are

$$\widehat{\boldsymbol{\epsilon}} = \mathbf{y} - \widehat{\mathbf{y}} = \mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}.$$

```r
# same as lmod$residuals
residuals(lmod) %>% head()
```

```
##            1            2            3            4            5            6
##  0.036946603 -0.006994927  0.065550577  0.002348407  0.003589940  0.014267264
```

- The **residual sum of squares** (RSS), also called **deviance**, is $\|\widehat{\boldsymbol{\epsilon}}\|^2$.

```r
deviance(lmod)
```

```
## [1] 0.09324918
```

- The **degree of freedom** of a linear model is $n - p$.

```r
nrow(gavote) - length(coef(lmod))
```

```
## [1] 156
```

```r
df.residual(lmod)
```

```
## [1] 156
```

`summary`

- The `summary` command computes some more regression quantities.

```r
(lmodsum <- summary(lmod))
```

```
##
## Call:
## lm(formula = undercount ~ pergore + perAA, data = gavote)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.046013 -0.014995 -0.003539  0.011784  0.142436
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.03238    0.01276   2.537   0.0122 *
## pergore      0.01098    0.04692   0.234   0.8153
## perAA        0.02853    0.03074   0.928   0.3547
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02445 on 156 degrees of freedom
## Multiple R-squared:  0.05309,    Adjusted R-squared:  0.04095
## F-statistic: 4.373 on 2 and 156 DF,  p-value: 0.01419
```

- An unbiased estimate of the error variance σ^2 is

$$\widehat{\sigma} = \sqrt{\frac{\text{RSS}}{\text{df}}}$$

```r
sqrt(deviance(lmod) / df.residual(lmod))
```

```
## [1] 0.02444895
```

```r
lmodsum$sigma
```

```
## [1] 0.02444895
```

- A commonly used goodness of fit mesure is R^2, or **coefficient of determination** or **percentage of variance explained**

$$R^2 = 1 - \frac{\sum_i (y_i - \widehat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} = 1 - \frac{\text{RSS}}{\text{TSS}},$$

  where $\text{TSS} = \sum_i (y_i - \bar{y})^2$ is the **total sum of squares**.

```r
lmodsum$r.squared
```

```
## [1] 0.05308861
```

An R^2 of about 5% indicates the model has a poor fit. R^2 can also be interpreted as the (squared) correlation between the predicted values and the response

```r
cor(predict(lmod), gavote$undercount)^2
```

```
## [1] 0.05308861
```

A model with both quantitative and qualitative predictors

- Now we also want to include factors `equip` and `usage`, and interaction between `pergore` and `usage` into the model.
- Before that, we first center the `pergore` and `perAA` variables.

```r
gavote <- gavote %>%
  mutate(cpergore = pergore - mean(pergore), cperAA = perAA - mean(perAA)) %>%
  print(width = Inf)
```

```
## # A tibble: 159 x 16
##    county   equip econ   perAA usage atlanta     gore  bush other votes ballots
##    <chr>    <fct> <fct>  <dbl> <fct> <fct>      <int> <int> <int> <int>   <int>
##  1 APPLING  LEVER poor   0.182 rural notAtlanta  2093  3940    66  6099    6617
##  2 ATKINSON LEVER poor   0.23  rural notAtlanta   821  1228    22  2071    2149
##  3 BACON    LEVER poor   0.131 rural notAtlanta   956  2010    29  2995    3347
##  4 BAKER    OS-CC poor   0.476 rural notAtlanta   893   615    11  1519    1607
##  5 BALDWIN  LEVER middle 0.359 rural notAtlanta  5893  6041   192 12126   12785
##  6 BANKS    LEVER middle 0.024 rural notAtlanta  1220  3202   111  4533    4773
##  7 BARROW   OS-CC middle 0.079 urban notAtlanta  3657  7925   520 12102   12522
##  8 BARTOW   OS-PC middle 0.079 urban Atlanta     7508 14720   552 22780   23735
##  9 BEN.HILL OS-PC poor   0.282 rural notAtlanta  2234  2381    46  4661    5741
## 10 BERRIEN  OS-CC poor   0.107 rural notAtlanta  1640  2718    52  4410    4475
##    undercount pergore perbush cpergore  cperAA
##         <dbl>   <dbl>   <dbl>    <dbl>   <dbl>
##  1     0.0783   0.343   0.646  -0.0652 -0.0610
##  2     0.0363   0.396   0.593  -0.0119 -0.0130
##  3     0.105    0.319   0.671  -0.0891 -0.112
##  4     0.0548   0.588   0.405   0.180   0.233
##  5     0.0515   0.486   0.498   0.0777  0.116
##  6     0.0503   0.269   0.706  -0.139  -0.219
##  7     0.0335   0.302   0.655  -0.106  -0.164
##  8     0.0402   0.330   0.646  -0.0787 -0.164
##  9     0.188    0.479   0.511   0.0710  0.0390
## 10     0.0145   0.372   0.616  -0.0364 -0.136
## # ... with 149 more rows
```

- Fit the new model with `lm`. We note the model respects the hierarchy. That is the main effects are automatically added to the model in presense of their interaction. **Question**: how to specify a formula involving just an interaction term but not their main effect?

```
lmodi <- lm(undercount ~ cperAA + cpergore * usage + equip, gavote)
summary(lmodi)
```

```
##
## Call:
## lm(formula = undercount ~ cperAA + cpergore * usage + equip,
##     data = gavote)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.059530 -0.012904 -0.002180  0.009013  0.127496
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.043297   0.002839  15.253  < 2e-16 ***
## cperAA               0.028264   0.031092   0.909   0.3648
## cpergore             0.008237   0.051156   0.161   0.8723
## usageurban          -0.018637   0.004648  -4.009 9.56e-05 ***
## equipOS-CC           0.006482   0.004680   1.385   0.1681
## equipOS-PC           0.015640   0.005827   2.684   0.0081 **
## equipPAPER          -0.009092   0.016926  -0.537   0.5920
## equipPUNCH           0.014150   0.006783   2.086   0.0387 *
## cpergore:usageurban -0.008799   0.038716  -0.227   0.8205
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02335 on 150 degrees of freedom
## Multiple R-squared:  0.1696, Adjusted R-squared:  0.1253
## F-statistic: 3.829 on 8 and 150 DF,  p-value: 0.0004001
```

The `gtsummary` package offers a more sensible diplay of regression results.

```r
library(gtsummary)
lmodi %>%
  tbl_regression() %>%
  bold_labels() %>%
  bold_p(t = 0.05)
```

```
## Table printed with `knitr::kable()`, not {gt}. Learn why at
## http://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in code chunk header.
```

Characteristic	Beta	95% CI	p-value
cperAA	0.03	-0.03, 0.09	0.4
cpergore	0.01	-0.09, 0.11	0.9
usage			
rural			
urban	-0.02	-0.03, -0.01	**<0.001**
equip			
LEVER			
OS-CC	0.01	0.00, 0.02	0.2
OS-PC	0.02	0.00, 0.03	**0.008**
PAPER	-0.01	-0.04, 0.02	0.6
PUNCH	0.01	0.00, 0.03	**0.039**
cpergore * usage			
cpergore * urban	-0.01	-0.09, 0.07	0.8

- From the output, we learn that the model is

$$\begin{aligned}\text{undercount} &= \beta_0 + \beta_1 \cdot \text{cperAA} + \beta_2 \cdot \text{cpergore} + \beta_3 \cdot \text{usageurban} + \beta_4 \cdot \text{equipOS-CC} + \beta_5 \cdot \text{equipOS-PC} \\ &\quad + \beta_6 \cdot \text{equipPAPER} + \beta_7 \cdot \text{equipPUNCH} + \beta_8 \cdot \text{cpergore:usageurban} + \epsilon.\end{aligned}$$

- **Exercise**: Explain how the variables in `gavote` are translated into $\mathbf{X}$.

```r
gavote %>%
  select(cperAA, cpergore, equip, usage) %>%
  head(10)
```

```
## # A tibble: 10 x 4
##     cperAA cpergore equip usage
##      <dbl>    <dbl> <fct> <fct>
##  1 -0.0610  -0.0652 LEVER rural
##  2 -0.0130  -0.0119 LEVER rural
##  3 -0.112   -0.0891 LEVER rural
##  4  0.233    0.180  OS-CC rural
##  5  0.116    0.0777 LEVER rural
##  6 -0.219   -0.139  LEVER rural
##  7 -0.164   -0.106  OS-CC urban
##  8 -0.164   -0.0787 OS-PC urban
```

```
## 9  0.0390   0.0710 OS-PC rural
## 10 -0.136   -0.0364 OS-CC rural
```

```r
model.matrix(lmodi) %>% head(10)
```

```
##    (Intercept)      cperAA    cpergore usageurban equipOS-CC equipOS-PC
## 1            1 -0.06098113 -0.06515076          0          0          0
## 2            1 -0.01298113 -0.01189493          0          0          0
## 3            1 -0.11198113 -0.08912311          0          0          0
## 4            1  0.23301887  0.17956499          0          1          0
## 5            1  0.11601887  0.07765876          0          0          0
## 6            1 -0.21898113 -0.13918434          0          0          0
## 7            1 -0.16398113 -0.10614032          1          1          0
## 8            1 -0.16398113 -0.07873442          1          0          1
## 9            1  0.03901887  0.07097452          0          0          1
## 10           1 -0.13598113 -0.03643969          0          1          0
##    equipPAPER equipPUNCH cpergore:usageurban
## 1           0          0          0.00000000
## 2           0          0          0.00000000
## 3           0          0          0.00000000
## 4           0          0          0.00000000
## 5           0          0          0.00000000
## 6           0          0          0.00000000
## 7           0          0         -0.10614032
## 8           0          0         -0.07873442
## 9           0          0          0.00000000
## 10          0          0          0.00000000
```

- **Exerciese**: Interpret regression coefficient.
  - How do we interpret $\widehat{\beta}_0 = 0.043$?
  - How do we interpret $\widehat{\beta}_{\text{cperAA}} = 0.0283$?
  - How do we interpret $\widehat{\beta}_{\text{equipOS-PC}} = 0.016$?
  - How do we interpret $\widehat{\beta}_{\text{usageurban}} = -0.019$?
  - How do we interpret $\widehat{\beta}_{\text{cpergore:usageurban}} = -0.009$?

Hypothesis testing

- We want to formally compare the two linear models.
  - A larger model Ω with $p = 9$ parameters and
  - a smaller model ω with $q = 3$ parameters.
- The F-test compares the F-statistic

$$F = \frac{(\text{RSS}_\omega - \text{RSS}_\Omega)/(p-q)}{\text{RSS}_\Omega/(n-p)}$$

to its null distribution $F_{p-q,n-p}$. The small p-value 0.0028 indicates we should reject the null model ω.

```r
anova(lmod, lmodi)
```

```
## Analysis of Variance Table
##
## Model 1: undercount ~ pergore + perAA
## Model 2: undercount ~ cperAA + cpergore * usage + equip
##   Res.Df      RSS Df Sum of Sq      F   Pr(>F)
```

```
## 1    156 0.093249
## 2    150 0.081775  6  0.011474 3.5077 0.002823 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- We can carry out a similar F-test for each predictor in a model using the `drop1` function. The nice thing is that the factors such as `equip` and `cpergore * usage` are droped as a group.

```r
drop1(lmodi, test = "F")
```

```
## Single term deletions
##
## Model:
## undercount ~ cperAA + cpergore * usage + equip
##                Df Sum of Sq      RSS     AIC F value  Pr(>F)
## <none>                      0.081775 -1186.1
## cperAA          1 0.0004505 0.082226 -1187.2  0.8264 0.36479
## equip           4 0.0054438 0.087219 -1183.8  2.4964 0.04521 *
## cpergore:usage  1 0.0000282 0.081804 -1188.0  0.0517 0.82051
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We also see F-test for quantitative variables, e.g., `cperAA`, conincides with the t-test reported by the `lm` function. **Question**: why `drop1` function does not drop predictors `cpergore` and `usage`?

Confidence intervals

- Confidence intervals for individual parameters can be construced based on their null distribution

$$\frac{\widehat{\beta}_j}{\text{se}(\widehat{\beta}_j)} \sim t_{n-p}.$$

  That is a $(1-\alpha)$ confidence interval is

$$\widehat{\beta}_j \pm t_{n-p}^{(\alpha/2)} \text{se}(\widehat{\beta}_j).$$

```r
confint(lmodi)
```

```
##                             2.5 %       97.5 %
## (Intercept)          0.0376884415  0.048906189
## cperAA              -0.0331710614  0.089699222
## cpergore            -0.0928429315  0.109316616
## usageurban          -0.0278208965 -0.009452268
## equipOS-CC          -0.0027646444  0.015729555
## equipOS-PC           0.0041252334  0.027153973
## equipPAPER          -0.0425368415  0.024352767
## equipPUNCH           0.0007477196  0.027551488
## cpergore:usageurban -0.0852990903  0.067700182
```

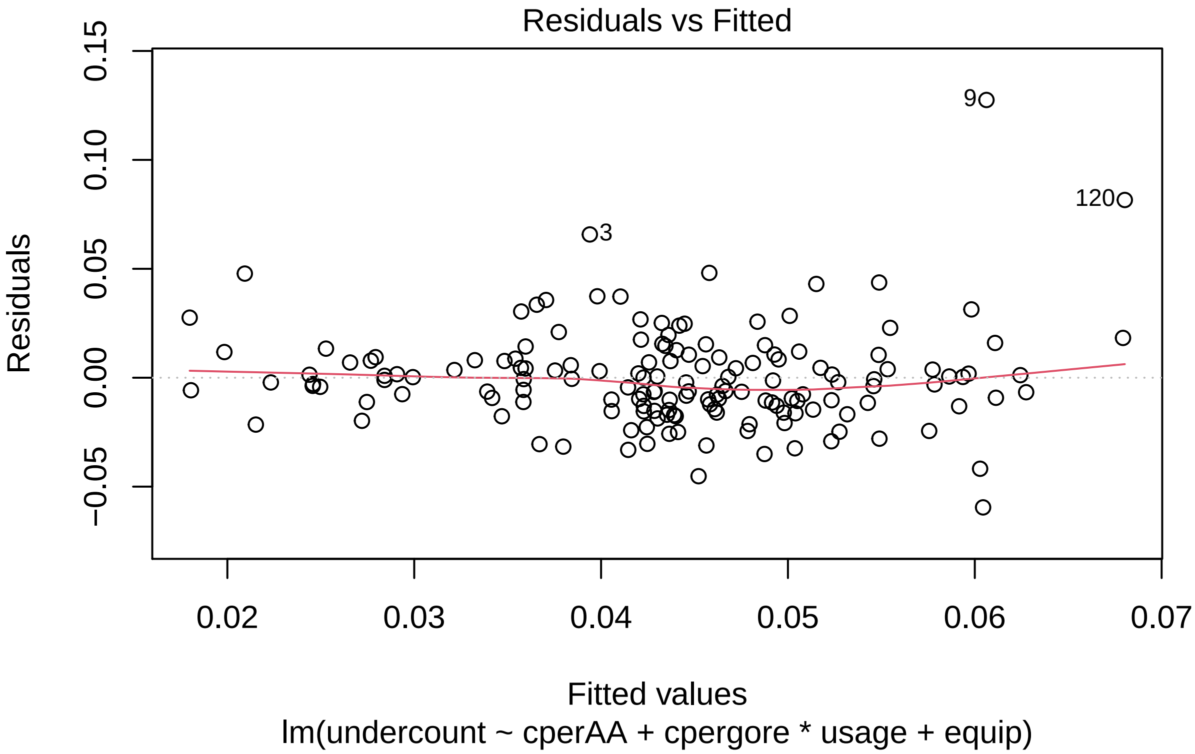
Diagnostics

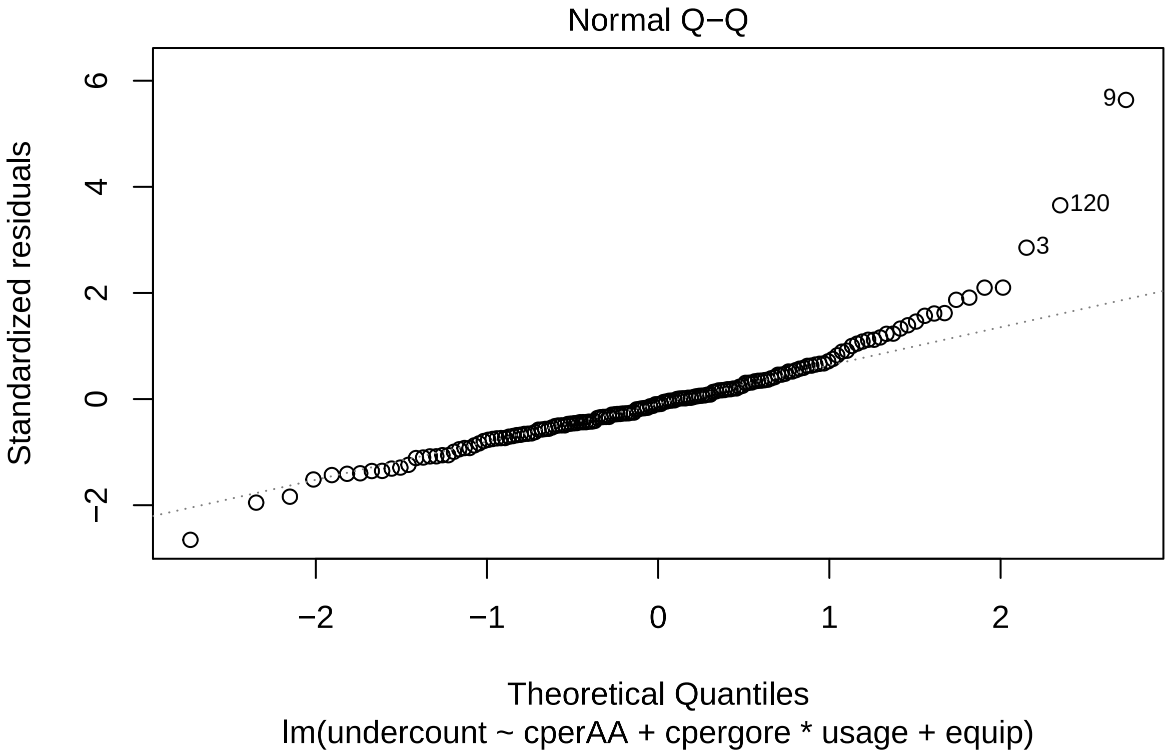
- Typical assumptions of linear models are
  1. $\mathbb{E}(\mathbf{Y}) = \mathbf{X}\boldsymbol{\beta}$, or equivalently, $\mathbb{E}(\boldsymbol{\epsilon}) = \mathbf{0}$. That is we have included all the right variables and Y depends on them linearly.

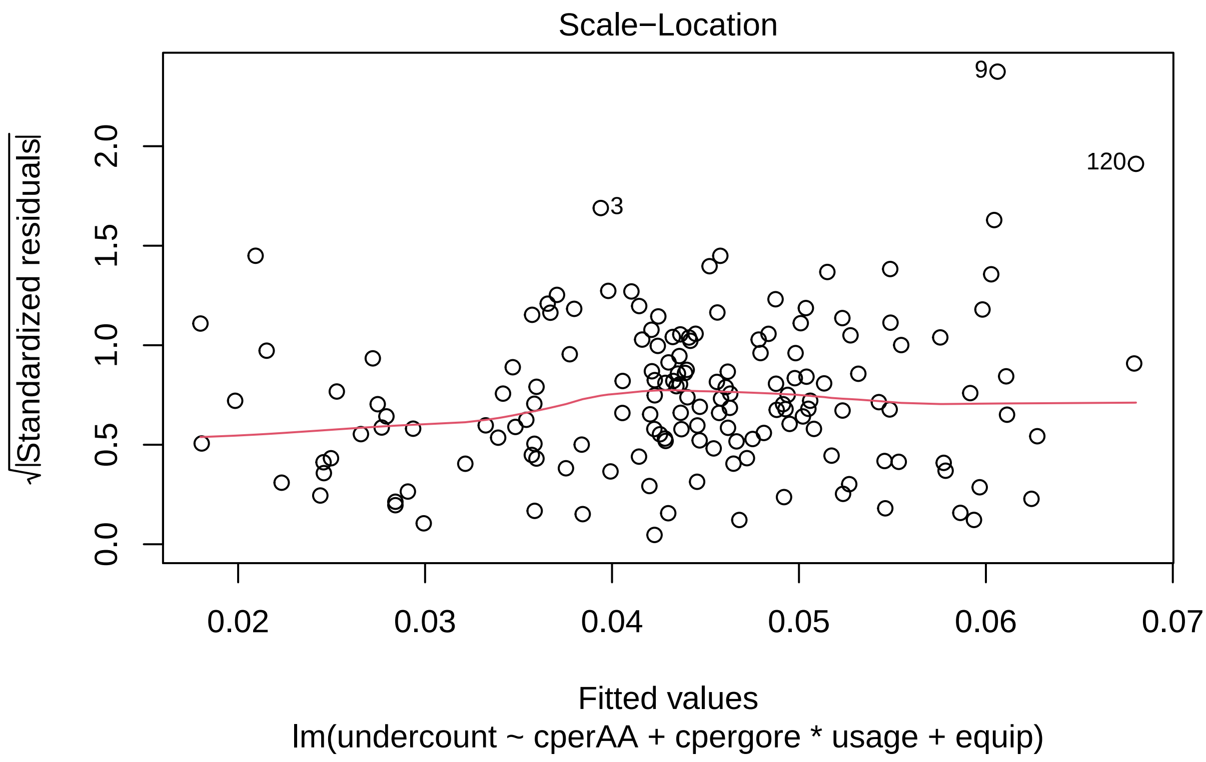
  2. Errors ϵ_i are independent and normally distributed with common variance σ^2. That is $\widehat{\boldsymbol{\epsilon}} \sim \text{N}(\mathbf{0}, \sigma_0^2 \mathbf{I}_n)$.
     We'd like to check these assumptions using graphical or numerical approaches.

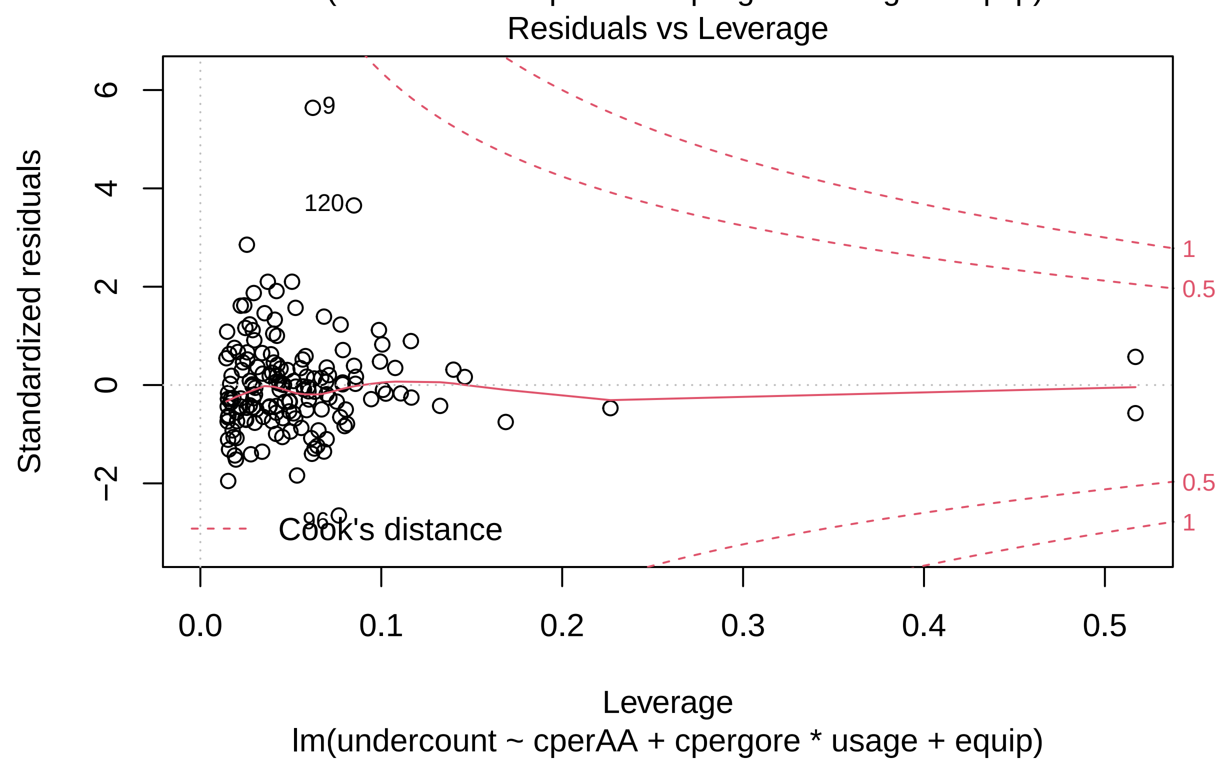
- Four commonly used diagnostic plots can be conveniently obtained by `plot` function.

```
plot(lmodi)
```









- The **residual-fitted value plot** is useful for checking the linearity and constant variance assumptions.
- The **scale-location plot** plots $\sqrt{|\hat{\epsilon}_i|}$ vs fitted values and serves similar purpose as the residual-fitted value plot.
- The **QQ plot** checks for the normality assumption. It plots sorted residuals vs the theoretical quantiles from a standard normal distribution $\Phi^{-1}\left(\frac{i}{n+1}\right)$, $i = 1, \ldots, n$.

- **Residual-leverage plot**. The fitted values are

$$\widehat{\mathbf{y}} = \mathbf{X}\widehat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \mathbf{H}\mathbf{y}.$$

  The diagonal entries of the **hat matrix**, $h_i = H_{ii}$, are called **leverages**. For example,

$$\text{Var}(\widehat{\boldsymbol{\epsilon}}) = \text{Var}(\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}) = \text{Var}[(\mathbf{I} - \mathbf{H})\mathbf{Y}] = (\mathbf{I} - \mathbf{H})\text{Var}(\mathbf{Y})(\mathbf{I} - \mathbf{H}) = \sigma^2(\mathbf{I} - \mathbf{H}).$$

  If h_i is large, then $\text{var}(\widehat{\epsilon}_i) = \sigma^2(1 - h_i)$ is small. The fit is "forced" to be close to y_i. Points on the boundary of the predictor space have the most leverage.

- The **Cook distance** is a popular influence diagnostic

$$D_i = \frac{(\widehat{y}_i - \widehat{y}_{(i)})^T(\widehat{y}_i - \widehat{y}_{(i)})}{p\widehat{\sigma}^2} = \frac{1}{p}r_i^2\frac{h_i}{1 - h_i},$$

  where r_i are the standardized residuals and $\widehat{y}_{(i)}$ are the predicted values if the i-th observation is dropped from data. A large residual combined with a large leverage results in a larger Cook statistic. In this sense it is an **influential point**.

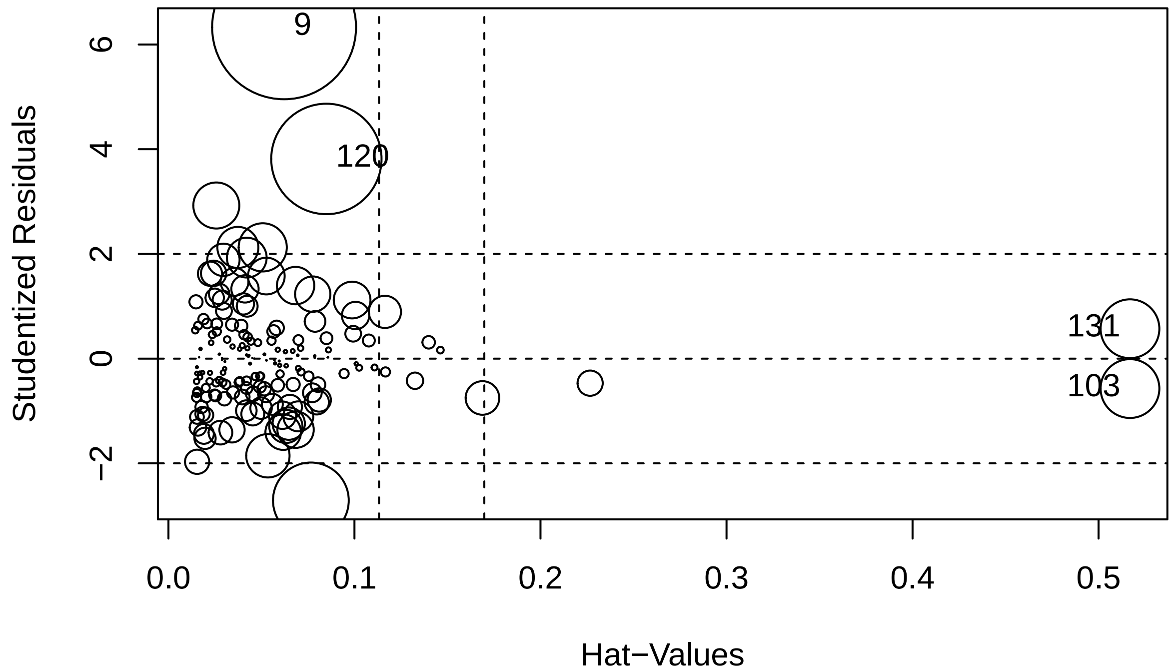
  Let's display counties with Cook distance > 0.1. These are those two counties with unusual large `undercount`.

```
gavote %>%
  mutate(cook = cooks.distance(lmodi)) %>%
  filter(cook >= 0.1) %>%
  print(width = Inf)
```

```
## # A tibble: 2 x 17
##   county    equip econ  perAA usage atlanta     gore  bush other votes ballots
##   <chr>     <fct> <fct> <dbl> <fct> <fct>      <int> <int> <int> <int>   <int>
## 1 BEN.HILL OS-PC poor  0.282 rural notAtlanta  2234  2381    46  4661    5741
## 2 RANDOLPH OS-PC poor  0.527 rural notAtlanta  1381  1174    14  2569    3021
##   undercount pergore perbush cpergore cperAA  cook
##        <dbl>   <dbl>   <dbl>    <dbl>  <dbl> <dbl>
## 1      0.188   0.479   0.511   0.0710 0.0390 0.234
## 2      0.150   0.538   0.457   0.129  0.284  0.138
```

Let's plot a bubble plot using `car` package

```
influencePlot(lmodi)
```



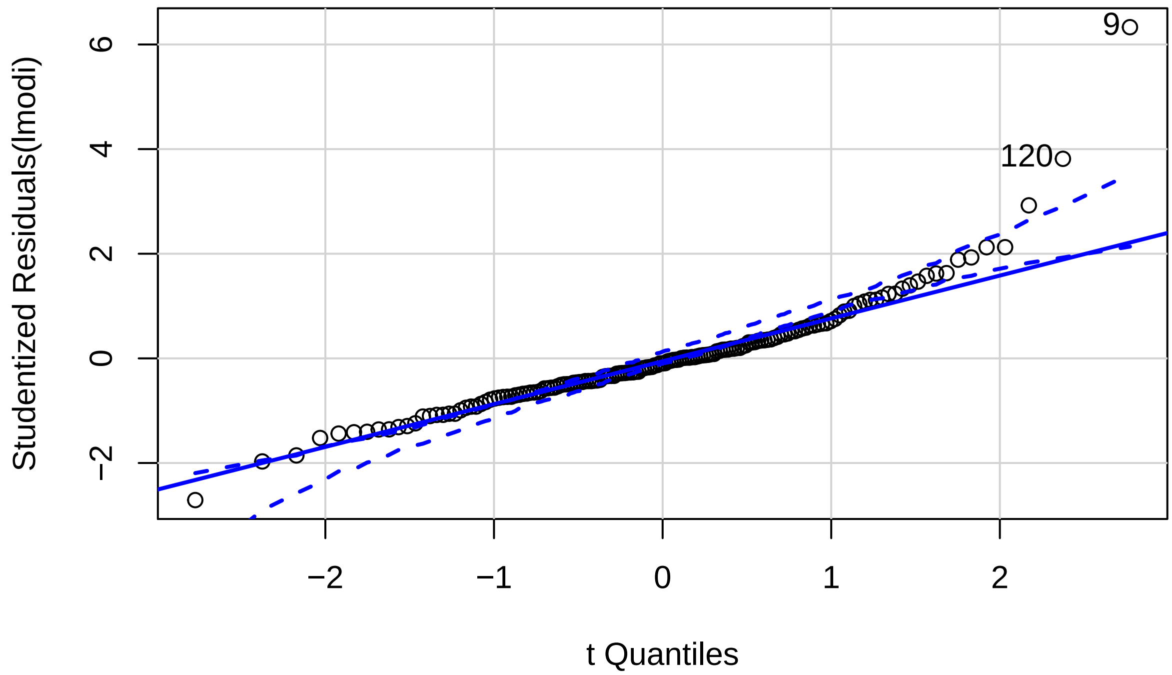

```
##         StudRes        Hat      CookD
## 9    6.3305112 0.06215999 0.23413887
## 103 -0.5714242 0.51689352 0.03899306
## 120  3.8143062 0.08489464 0.13754389
## 131  0.5714242 0.51689352 0.03899306
```

The **bubble plot** combines the display of Studentized residuals, hat-values, and Cook's distances, with the areas of the circles proportional to Cook's D_i.

Another way to generate a Q-Q plot using the `car` package. The default of the `aaPlot()` function in the car package plots Studentized residuals against the corresponding quantiles of $t(n-p-2)$ and generates a 95% pointwise confidence envelope for the Studentized residuals, using a parametric version of the bootstrap.

```
qqPlot(lmodi)
```

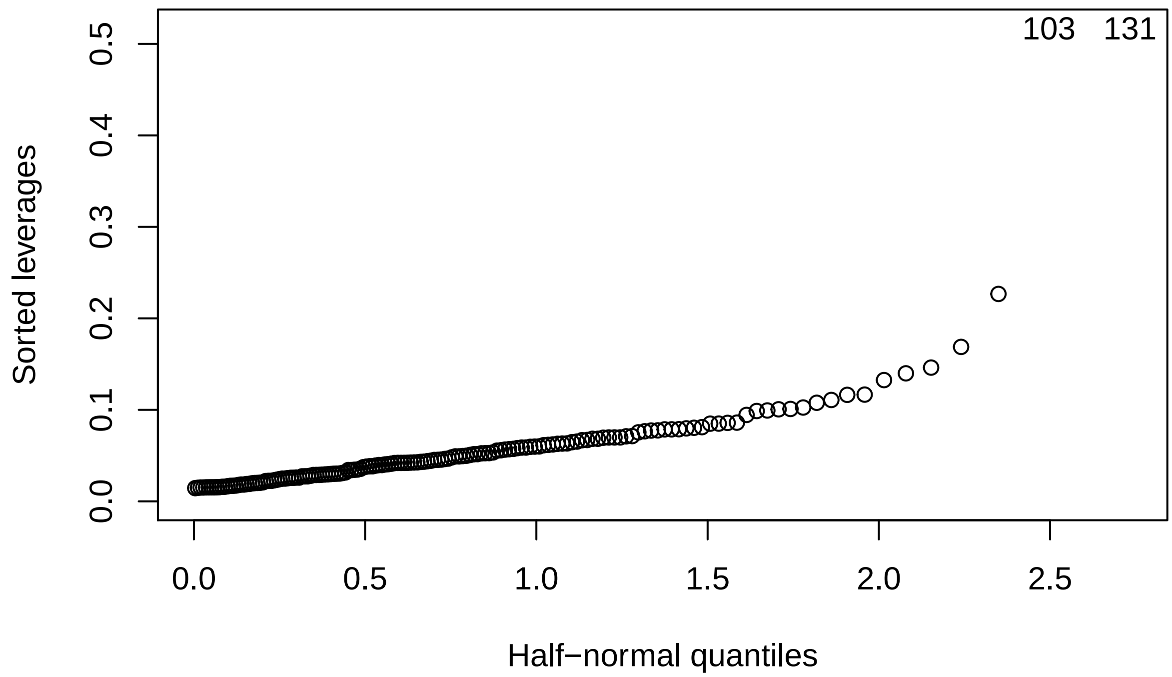
```
## Warning in rlm.default(x, y, weights, method = method, wt.method = wt.method, :
## 'rlm' failed to converge in 20 steps
```




```
## [1]   9 120
```

- Another useful plot to inspect potential outliers in positive values is the **half-normal plot**. Here we plot the sorted leverages h_i against the standard normal quantiles $\Phi^{-1}\left(\frac{n+i}{2n+1}\right)$. We do not expect a necessary straight line, just look for outliers, which is far away from the rest of the data.

```
# this function is available from faraway package
halfnorm(hatvalues(lmodi), ylab = "Sorted leverages")
```



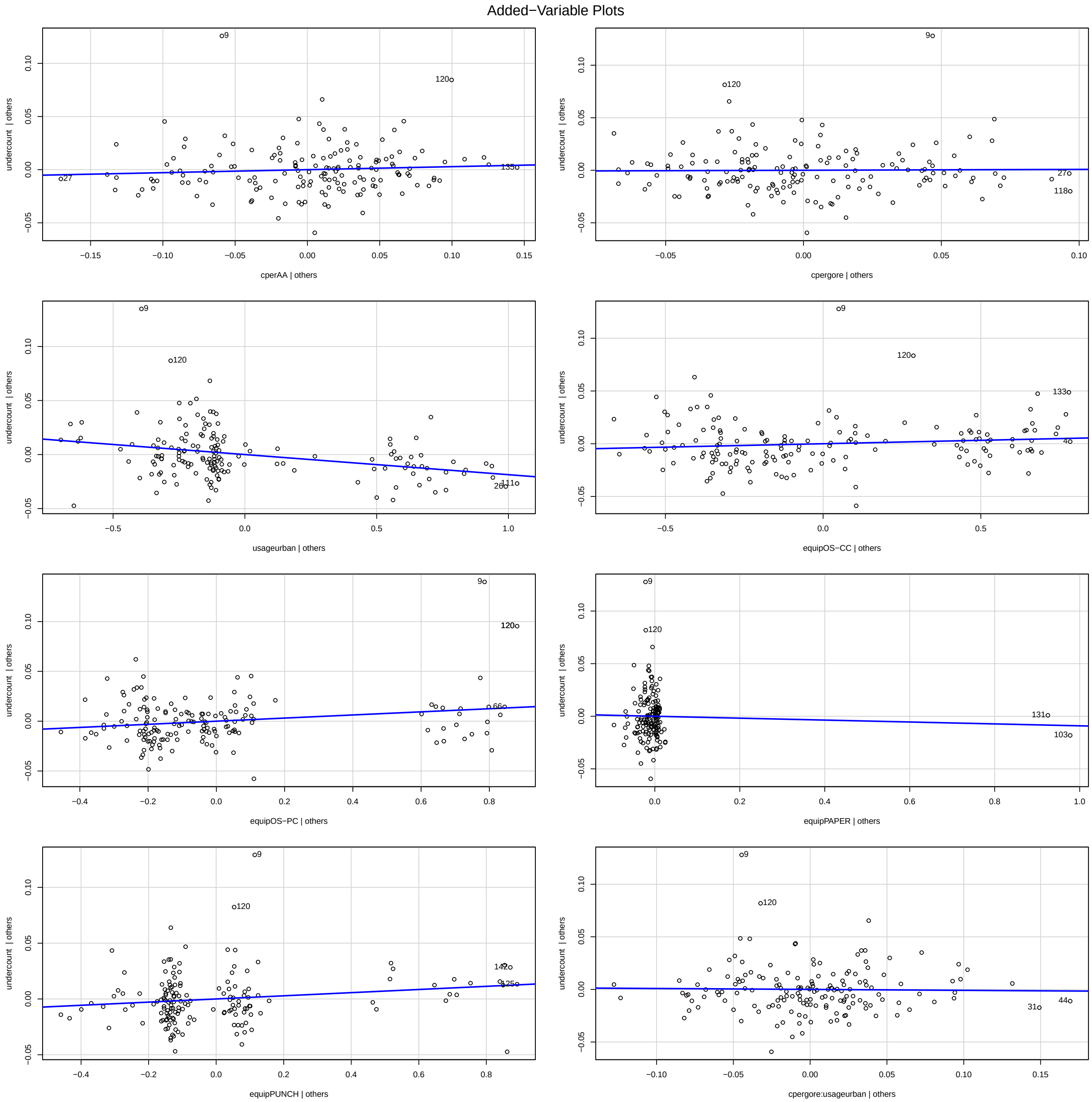
These two counties have unusually large leverages. They are actually the only counties that use paper ballot.

```r
gavote %>%
  # mutate(hi = hatvalues(lmodi)) %>%
  # filter(hi > 0.4) %>%
  slice(c(103, 131)) %>%
  print(width = Inf)
```

```
## # A tibble: 2 x 16
##   county     equip econ  perAA usage atlanta     gore  bush other votes ballots
##   <chr>      <fct> <fct> <dbl> <fct> <fct>      <int> <int> <int> <int>   <int>
## 1 MONTGOMERY PAPER poor  0.243 rural notAtlanta  1013  1465    31  2509    2573
## 2 TALIAFERRO PAPER poor  0.596 rural notAtlanta   556   271     5   832     881
##   undercount pergore perbush cpergore    cperAA
##        <dbl>   <dbl>   <dbl>    <dbl>     <dbl>
## 1     0.0249   0.404   0.584 -0.00458 0.0000189
## 2     0.0556   0.668   0.326  0.260   0.353
```

Added-variable plots

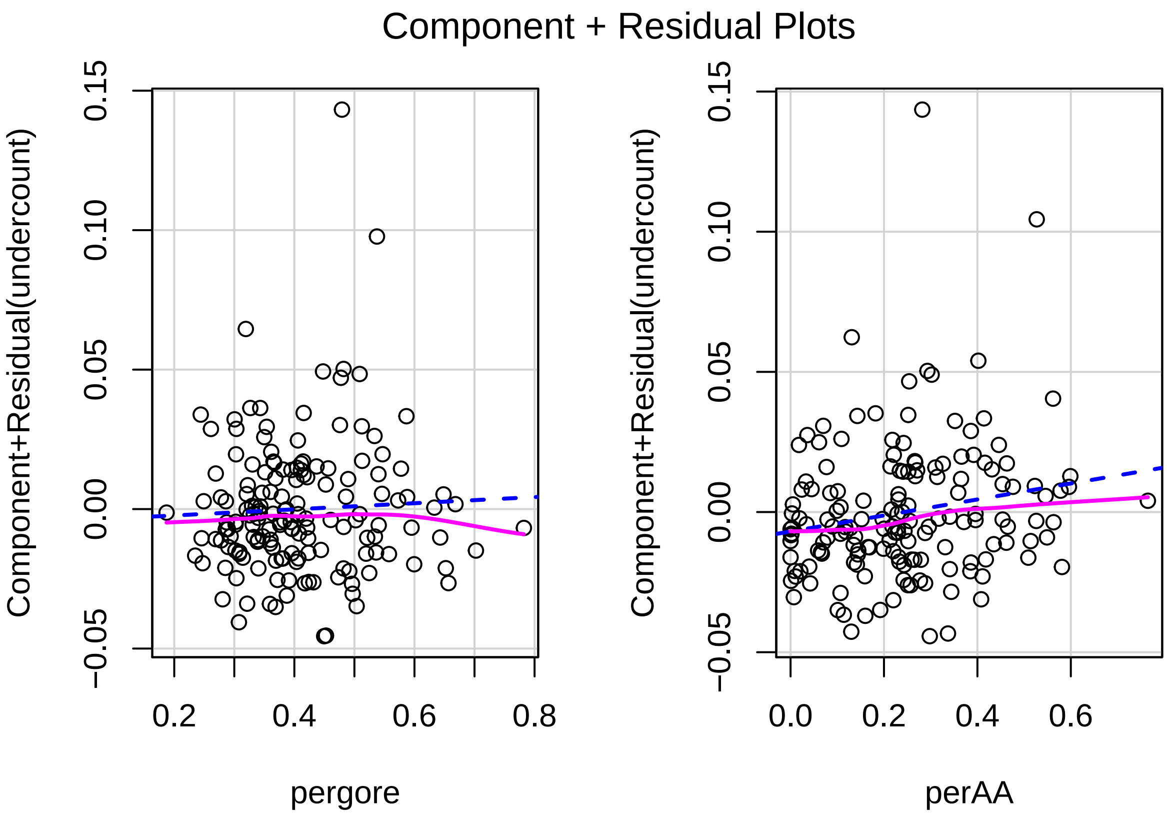
```r
# avPlots(lmodi)
# change layout
avPlots(lmodi, layout = c(4, 2))
```



Component-plus-residuals plot

We can generate component-plus-residual plots by `crPlots()` function

```r
crPlots(lmod)  # from car package
```



Component + Residual Plots
Component+Residual(undercount)
0.15
0.10
0.05
0.00
−0.05
0.2
0.4
0.6
0.8
pergore
Component+Residual(undercount)
0.0
0.2
0.4
0.6
perAA