432 Class 15 Slides

github.com/THOMASELOVE/2020-432

2020-03-19

Setup

```r
library(here); library(magrittr); library(janitor)

library(knitr)
library(MASS)
library(robustbase)
library(quantreg)
library(lmtest)
library(sandwich)
library(boot)
library(rms)
library(broom)
library(tidyverse)

decim <- function(x, k) format(round(x, k), nsmall=k)
theme_set(theme_bw())
```

Today's Materials

- The `crimestat` data
- Robust Linear Regression Methods
    - with Huber weights
    - with bisquare weights (biweights)
    - Bounded Influence Regression & Least Trimmed Squares
    - Penalized Least Squares using `ols` in `rms` package
    - Quantile Regression on the Median

The `crimestat` data and an OLS fit

The `crimestat` data set

For each of 51 states (including the District of Columbia), we have the state's ID number, postal abbreviation and full name, as well as:

- **crime** - the violent crime rate per 100,000 people
- **poverty** - the official poverty rate (% of people living in poverty in the state/district) in 2014
- **single** - the percentage of households in the state/district led by a female householder with no spouse present and with her own children under 18 years living in the household in 2016

The `crimestat` data set

```
crimestat <- read_csv("data/crimestat.csv")
crimestat
```

```
# A tibble: 51 x 6
     sid state crime poverty single state.full
   <dbl> <chr> <dbl>   <dbl>  <dbl> <chr>
 1     1 AL     427.    19.2   9.02 Alabama
 2     2 AK     636.    11.4   7.63 Alaska
 3     3 AZ     400.    18.2   8.31 Arizona
 4     4 AR     480.    18.7   9.41 Arkansas
 5     5 CA     396.    16.4   7.25 California
 6     6 CO     309.    12.1   6.75 Colorado
 7     7 CT     237.    10.8   8.04 Connecticut
 8     8 DE     489.    13     6.52 Delaware
 9     9 DC    1244.    18.4   8.41 District of Columbia
10    10 FL     540.    16.6   8.29 Florida
# ... with 41 more rows
```

Modeling `crime` with `poverty` and `single`

Our main goal will be to build a linear regression model to predict **crime** using centered versions of both **poverty** and **single**.

```
crimestat <- crimestat %>%
    mutate(pov_c = poverty - mean(poverty),
           single_c = single - mean(single))
```

Our original (OLS) model

Note the sneaky trick with the outside parentheses...

```
(mod1 <- lm(crime ~ pov_c + single_c, data = crimestat))
```

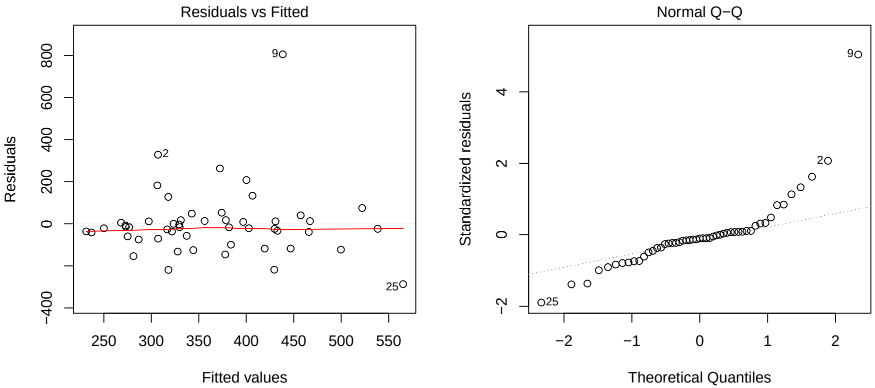
```
Call:
lm(formula = crime ~ pov_c + single_c, data = crimestat)

Coefficients:
(Intercept)        pov_c     single_c
     364.41        16.11        23.84
```

Coefficients?

```
tidy(mod1, conf.int = TRUE) %>%
  select(term, estimate, std.error,
         p.value, conf.low, conf.high) %>%
  kable(digits = 3)
```

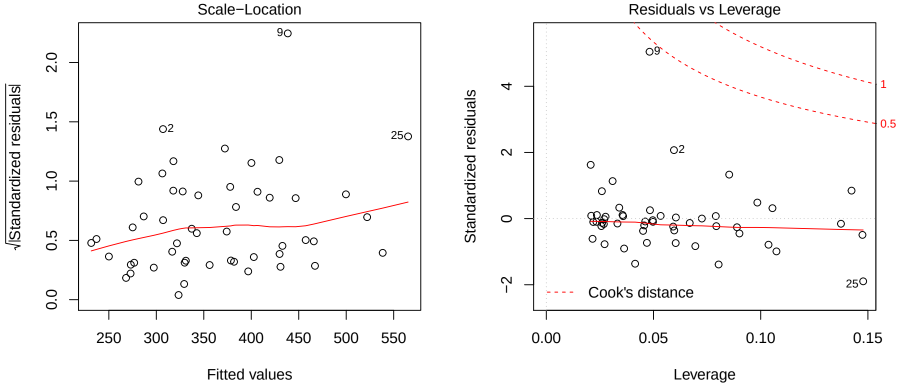
term	estimate	std.error	p.value	conf.low	conf.high
(Intercept)	364.406	22.933	0.000	318.297	410.515
pov_c	16.115	9.616	0.100	-3.219	35.448
single_c	23.843	18.384	0.201	-13.121	60.807

OLS Residuals



Which points are highlighted here?

Remaining Residual Plots from OLS



So which points are of special interest?

Which points are those?

```r
crimestat %>%
  slice(c(2, 9, 25))
```

```
# A tibble: 3 x 8
    sid state crime poverty single state.full pov_c single_c
  <dbl> <chr> <dbl>   <dbl>  <dbl> <chr>      <dbl>    <dbl>
1     2 AK     636.    11.4   7.63 Alaska     -3.47  -0.0588
2     9 DC    1244.    18.4   8.41 District ~  3.53   0.721
3    25 MS     278.    21.9  11.4  Mississip~  7.03   3.67
```

Robust Linear Regression with Huber Weights

Robust Linear Regression with Huber weights

There are several ways to do robust linear regression using M-estimation, including weighting using Huber and bisquare strategies.

- Robust linear regression here will make use of a method called iteratively re-weighted least squares (IRLS) to estimate models.
- M-estimation defines a weight function which is applied during estimation.
- The weights depend on the residuals and the residuals depend on the weights, so an iterative process is required.

We'll fit the model, using the default weighting choice: what are called Huber weights, where observations with small residuals get a weight of 1, and the larger the residual, the smaller the weight.

Our robust model (using `MASS::rlm`)

```
rob.huber <- rlm(crime ~ pov_c + single_c, data = crimestat)
```

Summary of the robust (Huber weights) model

```r
tidy(rob.huber) %>%
  kable(digits = 3)
```

term	estimate	std.error	statistic
(Intercept)	343.798	13.131	26.182
pov_c	11.910	5.506	2.163
single_c	30.987	10.527	2.944

Now, *both* predictors appear to have estimates that exceed twice their standard error. So this is a very different result than ordinary least squares gave us.

Glance at the robust model (vs. OLS)

```
glance(mod1)
```

```
# A tibble: 1 x 11
  r.squared adj.r.squared sigma statistic p.value    df
      <dbl>         <dbl> <dbl>     <dbl>   <dbl> <int>
1     0.197         0.163  164.      5.88 0.00518     3
# ... with 5 more variables: logLik <dbl>, AIC <dbl>,
#   BIC <dbl>, deviance <dbl>, df.residual <int>
```

```
glance(rob.huber)
```

```
# A tibble: 1 x 6
  sigma converged logLik   AIC   BIC deviance
  <dbl> <lgl>      <dbl> <dbl> <dbl>    <dbl>
1  59.1 TRUE       -331.  671.  678. 1314784.
```

Understanding the Huber weights a bit

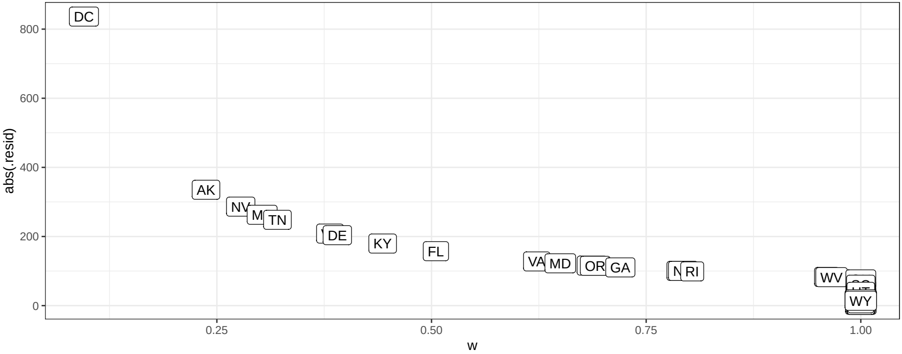
Let's augment the data with results from this model, including the weights used.

```
crime_with_huber <- augment(rob.huber, crimestat) %>%
    mutate(w = rob.huber$w) %>% arrange(w) %>% tbl_df

head(crime_with_huber, 3)
```

```
# A tibble: 3 x 14
    sid state crime poverty single state.full  pov_c
  <dbl> <chr> <dbl>   <dbl>  <dbl> <chr>       <dbl>
1     9 DC    1244.    18.4   8.41 District ~  3.53
2     2 AK     636.    11.4   7.63 Alaska     -3.47
3    29 NV     636.    15.4   7.66 Nevada      0.527
# ... with 7 more variables: single_c <dbl>, .fitted <dbl>,
#   .se.fit <dbl>, .resid <dbl>, .hat <dbl>, .sigma <dbl>,
#   w <dbl>
```

Are cases with large residuals down-weighted?

```r
ggplot(crime_with_huber, aes(x = w, y = abs(.resid))) +
    geom_label(aes(label = state))
```



Conclusions from the Plot of Weights

- The district of Columbia will be down-weighted the most, followed by Alaska and then Nevada and Mississippi.
- But many of the observations will have a weight of 1.
- In ordinary least squares, all observations would have weight 1.
- So the more cases in the robust regression that have a weight close to one, the closer the results of the OLS and robust procedures will be.

summary(rob.huber)

```
Call: rlm(formula = crime ~ pov_c + single_c, data = crimestat
Residuals:
     Min       1Q   Median       3Q      Max
-262.751  -45.641    1.762   36.732  836.244

Coefficients:
            Value    Std. Error t value
(Intercept) 343.7982  13.1309    26.1823
pov_c        11.9098   5.5058     2.1631
single_c     30.9868  10.5266     2.9437

Residual standard error: 59.14 on 48 degrees of freedom
```

Robust Linear Regression with the bisquare weighting function

Robust Linear Regression with the biweight

As mentioned there are several possible weighting functions - we'll next try the biweight, also called the bisquare or Tukey's bisquare, in which all cases with a non-zero residual get down-weighted at least a little. Here is the resulting fit...

```
(rob.biweight <- rlm(crime ~ pov_c + single_c,
                     data = crimestat, psi = psi.bisquare))
```

```
Call:
rlm(formula = crime ~ pov_c + single_c, data = crimestat, psi
Converged in 13 iterations

Coefficients:
(Intercept)       pov_c    single_c 
  336.17015    10.31578    34.70765 

Degrees of freedom: 51 total; 48 residual
Scale estimate: 67.3
```

Coefficients and Standard Errors

```
tidy(rob.biweight) %>% kable(digits = 3)
```

term	estimate	std.error	statistic
(Intercept)	336.170	12.673	26.526
pov_c	10.316	5.314	1.941
single_c	34.708	10.160	3.416

Understanding the biweights weights a bit

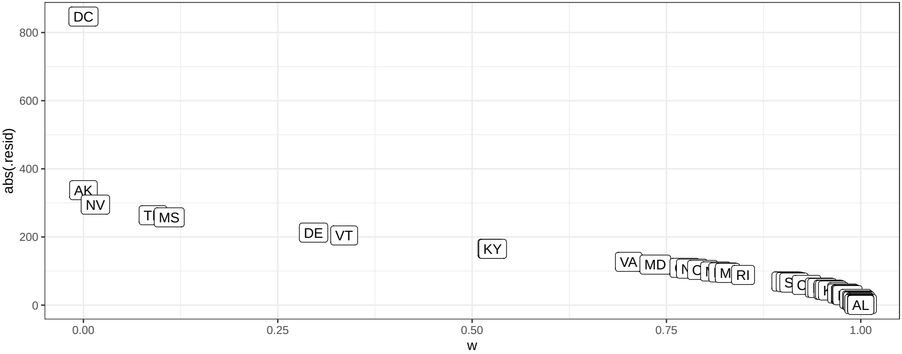
Let's augment the data, as above

```
crime_with_biweights <- augment(rob.biweight, crimestat) %>%
    mutate(w = rob.biweight$w) %>% arrange(w) %>% tbl_df

head(crime_with_biweights, 3)
```

```
# A tibble: 3 x 12
    sid state crime poverty single state.full  pov_c
  <dbl> <chr> <dbl>   <dbl>  <dbl> <chr>       <dbl>
1     2 AK     636.    11.4   7.63 Alaska     -3.47
2     9 DC    1244.    18.4   8.41 District ~  3.53
3    29 NV     636.    15.4   7.66 Nevada      0.527
# ... with 5 more variables: single_c <dbl>, .fitted <dbl>,
#   .se.fit <dbl>, .resid <dbl>, w <dbl>
```

Relationship of Weights and Residuals

```r
ggplot(crime_with_biweights, aes(x = w, y = abs(.resid))) +
    geom_label(aes(label = state))
```



Conclusions from the biweights plot

Again, cases with large residuals (in absolute value) are down-weighted generally, but here, Alaska and Washington DC receive no weight at all in fitting the final model.

- We can see that the weight given to DC and Alaska is dramatically lower (in fact it is zero) using the bisquare weighting function than the Huber weighting function and the parameter estimates from these two different weighting methods differ.
- The maximum weight (here, for Alabama) for any state using the biweight is still slightly smaller than 1.

summary(rob.biweight)

```
Call: rlm(formula = crime ~ pov_c + single_c, data = crimestat
Residuals:
    Min      1Q  Median      3Q     Max
-257.58  -40.53    8.01   45.30  846.81

Coefficients:
            Value    Std. Error t value
(Intercept) 336.1702  12.6733    26.5259
pov_c        10.3158   5.3139     1.9413
single_c     34.7077  10.1598     3.4162

Residual standard error: 67.27 on 48 degrees of freedom
```

Comparing OLS and the two weighting schemes

```
glance(mod1) # OLS
```

```
# A tibble: 1 x 11
  r.squared adj.r.squared sigma statistic p.value    df
      <dbl>         <dbl> <dbl>     <dbl>   <dbl> <int>
1     0.197         0.163  164.      5.88 0.00518     3
# ... with 5 more variables: logLik <dbl>, AIC <dbl>,
#   BIC <dbl>, deviance <dbl>, df.residual <int>
```

Comparing OLS and the two weighting schemes

```
glance(rob.biweight) # biweights
```

```
# A tibble: 1 x 6
  sigma converged logLik   AIC   BIC deviance
  <dbl> <lgl>      <dbl> <dbl> <dbl>    <dbl>
1  67.3 TRUE       -332.  672.  679. 1339850.
```

```
glance(rob.huber) # Huber weights
```

```
# A tibble: 1 x 6
  sigma converged logLik   AIC   BIC deviance
  <dbl> <lgl>      <dbl> <dbl> <dbl>    <dbl>
1  59.1 TRUE       -331.  671.  678. 1314784.
```

Bounded-Influence Regression

Bounded-Influence Regression and Least-Trimmed Squares

Under certain circumstances, M-estimators can be vulnerable to high-leverage observations, and so, bounded-influence estimators, like least-trimmed squares (LTS) regression have been proposed. The biweight that we have discussed is often fitted as part of what is called an MM-estimation procedure, by using an LTS estimate as a starting point.

The `ltsReg` function, which is part of the `robustbase` package (Note: **not** the `ltsreg` function from `MASS`) is what I use below to fit a least-trimmed squares model. The LTS approach minimizes the sum of the h smallest squared residuals, where h is greater than $n/2$, and by default is taken to be $(n + p + 1)/2$.

Least Trimmed Squares Model

```
lts1 <- ltsReg(crime ~ pov_c + single_c, data = crimestat)
```

Summarizing the LTS model

```r
summary(lts1)$coeff
```

```
           Estimate Std. Error   t value     Pr(>|t|)
Intercept 339.14817  11.616766 29.194715 1.601245e-29
pov_c      16.99322   4.973459  3.416781 1.418337e-03
single_c   24.99819   9.136683  2.736024 9.073473e-03
```

MM estimation

Specifying the argument `method="MM"` to `rlm` requests bisquare estimates with start values determined by a preliminary bounded-influence regression, as follows...

```
rob.MM <- rlm(crime ~ pov_c + single_c,
              data = crimestat, method = "MM")

glance(rob.MM)
```

```
# A tibble: 1 x 6
  sigma converged logLik   AIC   BIC deviance
  <dbl> <lgl>      <dbl> <dbl> <dbl>    <dbl>
1  75.8 TRUE       -332.  672.  679. 1337077.
```

summary(rob.MM)

```
Call: rlm(formula = crime ~ pov_c + single_c, data = crimestat
Residuals:
     Min       1Q   Median       3Q      Max
-252.412  -41.096    8.696   47.141  847.128

Coefficients:
            Value    Std. Error t value
(Intercept) 336.3928  13.1929    25.4980
pov_c        10.5579   5.5318     1.9086
single_c     32.7755  10.5763     3.0989

Residual standard error: 75.79 on 48 degrees of freedom
```

Penalized Least Squares

Penalized Least Squares with rms

We can apply a penalty to least squares directly through the `ols` function in the `rms` package.

```
d <- datadist(crimestat)
options(datadist = "d")
pls <- ols(crime ~ pov_c + single_c, penalty = 1,
            data = crimestat, x=T, y = T)
```

The pls fit

```
Linear Regression Model

 ols(formula = crime ~ pov_c + single_c, data = crimestat, x =
     y = T, penalty = 1)

                      Model Likelihood      Discrimination
                         Ratio Test            Indexes
 Obs          51     LR chi2      11.18     R2       0.197
 sigma159.1209     d.f.      1.946198     R2 adj   0.164
 d.f.   48.0538     Pr(> chi2)   0.0035     g       89.298

 Residuals

     Min       1Q   Median       3Q      Max
 -284.24   -65.93   -16.68    15.66   807.01


              Coef      S.E.      t      Pr(>|t|)
```

Continuing the pls output

```
     Coef      S.E.    t     Pr(>|t|)
Intercept 364.4059 22.2814 16.35 <0.0001
pov_c      15.8488  9.1387  1.73 0.0893
single_c   23.6857 17.4723  1.36 0.1816
```

How to Choose the Penalty in Penalized Least Squares?

The problem here is how to choose the penalty - and that's a subject I'll essentially skip today. The most common approach (that we've seen with the lasso) is cross-validation.

Meanwhile, what do we conclude about the fit here from AIC and BIC?

```
AIC(pls); BIC(pls)
```

```
    d.f.
669.5781
```

```
    d.f.
677.2014
```

Quantile Regression (on the Median)

Quantile Regression on the Median

We can use the rq function in the quantreg package to model the **median** of our outcome (violent crime rate) on the basis of our predictors, rather than the mean, as is the case in ordinary least squares.

```
rob.quan <- rq(crime ~ pov_c + single_c, data = crimestat)

glance(rob.quan)
```

```
# A tibble: 1 x 5
    tau logLik   AIC   BIC df.residual
  <dbl>  <dbl> <dbl> <dbl>       <int>
1   0.5  -316.  638.  643.          48
```

summary(rob.quan)

```
Call: rq(formula = crime ~ pov_c + single_c, data = crimestat)

tau: [1] 0.5

Coefficients:
            coefficients lower bd  upper bd
(Intercept) 344.75658    336.94534 366.23603
pov_c        10.54757      3.06714  28.95962
single_c     32.27249      4.45889  48.18925
```

Estimating a different quantile (tau = 0.70)

In fact, if we like, we can estimate any quantile by specifying the `tau` parameter (here `tau` = 0.5, by default, so we estimate the median.)

```
(rob.quan70 <- rq(crime ~ pov_c + single_c, tau = 0.70,
                  data = crimestat))
```

```
Call:
rq(formula = crime ~ pov_c + single_c, tau = 0.7, data = crime

Coefficients:
(Intercept)       pov_c    single_c
  379.72818    19.30376    32.15827

Degrees of freedom: 51 total; 48 residual
```

Conclusions

Comparing Five of the Models

Estimating the Mean

Fit	Intercept CI	pov_c CI	single_c CI
OLS	(318.6, 410.2)	(-3.13, 35.35)	(-12.92, 60.60)
Robust (Huber)	(320.0, 367.6)	(0.89, 22.93)	(9.93, 52.05)
Robust (biweight)	(310.7, 361.5)	(-0.30, 20.94)	(14.39, 55.03)
Robust (MM)	(310.0, 362.8)	(-0.50, 21.62)	(11.62, 53.94)

Note: CIs estimated for OLS and Robust methods as point estimate $\pm$ 2 standard errors

Estimating the Median

Fit	Intercept CI	pov_c CI	single_c CI
Quantile (Median) Reg	(336.9, 366.2)	(3.07, 28.96)	(4.46, 48,19)

Comparing AIC and BIC

Fit	AIC	BIC
OLS	669.7	677.4
Robust (Huber)	670.8	678.5
Robust (biweight)	671.7	679.4
Robust (MM)	671.6	679.3
Quantile (median)	637.5	643.3

Some General Thoughts

1. When comparing the results of a regular OLS regression and a robust regression for a data set which displays outliers, if the results are very different, you will most likely want to use the results from the robust regression.
    - Large differences suggest that the model parameters are being highly influenced by outliers.
2. Different weighting functions have advantages and drawbacks.
    - Huber weights can have difficulties with really severe outliers.
    - Bisquare weights can have difficulties converging or may yield multiple solutions.
    - Quantile regression approaches have some nice properties, but describe medians (or other quantiles) rather than means.

Next Time

Regression on a Count Outcome