Lecture 25: Robust Regression

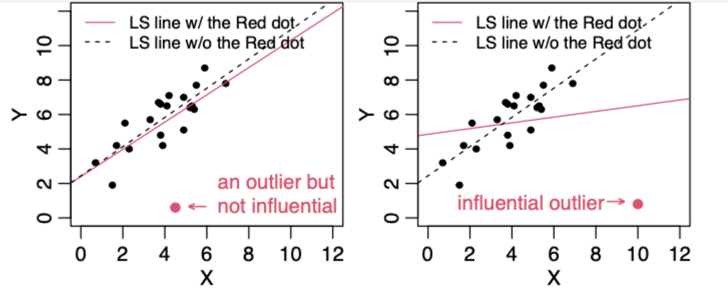
Ailin Zhang

2023-07-19

Agenda

- Need for Robust Regression
- M-Estimation
- Conclusion

Need for Robust Regression



- There are outliers that affect the regression model.
- The response variable is nonnormal. (A distribution that has longer or heavier tails than the normal is common)

Treatment for outliers

- One way to deal with this situation is to discard the observation. Data can sometimes be discarded (or modified) on the basis of subject-matter knowledge.

Treatment for outliers

- One way to deal with this situation is to discard the observation. Data can sometimes be discarded (or modified) on the basis of subject-matter knowledge.
- However, we are now discarding observations on a statistical basis, simply because it is expedient from a statistical modeling viewpoint, and generally, this is not a good practice.
- A robust regression procedure is one that dampens the effect of observations that would be highly influential if least squares were used.

M Estimation

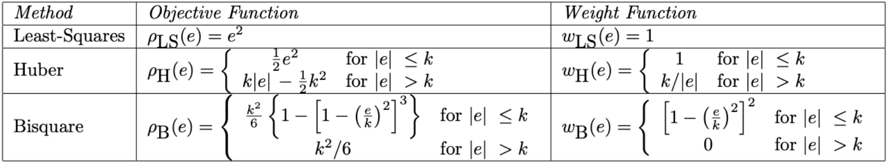
- Linear least-squares estimates can behave badly when the error distribution is not normal, particularly when the errors are heavy-tailed.
- The most common general method of robust regression is M-estimation.
- This class of estimators can be regarded as a generalization of maximum-likelihood estimation
- With M-estimation, the estimates β are determined by minimizing a particular objective function $\sum_{i=1}^{n} \rho(e_i) = \sum_{i=1}^{n} \rho(y_i - x_i^T \beta)$. where the function ρ should have the following properties:
    - Always nonnegative
    - Equal to zero when the residual is zero $\rho(0) = 0$
    - Symmetric: $\rho(e) = \rho(-e)$
    - Monotone in $|e_i|$: $\rho(e_i) \geq \rho(e_{i'})$ for $|e_i| > |e_{i'}|$
    - For example: $\rho(e_i) = e_i^2$, $\rho(e_i) = |e_i|$

Fitting to β

- Differentiating the objective function with respect to β:

$$\sum_{i=1}^{n} \rho'(y_i - x_i^T\beta)x_i^T = 0, \text{ we introduce a weight function } w_i = w(e_i) = \frac{\rho'(e_i)}{e_i}$$

- The estimating equations may be written as: $\sum_{i=1}^{n} w_i(y_i - x_i^T\beta)x_i^T = 0$
- Solving these estimating equations is equivalent to a weighted least-squares problem, minimizing $\sum w_i^2 e_i^2$.
- Iteratively Reweighted Least-Squares (IRLS) can be used here



Method	*Objective Function*	*Weight Function*
Least-Squares	$\rho_{\mathrm{LS}}(e) = e^2$	$w_{\mathrm{LS}}(e) = 1$
Huber	$\rho_{\mathrm{H}}(e) = \begin{cases} \frac{1}{2}e^2 & \text{for } \lvert e\rvert \leq k \\ k\lvert e\rvert - \frac{1}{2}k^2 & \text{for } \lvert e\rvert > k \end{cases}$	$w_{\mathrm{H}}(e) = \begin{cases} 1 & \text{for } \lvert e\rvert \leq k \\ k/\lvert e\rvert & \text{for } \lvert e\rvert > k \end{cases}$
Bisquare	$\rho_{\mathrm{B}}(e) = \begin{cases} \frac{k^2}{6}\left\{1 - \left[1 - \left(\frac{e}{k}\right)^2\right]^3\right\} & \text{for } \lvert e\rvert \leq k \\ k^2/6 & \text{for } \lvert e\rvert > k \end{cases}$	$w_{\mathrm{B}}(e) = \begin{cases} \left[1 - \left(\frac{e}{k}\right)^2\right]^2 & \text{for } \lvert e\rvert \leq k \\ 0 & \text{for } \lvert e\rvert > k \end{cases}$

Iteratively re-weighted least squares

Algorithm for robust regression:

1. Start with an estimate of the regression line, e.g. standard least squares

Repeat until convergence:

Iteratively re-weighted least squares

Algorithm for robust regression:

1. Start with an estimate of the regression line, e.g. standard least squares

Repeat until convergence:

2. Compute robustness weights based on the estimated regression line.

Iteratively re-weighted least squares

Algorithm for robust regression:

1. Start with an estimate of the regression line, e.g. standard least squares

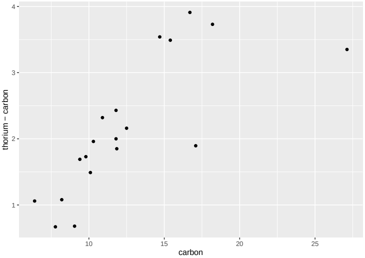
Repeat until convergence:

2. Compute robustness weights based on the estimated regression line.
3. Perform weighted least squares with the robustness weights to get a new estimate of the regression line.

Example

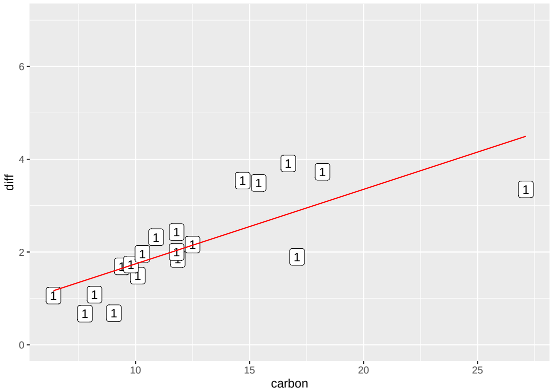
Paired observations giving the estimated ages of 19 coral samples in thousands of years using both carbon dating (the traditional method) and thorium dating (a modern and purportedly more accurate method.)

```r
load("lattice.RData")
ggplot(dating, aes(x = carbon, y = thorium - carbon)) +
    geom_point()
```



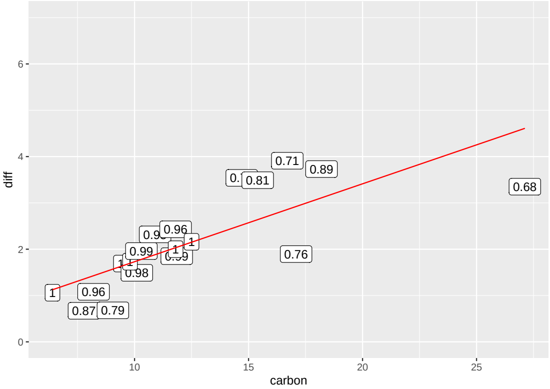
IRLS Steps

```
dating = dating %>% mutate(diff = thorium - carbon)
dating.rlm.0 = lm(diff ~ carbon, data = dating)
ggplot(data.frame(dating, weights = 1)) +
    geom_point(aes(x = carbon, y = diff)) +
    geom_label(aes(x = carbon, y = diff, label = weights)) +
    geom_line(aes(x = carbon, y = dating.rlm.0$fitted.values), color = 'red') +
    ylim(c(0,7))
```



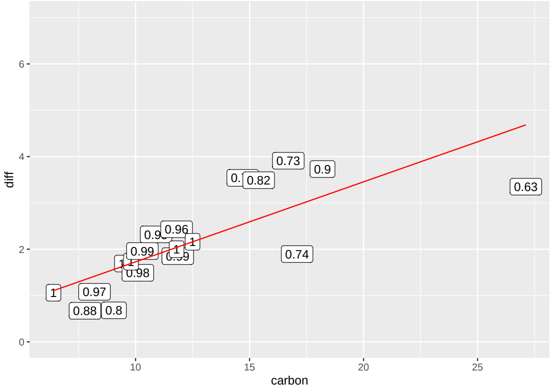
IRLS Steps

```
dating.rlm.1 = rlm(diff ~ carbon, data = dating, maxit = 1, psi = psi.bisquare)
weight_strings = as.character(round(dating.rlm.1$w, digits = 2))
ggplot(data.frame(dating, weights = weight_strings)) +
    geom_point(aes(x = carbon, y = diff)) +
    geom_label(aes(x = carbon, y = diff, label = weights)) +
    geom_line(aes(x = carbon, y = dating.rlm.1$fitted.values), color = 'red') +
    ylim(c(0,7))
```



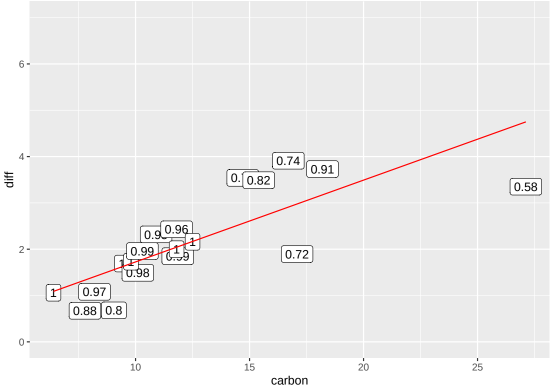
IRLS Steps

```
dating.rlm.1 = rlm(diff ~ carbon, data = dating, maxit = 2, psi = psi.bisquare)
weight_strings = as.character(round(dating.rlm.1$w, digits = 2))
ggplot(data.frame(dating, weights = weight_strings)) +
    geom_point(aes(x = carbon, y = diff)) +
    geom_label(aes(x = carbon, y = diff, label = weights)) +
    geom_line(aes(x = carbon, y = dating.rlm.1$fitted.values), color = 'red') +
    ylim(c(0,7))
```



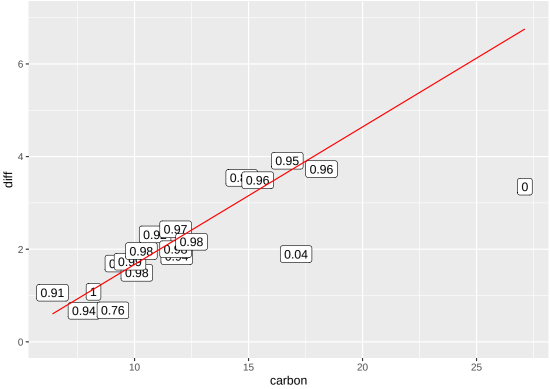
IRLS Steps

```
dating.rlm.1 = rlm(diff ~ carbon, data = dating, maxit = 3, psi = psi.bisquare)
weight_strings = as.character(round(dating.rlm.1$w, digits = 2))
ggplot(data.frame(dating, weights = weight_strings)) +
    geom_point(aes(x = carbon, y = diff)) +
    geom_label(aes(x = carbon, y = diff, label = weights)) +
    geom_line(aes(x = carbon, y = dating.rlm.1$fitted.values), color = 'red') +
    ylim(c(0,7))
```



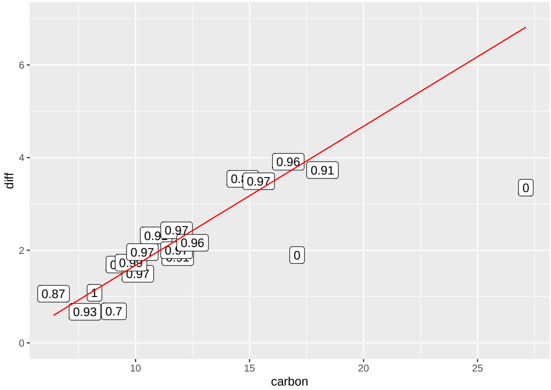
IRLS Steps

```
dating.rlm.1 = rlm(diff ~ carbon, data = dating, maxit = 11, psi = psi.bisquare)
weight_strings = as.character(round(dating.rlm.1$w, digits = 2))
ggplot(data.frame(dating, weights = weight_strings)) +
    geom_point(aes(x = carbon, y = diff)) +
    geom_label(aes(x = carbon, y = diff, label = weights)) +
    geom_line(aes(x = carbon, y = dating.rlm.1$fitted.values), color = 'red') +
    ylim(c(0,7))
```

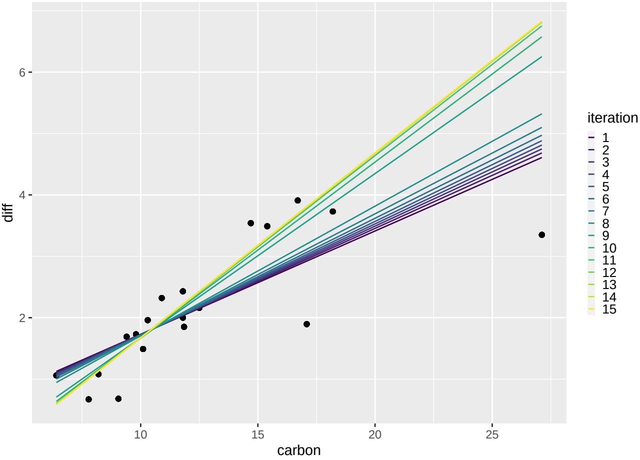


IRLS Steps

```r
dating.rlm.1 = rlm(diff ~ carbon, data = dating, maxit = 12, psi = psi.bisquare)
weight_strings = as.character(round(dating.rlm.1$w, digits = 2))
ggplot(data.frame(dating, weights = weight_strings)) +
    geom_point(aes(x = carbon, y = diff)) +
    geom_label(aes(x = carbon, y = diff, label = weights)) +
    geom_line(aes(x = carbon, y = dating.rlm.1$fitted.values), color = 'red') +
    ylim(c(0,7))
```



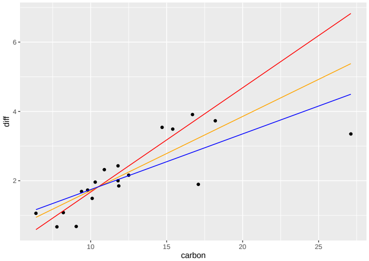
To see how the algorithm gets to the final fit, we can solve for the fits for all iterations between 1 and 15 and plot them:



M-Estimation

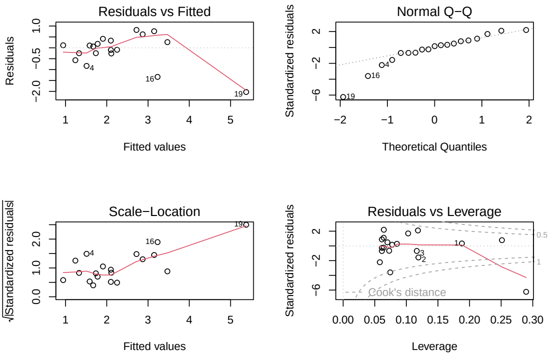
```
dating.rlm.huber = rlm(diff ~ carbon, data = dating, psi = psi.huber, maxit = 100)
dating.rlm.bisquare = rlm(diff ~ carbon, data = dating, psi = psi.bisquare)

ggplot(dating) +
    ## raw data
    geom_point(aes(x = carbon, y = diff)) +
    ## rlm bisquare fit
    geom_line(aes(x = carbon, y = dating.rlm.bisquare$fitted.values), color = 'red'
    ## rlm huber fit
    geom_line(aes(x = carbon, y = .fitted), color = 'orange', data=augment(dating.rl
    ## lm fit
    geom_line(aes(x = carbon, y = .fitted), data = augment(dating.lm), color = 'blue
```



Diagnostics of M-Estimation

```r
par(mfrow=c(2,2))
plot(dating.rlm.huber)
```



Comparison of the two methods

Bisquare:

- Can completely remove the influnce of outliers from the regression line (the weight function is equal to 0 for outliers more than 3 or 4 standard deviations).
- Doesn't have a unique solution, can get stuck in local optima, sometimes helps to give it a good initialization point.

Huber:

- Never completely removes the influence of outliers from the regression line (weight function is never equal to 0).
- Has a unique solution, no need to worry about local optima or good starting points.

Other methods have other definitions of the weight function and allow you to make different tradeoffs between tractability and outlier removal.

Conclusion

- Simple (Multiple) Linear Regression
  - Assumptions
  - Parameter Estimation (OLS, Matrix Algebra) and Interpretation
  - Test of Hypotheses
    - Testing a single coefficient
    - Testing all coefficients equal to zero
    - Testing a subset of coefficients equal to zero
    - Testing the equality(constraints) of coefficients
  - Confidence intervals and prediction intervals

Conclusion

- Dummy Variable Regression
    - Dichotomous Factor
    - Polytomous Factors
    - Interactions with predictors
    - Interpreting parameters and Hypothesis tests
- Analysis of Variance (ANOVA)
    - One-Way ANOVA Model
    - Two-Way ANOVA Model and Testing Hypothesis: main effect and interaction
    - Tukey's method
    - Higher-Way Analysis of Variance
    - ANCOVA (categorical + continuouns)
- Transformation of Variables
    - Polynomial
    - Ordinal

Conclusion

- Model Diagnostic
    - Unusual and Influential Data: Outliers, Leverage and Influence
        - Leverage: Hat-values
        - Detecting outliers: studentized residuals
        - Influence: Cook's Distance
        - Added-Variable Plots
    - Non-constant Error Variance
        - Residual Plots
        - Transformations to stabilize variance: Log, Power, Box-Cox
        - Weighted Least Squares
    - Nonlinearity
        - Component plus residual plots
        - Transformations
    - Nonnormality (skewness)
        - Normal QQ Plot
        - Power transformations

Conclusion: Model Diagnostic

- Multicollinearity
  - Detection (VIFs, etc. . .)
  - Remedies:
    - Model Respecification
    - Variable Selection
    - More Data
    - PCA
    - LASSO, Ridge Regression
- Correlated Errors
  - Time series/ sequential data
  - Hypothesis test: runs test, Durbin-Watson Statistic
  - Seasonality
  - Time series regression:
    - Cochrane-Orcutt Method
    - AR/MA/ARMA/ARIMA/SARIMA

Conclusion

- Model Building and Selection
    - Model Selection Criteria
        - Residual Mean Square
        - Information Criteria
    - Forward Selection
    - Backward Elimination
    - Stepwise Method
- Other Regression models
    - Generalized linear models
    - Nonlinear regression
    - Time-series regression
    - Robust regression

Plans for 2023 Fall

- STAT 4060J: Computational Methods in Statistics

  Computational topics of different algorithms, i.e. Computational Algorithms for fitting a regression.

- STAT 4510J: Bayesian Analysis

  The word "Bayesian" is almost everywhere, for example naive bayes classifier, bayesian networks, and bayesian optimization. We will look at Bayesian foundations and Bayesian modeling. You will be able to use Bayesian methods whenever needed in applications

- STAT 4710J: Data Science And Analytics in Python (Likely)

  Survey over most commonly used techniques and methods in DS

Join us!

We are always looking for passionate TAs for DS courses. If you find any of the courses interesting, we would like to encourage you to contribute as a TA.