Gov 2000: 12. Troubleshooting the Linear Model

Matthew Blackwell

Fall 2016

1. Outliers, leverage points, and influential observations

2. Heteroskedasticity

3. Nonlinearity of the regression function

Where are we? Where are we going?

- Last few weeks: estimation and inference for the linear model under Gauss-Markov assumptions (and sometimes conditional Normality)
- This week: what happens when the assumptions fail? Can we tell? Can we fix it?
- Next weeks: dealing with panel data.

Review of the OLS assumptions

1. Linearity: $y_i = \mathbf{x}_i' \boldsymbol{\beta} + u_i$
2. Random sample: $(y_i, \mathbf{x}_i')$ are a iid sample from the population.
3. Full rank: $\mathbf{X}$ is an $n \times (k+1)$ matrix with rank $k+1$
4. Zero conditional mean: $\mathbb{E}[u_i|\mathbf{x}_i] = 0$
5. Homoskedasticity: $\mathbb{V}[u_i|\mathbf{x}_i] = \sigma_u^2$
6. Normality: $u_i|\mathbf{x}_i \sim N(0, \sigma_u^2)$

- 1-4 give us unbiasedness/consistency
- 1-5 are the Gauss-Markov, allow for large-sample inference
- 1-6 allow for small-sample inference

Violations of the assumptions

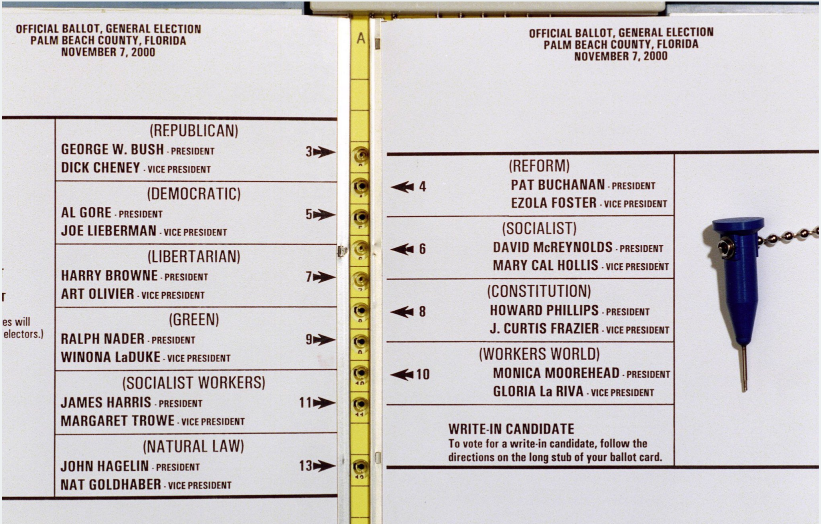
Three issues today:

1. Influential observations that skew regression estimates
2. Violations of homoskedaticity
   - $\rightsquigarrow$ SEs are biased (usually downward)
3. Incorrect functional form/nonlinearity
   - $\rightsquigarrow$ biased/inconsistent estimates

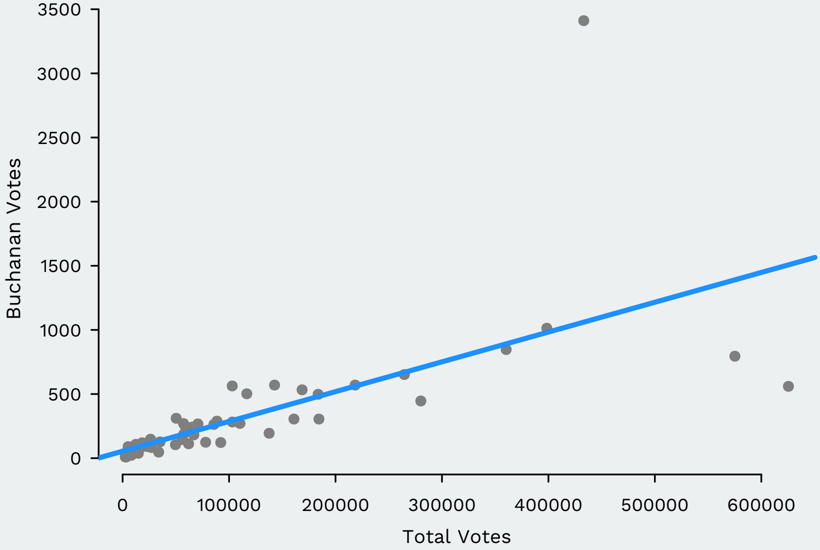
1/ Outliers, leverage points, and influential observations

Example: Buchanan votes in Florida, 2000

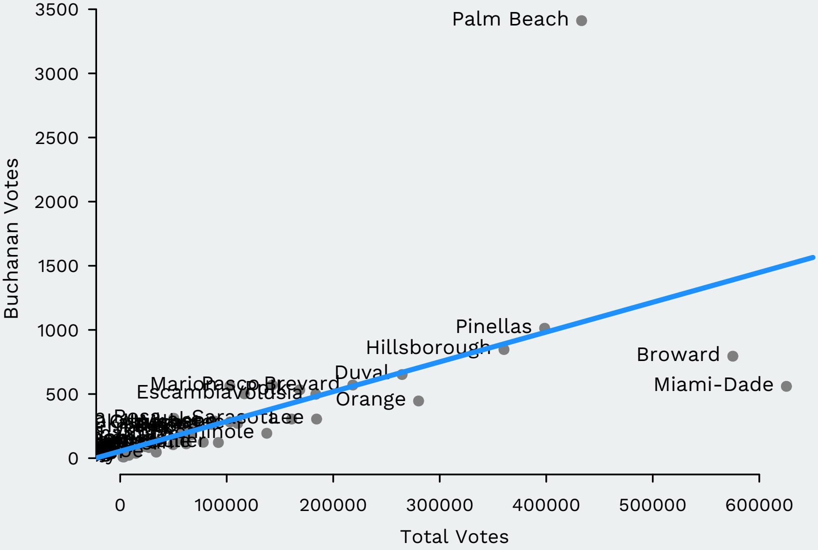
- 2000 Presidential election in FL (Wand et al., 2001, APSR)



Example: Buchanan votes in Florida, 2000



Example: Buchanan votes in Florida, 2000



Example: Buchanan votes

```r
mod <- lm(edaybuchanan ~ edaytotal, data = flvote)
summary(mod)
```

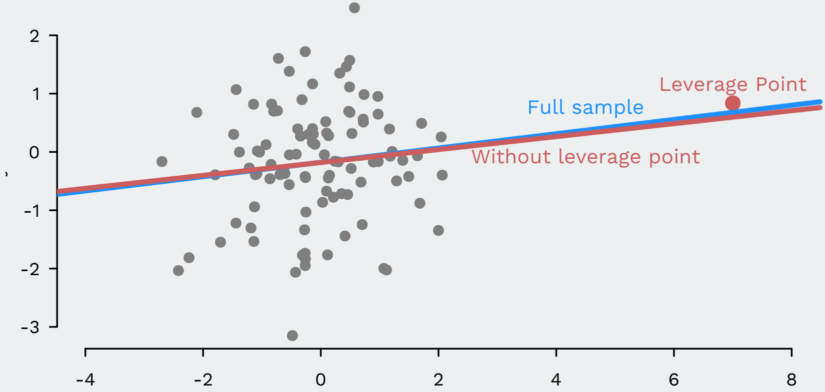
```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 54.22945   49.14146    1.10     0.27
## edaytotal    0.00232    0.00031    7.48  2.4e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 333 on 65 degrees of freedom
## Multiple R-squared:  0.463,  Adjusted R-squared:  0.455
## F-statistic:   56 on 1 and 65 DF,  p-value: 2.42e-10
```

Three types of extreme values

1. Leverage point: extreme in one x direction
2. Outlier: extreme in the y direction
3. Influence point: extreme in both directions

- Not all of these are problematic
- If the data are truly "contaminated" (come from a different distribution), can cause inefficiency and possibly bias
- Can be a violation of iid (not identically distributed)
- Diagnostics are loose

Leverage point definition



- Values that are extreme in the x direction
- That is, values far from the center of the covariate distribution
- Decrease SEs (more x variation)
- No bias if typical in y dimension

Hat matrix

- First we need to define an important matrix $\mathbf{H} = \mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'$

$$\begin{aligned}\hat{\mathbf{u}} &= \mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}} \\ &= \mathbf{y} - \mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{y} \\ &\equiv \mathbf{y} - \mathbf{H}\mathbf{y} \\ &= (\mathbf{I} - \mathbf{H})\mathbf{y}\end{aligned}$$

- $\mathbf{H}$ is the hat matrix because it puts the "hat" on $\mathbf{y}$:

$$\hat{\mathbf{y}} = \mathbf{H}\mathbf{y}$$

  - $\mathbf{H}$ is an $n \times n$ symmetric matrix
  - $\mathbf{H}$ is idempotent: $\mathbf{H}\mathbf{H} = \mathbf{H}$

Hat values

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

- For a particular observation i, we can show this means:

$$\hat{y}_i = \sum_{j=1}^{n} h_{ij} y_j$$

- h_{ij} = importance of observation j is for the fitted value $\hat{y}_i$
- Leverage/hat values: $h_i = h_{ii}$ diagonal entries of the hat matrix
- With a simple linear regression, we have

$$h_i = \frac{1}{n} + \frac{(x_i - \overline{x})^2}{\sum_{j=1}^{n}(x_j - \overline{x})^2}$$

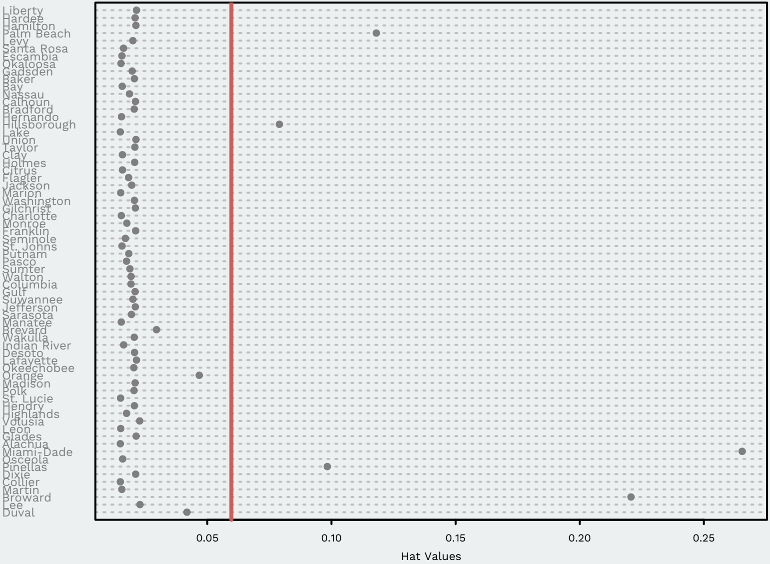
  - $\rightsquigarrow$ how far i is from the center of the $\mathbf{X}$ distribution
- Rule of thumb: examine hat values greater than $2(k+1)/n$

Buchanan hats

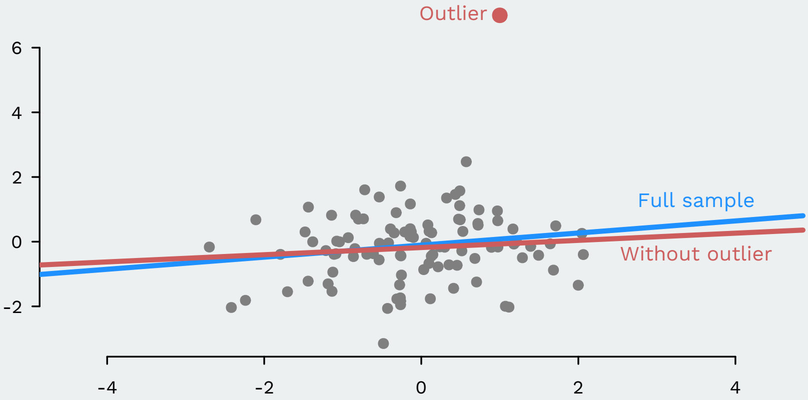
```
head(hatvalues(mod), 5)
```

```
##       1       2       3       4       5
## 0.04179 0.02285 0.22066 0.01556 0.01493
```

Buchanan hats



Outlier definition



- An outlier is a data point with very large regression errors, u_i
- Very distant from the rest of the data in the y-dimension
- Increases standard errors (by increasing $\widehat{\sigma}^2$)
- No bias if typical in the x's

Detecting outliers

- Look for big residuals, right?
  - Problem: $\hat{u}_i$ are not identically distributed.
  - Variance of the ith residual:

$$\mathbb{V}[\hat{u}_i|\mathbf{X}] = \sigma_u^2(1 - h_{ii})$$

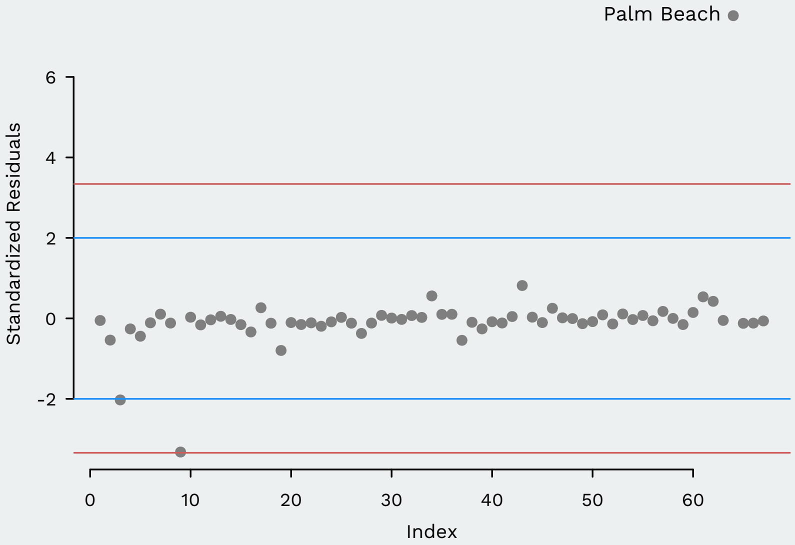
- Rescale to get standardized residuals with constant variance:

$$\hat{u}_i' = \frac{\hat{u}_i}{\widehat{\sigma}\sqrt{1 - h_{ii}}}$$

- Rule of thumb:
  - $|\hat{u}_i'| > 2$ will be relatively rare.
  - $|\hat{u}_i'| > 4 - 5$ should definitely be checked.

Buchanan outliers

```r
std.resids <- rstandard(mod)
```



Detecting outliers

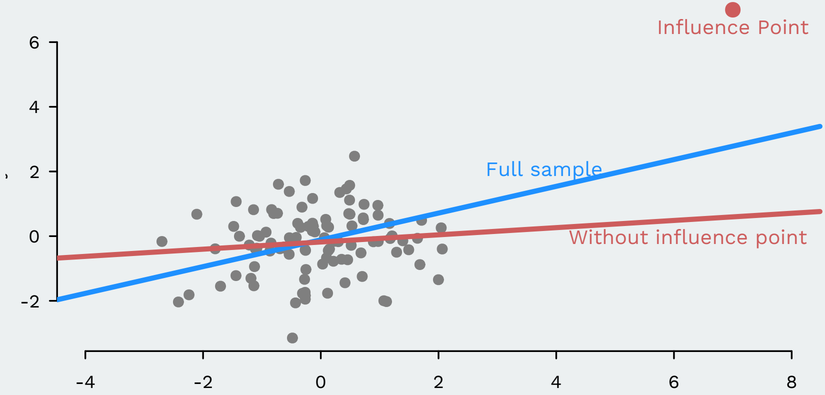
- Standardized or regular residuals are not good for detecting outliers because they might pull the regression line close to them.
- Better: leave-one-out prediction errors,
  1. Regress $\mathbf{X}_{(-i)}$ on $\mathbf{y}_{(-i)}$, where these omit unit i:

     $$\widehat{\boldsymbol{\beta}}_{(-i)} = \left(\mathbf{X}'_{(-i)}\mathbf{X}_{(-i)}\right)^{-1}\mathbf{X}'_{(-i)}\mathbf{y}_{(-i)}$$

  2. Calculate predicted value of y_i using that regression: $\tilde{y}_i = \mathbf{x}'_i\widehat{\boldsymbol{\beta}}_{(-i)}$
  3. Calculate prediction error: $\tilde{u}_i = y_i - \tilde{y}_i$
- Possible relate prediction errors to residuals:

$$\tilde{u}_i = \frac{\widehat{u}_i}{1 - h_i}$$

Influence points



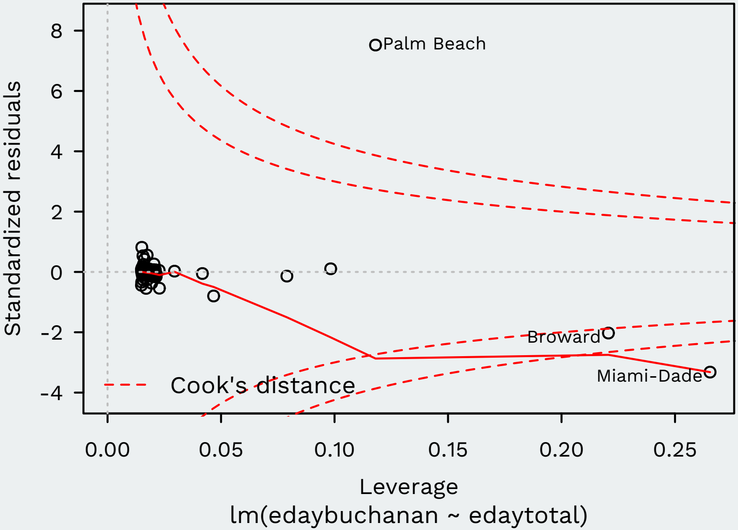
- An influence point is one that is both an outlier and a leverage point.
- Extreme in both the x and y dimensions
- Causes the regression line to move toward it (bias?)

Overall measures of influence

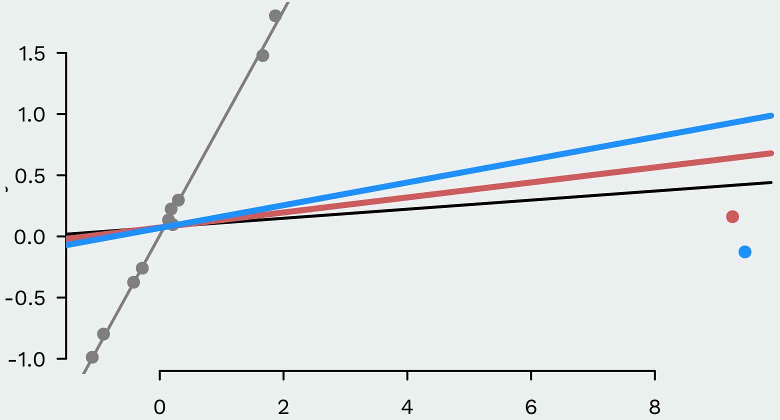
- A rough measure of influence is to look at how the difference between the fitted value and the predicted leave-one-out value: $\widehat{y}_i - \tilde{y}_i$
  - This is equivalent to $\tilde{u}_i h_i$, which is just the "outlier-ness × leverage"
- Cook's distance (`cooks.distance()`): $D_i = \frac{\tilde{u}_i^2}{(k+1)\widehat{\sigma}^2} \times h_i$
  - Basically: "normalized outlier-ness × leverage"
  - $D_i > 4/(n-k-1)$ considered "large", but cutoffs are arbitrary
- Influence plot:
  - x-axis: hat values, h_i
  - y-axis: standardized residuals, $\widehat{u}_i'$

Influence plot from lm output

```r
plot(mod, which = 5, labels.id = flvote$county)
```



Limitations of the standard tools



- What happens when there are two influence points?
- Red line drops the red influence point
- Blue line drops the blue influence point
- "Leave-one-out" approaches helps recover the line

What to do about outliers and influential units?

- Is the data corrupted?
  - Fix the observation (obvious data entry errors)
  - Remove the observation
  - Be transparent either way
- Is the outlier part of the data generating process?
  - Transform the dependent variable ($\log(y)$)
  - Use a method that is robust to outliers (robust regression, least absolute deviations)

2/ Heteroskedasticity

Review of homoskedasticity

- Remember:

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

- $\mathbb{V}[\mathbf{u}|\mathbf{X}] = \Sigma$ is the variance-covariance matrix of the errors.
- Assumptions 1-4 give us this expression for sampling variance:

$$\mathbb{V}[\hat{\boldsymbol{\beta}}|\mathbf{X}] = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\Sigma\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}$$

- Under homoskedasticity, we simplified this to:

$$\mathbb{V}[\widehat{\boldsymbol{\beta}}|\mathbf{X}] = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$$

- Replace σ^2 with estimate $\widehat{\sigma}^2$ will give us our estimate of the covariance matrix

Non-constant error variance

- Homoskedastic:

$$\mathbb{V}[\mathbf{u}|\mathbf{X}] = \sigma^2\mathbf{I} = \begin{bmatrix} \sigma^2 & 0 & 0 & \dots & 0 \\ 0 & \sigma^2 & 0 & \dots & 0 \\ & & & \vdots & \\ 0 & 0 & 0 & \dots & \sigma^2 \end{bmatrix}$$

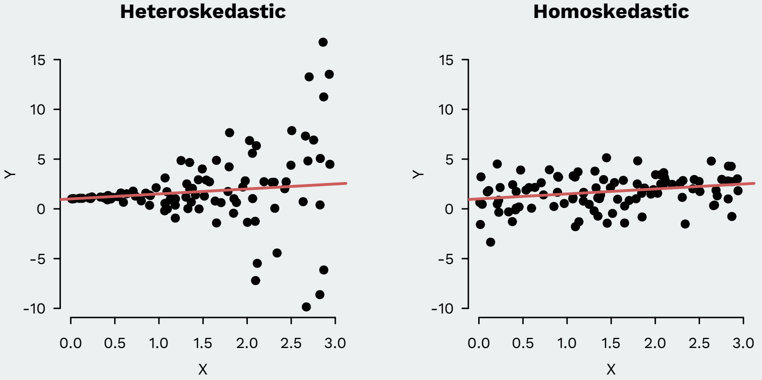
- Heteroskedastic:

$$\mathbb{V}[\mathbf{u}|\mathbf{X}] = \begin{bmatrix} \sigma_1^2 & 0 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & 0 & \dots & 0 \\ & & & \vdots & \\ 0 & 0 & 0 & \dots & \sigma_n^2 \end{bmatrix}$$

- Independent, not identical
- $\text{Cov}[u_i, u_j|\mathbf{X}] = 0$
- $\mathbb{V}[u_i|\mathbf{x}_i] = \sigma_i^2$

Violations of homoskedasticity

- Violations: magnitude of u_i differ at different levels of X_i.



Consequences of Heteroskedasticity

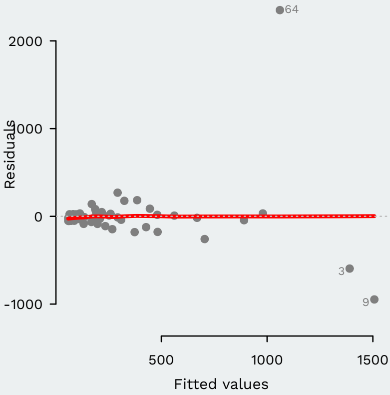
- Standard error estimates biased, likely downward
- Test statistics won't have t or F distributions
- α-level tests, the probability of Type I error $\neq \alpha$
- Coverage of $1 - \alpha$ CIs $\neq 1 - \alpha$
- OLS is not BLUE
- $\widehat{\boldsymbol{\beta}}$ still unbiased and consistent for $\boldsymbol{\beta}$

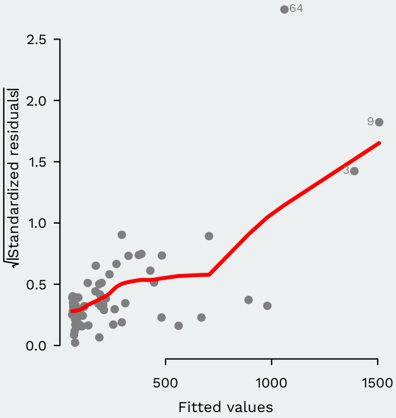
Visual diagnostics

1. Plot of residuals versus fitted values
    - In R, `plot(mod, which = 1)`
    - Residuals should have the same variance across x-axis
2. Spread location plots
    - y-axis: Square-root of the absolute value of the residuals
    - x-axis: Fitted values
    - Usually has loess trend curve, should be flat
    - In R, `plot(mod, which = 3)`

Diagnostics

```
plot(mod, which = 1, lwd = 3)
plot(mod, which = 3, lwd = 3)
```





Dealing with non-constant error variance

1. Transform the dependent variable
2. Model the heteroskedasticity using Weighted Least Squares (WLS)
3. Use an estimator of $\mathbb{V}[\widehat{\boldsymbol{\beta}}|\mathbf{X}]$ that is robust to heteroskedasticity
4. Admit we have the wrong model and use a different approach

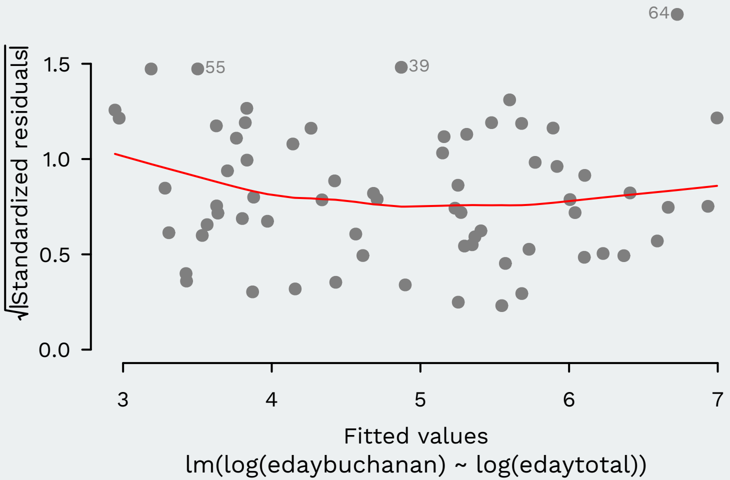
Example: Transforming Buchanan votes

```
mod2 <- lm(log(edaybuchanan) ~ log(edaytotal), data = flvote)
summary(mod2)
```

```
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -2.728      0.400   -6.83  3.5e-09 ***
## log(edaytotal)      0.729      0.038   19.15  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.469 on 65 degrees of freedom
## Multiple R-squared:  0.849,  Adjusted R-squared:  0.847
## F-statistic:  367 on 1 and 65 DF,  p-value: <2e-16
```

Example: Transformed scale-location plot

```r
plot(mod2, which = 3)
```



Weighted least squares

- Suppose that the heteroskedasticity is known up to a multiplicative constant:

$$\mathbb{V}[u_i|\mathbf{X}] = a_i\sigma^2$$

  where $a_i = a_i(\mathbf{x}_i')$ is a positive and known function of $\mathbf{x}_i'$
- WLS: multiply y_i by $1/\sqrt{a_i}$:

$$\frac{y_i}{\sqrt{a_i}} = \beta_0\frac{1}{\sqrt{a_i}} + \beta_1\frac{x_{i1}}{\sqrt{a_i}} + \cdots + \beta_k\frac{x_{ik}}{\sqrt{a_i}} + \frac{u_i}{\sqrt{a_i}}$$

WLS intuition

- Rescales errors to $u_i/\sqrt{a_i}$, which maintains zero mean error
- But makes the error variance constant again:

$$\begin{aligned}
\mathbb{V}\left[\frac{1}{\sqrt{a_i}}u_i|\mathbf{X}\right] &= \frac{1}{a_i}\mathbb{V}\left[u_i|\mathbf{X}\right] \\
&= \frac{1}{a_i}a_i\sigma^2 \\
&= \sigma^2
\end{aligned}$$

- If you know a_i, then you can use this approach to makes the model homoeskedastic and, thus, BLUE again
- When do we know a_i?

WLS procedure

- Define the weighting matrix:

$$\mathbf{W} = \begin{bmatrix} 1/\sqrt{a_1} & 0 & 0 & 0 \\ 0 & 1/\sqrt{a_2} & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 1/\sqrt{a_n} \end{bmatrix}$$

- Run the following regression:

$$\begin{aligned} \mathbf{Wy} &= \mathbf{WX}\boldsymbol{\beta} + \mathbf{Wu} \\ \mathbf{y}^* &= \mathbf{X}^*\boldsymbol{\beta} + \mathbf{u}^* \end{aligned}$$

- Run regression of $\mathbf{y}^* = \mathbf{Wy}$ on $\mathbf{X}^* = \mathbf{WX}$ and all Gauss-Markov assumptions are satisfied
- Plugging into the usual formula for $\widehat{\boldsymbol{\beta}}$:

$$\widehat{\boldsymbol{\beta}}_W = (\mathbf{X}'\mathbf{W}'\mathbf{WX})^{-1}\mathbf{X}'\mathbf{W}'\mathbf{Wy}$$

WLS example

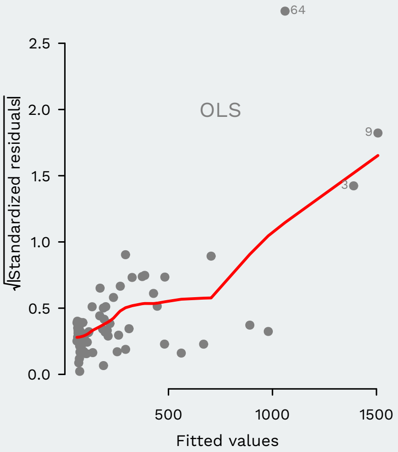
- In R, use `weights =` argument to `lm` and give the weights squared: $1/a_i$
- With the Buchanan data, maybe the variance is proportional to the total number of ballots cast:

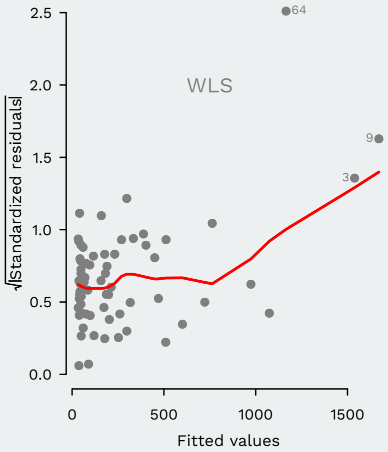
```
mod.wls <- lm(edaybuchanan ~ edaytotal, weights = 1/edaytotal, data = flvot
summary(mod.wls)
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 27.06785    8.50723    3.18   0.0022 **
## edaytotal    0.00263    0.00025   10.50  1.2e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.565 on 65 degrees of freedom
## Multiple R-squared:  0.629,	Adjusted R-squared:  0.624
## F-statistic:  110 on 1 and 65 DF,  p-value: 1.22e-15
```

Comparing WLS to OLS

```
plot(mod, which = 3, lwd = 2, sub = "")
plot(mod.wls, which = 3, lwd = 2, sub = "")
```





Heteroskedasticity consistent estimator

- Under non-constant error variance:

$$\mathbb{V}[\mathbf{u}|\mathbf{X}] = \Sigma = \begin{bmatrix} \sigma_1^2 & 0 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & 0 & \dots & 0 \\ & & & \vdots & \\ 0 & 0 & 0 & \dots & \sigma_n^2 \end{bmatrix}$$

- When $\boldsymbol{\Sigma} \neq \sigma^2\mathbf{I}$, we are stuck with this expression:

$$\mathbb{V}[\widehat{\boldsymbol{\beta}}|\mathbf{X}] = \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\Sigma\mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}$$

- White (1980) shows that we can consistently estimate this if we have an estimate of Σ:

$$\widehat{\mathbb{V}}[\widehat{\boldsymbol{\beta}}|\mathbf{X}] = \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\widehat{\Sigma}\mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}$$

- Sandwich estimator with bread $\left(\mathbf{X}'\mathbf{X}\right)^{-1}$ and meat $\mathbf{X}'\widehat{\Sigma}\mathbf{X}$

Computing HC/robust standard errors

1. Fit regression and obtain residuals $\hat{\mathbf{u}}$
2. Construct the "meat" matrix $\widehat{\Sigma}$ with squared residuals in diagonal:

$$\widehat{\Sigma} = \begin{bmatrix} \hat{u}_1^2 & 0 & 0 & \dots & 0 \\ 0 & \hat{u}_2^2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \hat{u}_n^2 \end{bmatrix}$$

3. Plug $\widehat{\Sigma}$ into sandwich formula to obtain HC/robust estimator of the covariance matrix:

$$\widehat{\mathbb{V}}[\hat{\boldsymbol{\beta}}|\mathbf{X}] = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\widehat{\Sigma}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}$$

- Small-sample corrections (called 'HC1'):

$$\widehat{\mathbb{V}}[\hat{\boldsymbol{\beta}}|\mathbf{X}] = \frac{n}{n-k-1} \cdot (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\widehat{\Sigma}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}$$

Robust SEs in Florida data

```r
coeftest(mod)
```

```
##
## t test of coefficients:
##
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 54.22945   49.14146    1.10     0.27
## edaytotal    0.00232    0.00031    7.48  2.4e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
coeftest(mod, vcovHC(mod, type = "HC0"))
```

```
##
## t test of coefficients:
##
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 54.22945   40.61283    1.34   0.1864
## edaytotal    0.00232    0.00087    2.67   0.0096 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Robust SEs with correction

```
lmtest::coeftest(mod, sandwich::vcovHC(mod, type = "HC0"))
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 54.22945   40.61283    1.34   0.1864
## edaytotal    0.00232    0.00087    2.67   0.0096 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lmtest::coeftest(mod, sandwich::vcovHC(mod, type = "HC1"))
```

```
##
## t test of coefficients:
##
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 54.229453  41.232904    1.32    0.193
## edaytotal    0.002323   0.000884    2.63    0.011 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

WLS vs. White's Estimator

- WLS:
  - With known weights, WLS is efficient
  - and $\widehat{SE}[\widehat{\boldsymbol{\beta}}_{WLS}]$ is consistent
  - but weights usually aren't known
- White's Estimator:
  - Doesn't change estimate $\widehat{\boldsymbol{\beta}}$
  - Consistent for $\mathbb{V}[\widehat{\boldsymbol{\beta}}]$ under any form of heteroskedasticity
  - Because it relies on consistency, it is a large sample result, best with large n
  - For small n, performance might be poor

3/ Nonlinearity of the regression function

Buchanan model, part 2

```
mod3 <- lm(edaybuchanan ~ edaytotal + absnbuchanan, data = fl
summary(mod3)
```

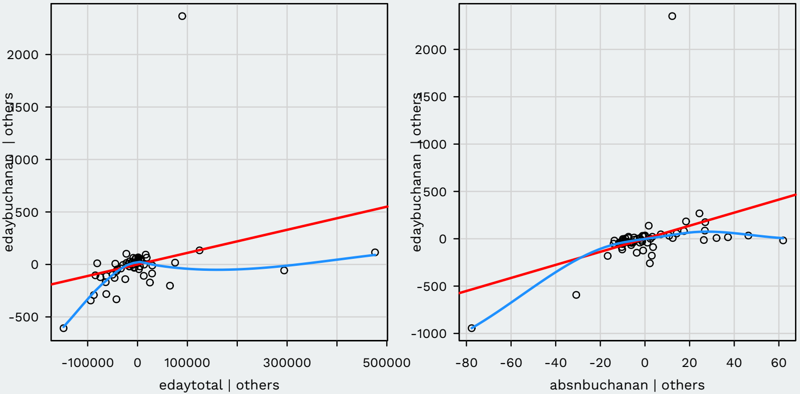
```
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -29.34807   55.19635   -0.53   0.5969
## edaytotal      0.00110    0.00048    2.29   0.0253 *
## absnbuchanan   6.89546    2.12942    3.24   0.0019 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 317 on 61 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.536,  Adjusted R-squared:  0.521
## F-statistic: 35.2 on 2 and 61 DF,  p-value: 6.71e-11
```

Added variable plot

- Need a way to visualize conditional relationship between Y and X_j
- How to construct an added variable plot:
  1. Get residuals from regression of Y on all covariates except X_j
  2. Get residuals from regression of X_j on all other covariates
  3. Plot residuals from (1) against residuals from (2)
- In R: `avPlots(model)` from the `car` package
- OLS fit to this plot will have exactly $\widehat{\beta}_j$ and 0 intercept
- Use local smoother (loess) to detect any non-linearity

Buchanan AV plot

```
par(mfrow = c(1, 2))
out <- car::avPlots(mod3, "edaytotal")
lines(loess.smooth(x = out$edaytotal[, 1], y = out$edaytotal[, 2]),
    col = "dodgerblue", lwd = 2)
out2 <- car::avPlots(mod3, "absnbuchanan")
lines(loess.smooth(x = out2$absnbuchanan[, 1], y = out2$absnbuchanan[,
    2]), col = "dodgerblue", lwd = 2)
```



How to deal with non-linearity

- Breaking up categorical variables into dummy variables
- Including interaction terms
- Including polynomial terms
- Using transformations
- Using more flexible models:
  - Generalized additive models and splines allow the data to tell us what the functional form is.
  - Complicated math, but important ideas.

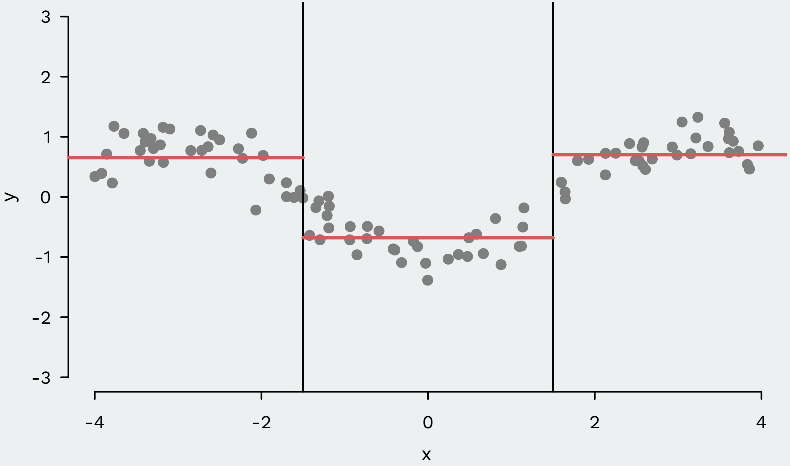
Basis functions

- Basis functions are the function of x_i that we include in the model:
  - Examples we've seen: $h_m(x_i) = x_i$, $h_m(x_i) = x_i^2$, $h_m(x_i) = \log(x_i)$
- Different basis functions will allow for different forms of non-linearity
- We could always break up X_i into bins and estimate piecewise constant:

$$h_1 = 1, \quad h_2 = \mathbb{1}(b_1 < x_i < b_2), \quad h_3 = \mathbb{1}(x_i > b_2)$$

- $b_1 < b_2$ are knots

Piecewise constant

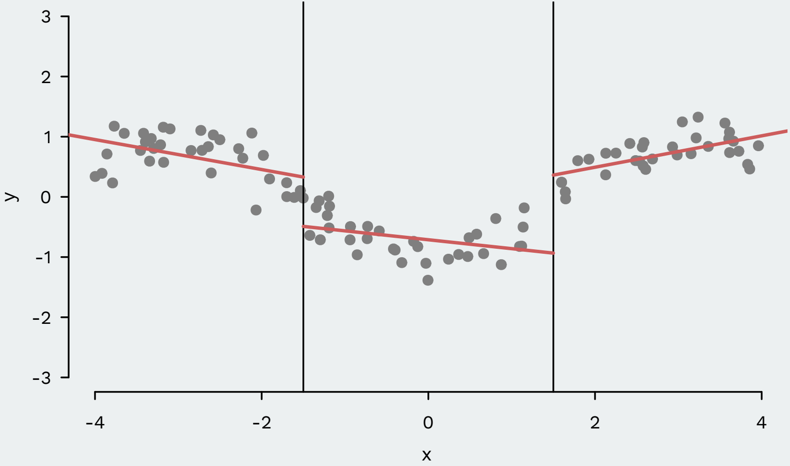


Piecewise linear

- We could allow there to be different regression lines in each bin by adding interactions:

$$
\begin{aligned}
h_1(x_i) &= 1, & h_2(x_i) &= x_i, \\
h_3(x_i) &= \mathbb{1}(b_1 < x_i < b_2), & h_4(x_i) &= x_i \mathbb{1}(b_1 < x_i < b_2), \\
h_5(x_i) &= \mathbb{1}(x_i \geq b_2), & h_6(x_i) &= x_i \mathbb{1}(x_i \geq b_2)
\end{aligned}
$$

Piecewise linear



Continuous piecewise linear

- Problem: piecewise functions are discontinuous.
- Can use clever basis functions to get continuous piecewise linear function of X_i:

$$h_1(x_i) = 1, \qquad h_2(x_i) = x_i,$$
$$h_3(x_i) = (x_i - b_1)_+, \qquad h_4(x_i) = (x_i - b_2)_+$$

- $(x_i - b_1)_+ = x_i - b_1$ when $x_i > b_1$, 0, otherwise

Why continuous?

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 (x_i - b_1)_+ + \beta_3 (x_i - b_2)_+ + u_i$$

- Value at b_1 approaching from below:

$$\beta_0 + \beta_1 b_1$$

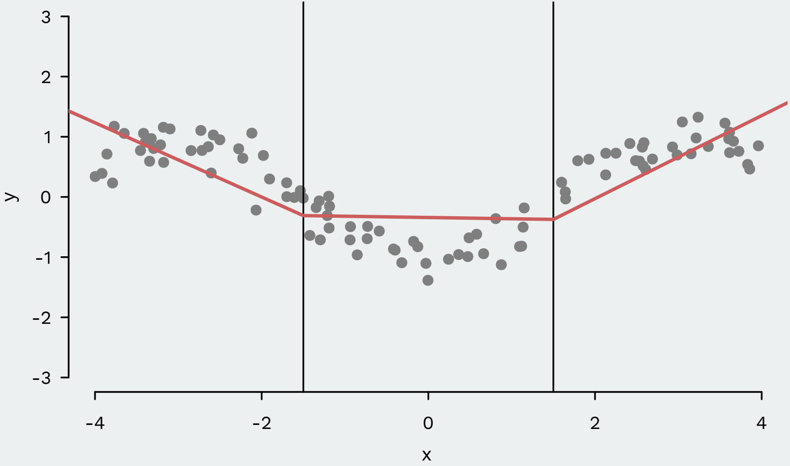
- Value at b_1 approaching from above:

$$\beta_0 + \beta_1 b_1 + \beta_2 (b_1 - b_1)_+ = \beta_0 + \beta_1 b_1$$

- Function is thus continuous at the knot points, but slopes change:
  - $\beta_1 =$ slope when $X_i < b_1$
  - $\beta_1 + \beta_2 =$ slope when $b_1 < X_i < b_2$
  - $\beta_1 + \beta_2 + \beta_3 =$ slope when $X_i > b_2$
  - Function is continuous at cutpoints

Continuous piecewise linear

```
h2 <- x
h3 <- 1 * (x > -1.5) * (x - -1.5)
h4 <- 1 * (x > 1.5) * (x - 1.5)
reg <- lm(y ~ h2 + h3 + h4)
```



Cubic splines

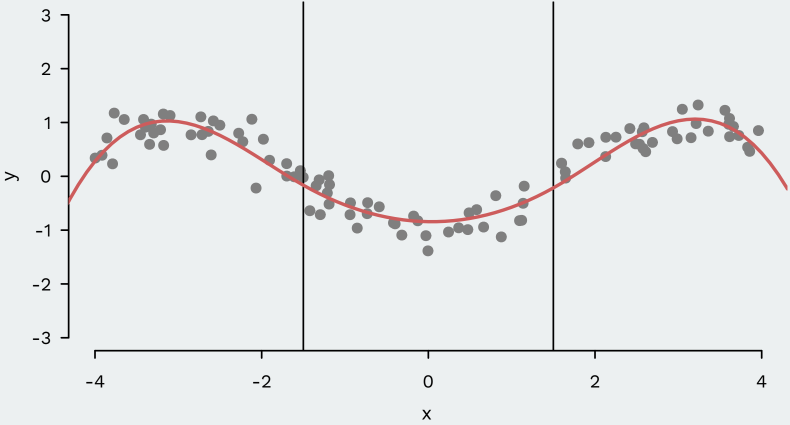
- Continuous piecewise linear has "kinks" at the knots, but we probably want "smooth" functions.
  - What does smooth mean? Continuous derivatives!
  - $\rightsquigarrow$ use higher-order polynomials in the basis functions
- Cubic spline basis: bases that produce continuous functions with continuous first and second derivatives

$$
\begin{aligned}
h_1(x_i) &= 1, & h_2(x_i) &= x_i, & h_3(x_i) &= x_i^2 \\
h_4(x_i) &= x_i^3, & h_5(x_i) &= (x_i - b_1)_+^3, & h_6(x_i) &= (x_i - b_2)_+^3
\end{aligned}
$$

- Basic idea: local polynomial regression (between knots) that have to connect and be smooth at the knots.
  - Ensure this by allowing only the coefficient on the cubic term to change at the knot point.

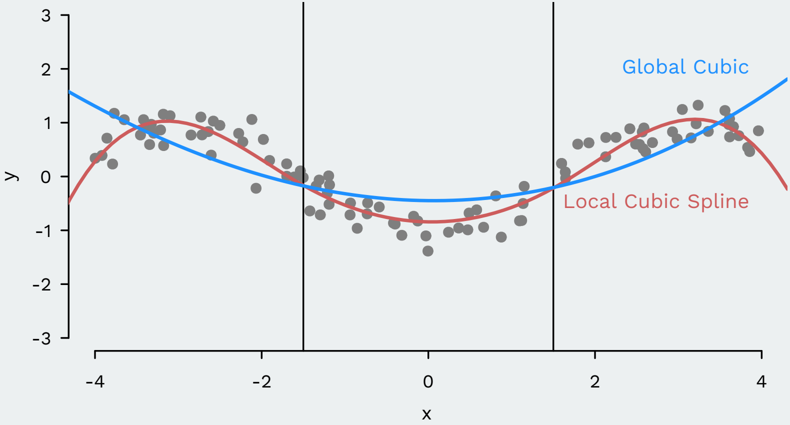
Cubic spline

```
h2 <- x
h3 <- x^2
h4 <- x^3
h5 <- 1 * (x > -1.5) * (x - -1.5)^3
h6 <- 1 * (x > 1.5) * (x - 1.5)^3
reg <- lm(y ~ h2 + h3 + h4 + h5 + h6)
```



Cubic spline vs global

```r
h2 <- x
h3 <- x^2
h4 <- x^3
rr <- lm(y ~ h2 + h3 + h4)
```



Knotty problems

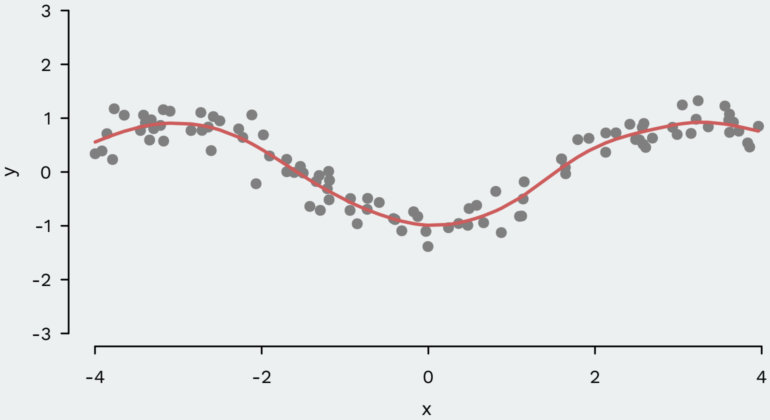
- Any function can be approximated as we increase the number of knot points.
- How to choose the number/location of knot points?
  - More knot points $\rightsquigarrow$ "rougher" function, less in-sample bias, more variance.
  - Fewer knot points $\rightsquigarrow$ "smoother" function, more in-sample bias, less variance.
- In-sample fit might be great, out-of-sample fit might be terrible.
- More general smoothing approaches have different ways of representing this trade-off other than knots.

Cross-validation

- General strategy for bias-variance trade-offs: cross-validation.
- Set aside units to test out-of-sample prediction
- Cross-validation procedure:
  1. Choose a number of evenly spread knots, b.
  2. Withhold unit i, estimate the CEF of y_i given x_i using a cubic spline with b knots.
  3. Get predicted value for i, $\hat{y}_{ib}^{-i}$ and caculate squared prediction error: $(y_i - \hat{y}_{ib}^{-i})^2$.
  4. Repeat 2-3 for each observation and take that average to get the MSE with b knots.
  5. Repeat 1-4 for different values of b and choose the value of b that has the lowest MSE.

Automatic knot selection

```
smth <- smooth.spline(x, y)
plot(x, y, ylim = c(-3, 3), pch = 19, col = "grey50", bty = "n")
lines(smth, col = "indianred", lwd = 2)
```



Generalized additive models

- Generalized additive models (GAMs) allow you to estimate the spline of any particular variable in the regression.
  - Each spline is additive: $y_i = f_1(x_{i1}) + f_x(x_{i2}) + u_i$
- Can plot the AV-plot of the spline to get a sense for the nonlinearity of the functional form.
- Use cross-validation to select the number of knots

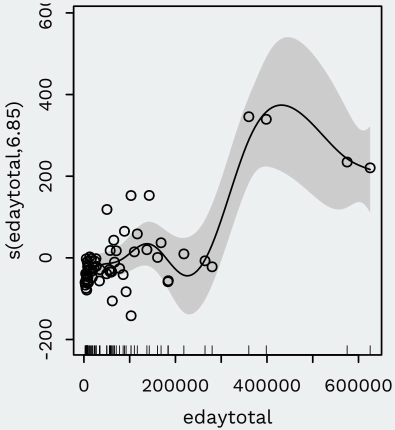
GAM example fit

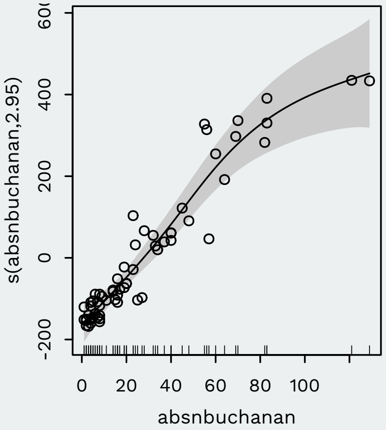
```r
## library(mgcv)  ## GAM package
out <- gam(edaybuchanan ~ s(edaytotal) + s(absnbuchanan), data = flvote,
    subset = county != "Palm Beach")
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## edaybuchanan ~ s(edaytotal) + s(absnbuchanan)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   221.84       6.41    34.6   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                  edf Ref.df    F p-value
## s(edaytotal)    6.85   7.82 10.6 1.6e-09 ***
## s(absnbuchanan) 2.95   3.64 22.6 1.6e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   0.95   Deviance explained = 95.8%
## GCV =   3129  Scale est. = 2592.3    n = 63
```

Example: generalized additive models

```
plot(out, shade = TRUE, residual = TRUE, pch = 1)
```





Summary

- For influential points, and nonlinearity:
  - Check your data! `summary()`, `plot()`, etc
  - Use transformations to make assumptions more plausible
  - Weaken linearity when you need to.