

# PLSC 503 – Spring 2023

## Variances, Collinearity, etc.

February 13, 2023

# Variances: Why We Care

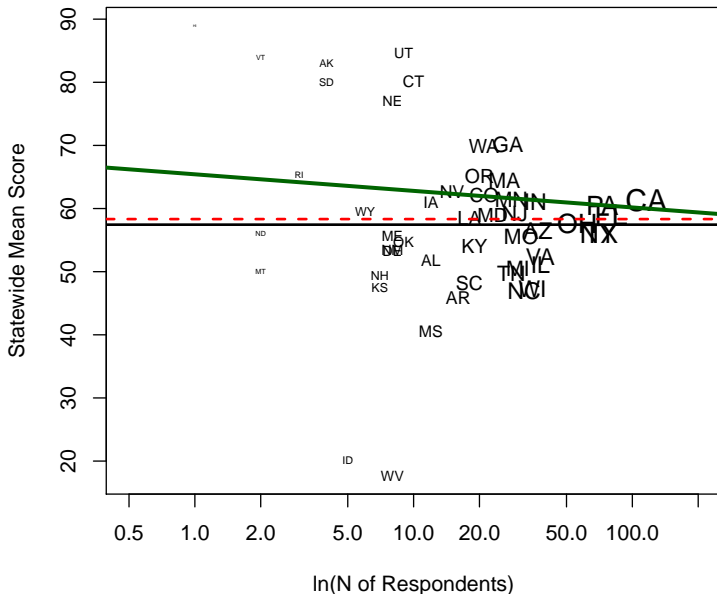
2016 ANES pilot study “feeling thermometer” toward gays and lesbians ( $N = 1200$ ):

```
> summary(ANES$ftgay)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
  0.00  40.50   54.00   57.45  88.50  100.00     1
```

Suppose we wanted to create aggregate measures, by state ( $N = 51$ ). We would get:

```
> summary(StateFT)
  State          Nresp          meantherm
Length:50      Min.   : 1.00      Min.   :17.62
Class :character 1st Qu.: 8.00      1st Qu.:51.33
Mode  :character Median : 18.00      Median :57.11
              Mean   : 24.00      Mean   :58.33
              3rd Qu.: 30.75      3rd Qu.:62.55
              Max.   :116.00      Max.   :89.00
```

# Variances: Why We Care



# Variances: A Generalization

Start with:

$$Y_i = \mathbf{X}_i\boldsymbol{\beta} + u_i$$

with:

$$\text{Var}(u_i) = \sigma^2/w_i$$

with  $w_{iu}$  known.

# Weighted Least Squares

WLS now minimizes:

$$\text{RSS} = \sum_{i=1}^N w_i (Y_i - \mathbf{X}_i \boldsymbol{\beta}).$$

which gives:

$$\begin{aligned}\hat{\boldsymbol{\beta}}_{\text{WLS}} &= [\mathbf{X}'(\sigma^2 \boldsymbol{\Omega})^{-1} \mathbf{X}]^{-1} \mathbf{X}'(\sigma^2 \boldsymbol{\Omega})^{-1} \mathbf{Y} \\ &= [\mathbf{X}' \mathbf{W}^{-1} \mathbf{X}]^{-1} \mathbf{X}' \mathbf{W}^{-1} \mathbf{Y}\end{aligned}$$

where:

$$\mathbf{W} = \begin{bmatrix} \frac{\sigma^2}{w_1} & 0 & \dots & 0 \\ 0 & \frac{\sigma^2}{w_2} & \dots & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \dots & 0 & \frac{\sigma^2}{w_N} \end{bmatrix}$$

The variance-covariance matrix is:

$$\begin{aligned}\text{Var}(\hat{\beta}_{WLS}) &= \sigma^2(\mathbf{X}'\mathbf{\Omega}^{-1}\mathbf{X})^{-1} \\ &\equiv (\mathbf{X}'\mathbf{W}^{-1}\mathbf{X})^{-1}\end{aligned}$$

A common case is:

$$\text{Var}(u_i) = \sigma^2 \frac{1}{N_i}$$

where  $N_i$  is the number of observations upon which (aggregate) observation  $i$  is based.

## “Robust” Variance Estimators

Recall that, if  $\sigma_i^2 \neq \sigma_j^2 \forall i \neq j$ ,

$$\begin{aligned}\text{Var}(\beta_{\text{Het.}}) &= (\mathbf{X}'\mathbf{X})^{-1}(\mathbf{X}'\mathbf{W}^{-1}\mathbf{X})(\mathbf{X}'\mathbf{X})^{-1} \\ &= (\mathbf{X}'\mathbf{X})^{-1} \mathbf{Q} (\mathbf{X}'\mathbf{X})^{-1}\end{aligned}$$

where  $\mathbf{Q} = (\mathbf{X}'\mathbf{W}^{-1}\mathbf{X})$  and  $\mathbf{W} = \sigma^2\mathbf{\Omega}$ .

We can rewrite  $\mathbf{Q}$  as

$$\begin{aligned}\mathbf{Q} &= \sigma^2(\mathbf{X}'\mathbf{\Omega}^{-1}\mathbf{X}) \\ &= \sum_{i=1}^N \sigma_i^2 \mathbf{x}_i \mathbf{x}_i'\end{aligned}$$

Estimate  $\hat{\mathbf{Q}}$  as:

$$\hat{\mathbf{Q}} = \sum_{i=1}^N \hat{u}_i^2 \mathbf{x}_i \mathbf{x}_i'$$

Yields:

$$\begin{aligned} \widehat{\text{Var}(\boldsymbol{\beta})}_{\text{Robust}} &= (\mathbf{X}'\mathbf{X})^{-1}(\mathbf{X}'\hat{\mathbf{Q}}^{-1}\mathbf{X})(\mathbf{X}'\mathbf{X})^{-1} \\ &= (\mathbf{X}'\mathbf{X})^{-1} \left[ \mathbf{X}' \left( \sum_{i=1}^N \hat{u}_i^2 \mathbf{x}_i \mathbf{x}_i' \right)^{-1} \mathbf{X} \right] (\mathbf{X}'\mathbf{X})^{-1} \end{aligned}$$



“Robust” VCV estimates:

- are heteroscedasticity-consistent, but
- are biased in small samples, and
- are less efficient than “naive” estimates when  $\text{Var}(u) = \sigma^2 \mathbf{I}$ .

# “Clustering”

Huber / White

?????????

WLS / GLS

I know very little  
about my error  
variances...

I know a great  
deal about my  
error variances...

A common case:

$$Y_{ij} = \mathbf{X}_{ij}\beta + u_{ij}$$

with

$$\sigma_{ij}^2 = \sigma_{ik}^2.$$

“Robust, clustered” estimator:

$$\widehat{\text{Var}}(\beta)_{\text{Clustered}} = (\mathbf{X}'\mathbf{X})^{-1} \left\{ \mathbf{X}' \left[ \sum_{i=1}^N \left( \sum_{j=1}^{n_j} \hat{u}_{ij}^2 \mathbf{X}_{ij} \mathbf{X}_{ij}' \right) \right]^{-1} \mathbf{X} \right\} (\mathbf{X}'\mathbf{X})^{-1}$$

# Robust / Clustered SEs: A Simulation

```
url_robust <- "https://raw.githubusercontent.com/IsidoreBeautrelet/economictheoryblog/master/robust_summary.R"
eval(parse(text = getURL(url_robust, ssl.verifypeer = FALSE)),
      envir=.GlobalEnv)
```

```
> set.seed(7222009)
> X <- rnorm(10)
> Y <- 1 + X + rnorm(10)
> df10 <- data.frame(ID=seq(1:10),X=X,Y=Y)
>
> fit10 <- lm(Y~X,data=df10)
> summary(fit10)
```

Residuals:

Min	1Q	Median	3Q	Max
-1.12328	-0.65321	-0.05073	0.43937	1.81661

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.8438	0.3020	2.794	0.0234 *
X	0.3834	0.3938	0.974	0.3588

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9313 on 8 degrees of freedom

Multiple R-squared: 0.1059, Adjusted R-squared: -0.005832

F-statistic: 0.9478 on 1 and 8 DF, p-value: 0.3588

```
> rob10 <- vcovHC(fit10,type="HC1")
> sqrt(diag(rob10))
(Intercept)      X
  0.2932735    0.2859552
```

# Robust / Clustered SEs: A Simulation (continued)

```
> # "Clone" each observation 100 times
>
> df1K <- df10[rep(seq_len(nrow(df10)), each=100),]
> df1K <- pdata.frame(df1K, index="ID")
>
> fit1K <- lm(Y~X,data=df1K)
> summary(fit1K)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.84383	0.02704	31.20	<2e-16 ***
X	0.38341	0.03526	10.87	<2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8338 on 998 degrees of freedom  
Multiple R-squared: 0.1059, Adjusted R-squared: 0.105  
F-statistic: 118.2 on 1 and 998 DF, p-value: < 2.2e-16

```
> summary(fit1K, cluster="ID")
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.8438	0.2766	3.050	0.00235 **
X	0.3834	0.2697	1.421	0.15551

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8338 on 998 degrees of freedom  
Multiple R-squared: 0.1059, Adjusted R-squared: 0.105  
F-statistic: 2.02 on 1 and 9 DF, p-value: 0.1889

# Expanded State-Level ANES Example

```
> describe(StateData)
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
State*	1	50	25.50	14.58	25.50	25.50	18.53	1.00	50.00	49.00	0.00	-1.27	2.06
NResp	2	50	24.00	23.74	18.00	19.48	16.31	1.00	116.00	115.00	1.79	3.34	3.36
LGBTTherm	3	50	58.33	13.74	57.11	58.11	8.51	17.62	89.00	71.38	-0.22	1.40	1.94
MeanCons	4	50	3.97	0.77	4.00	3.98	0.55	1.50	5.60	4.10	-0.47	1.28	0.11
MeanAge	5	50	4.74	0.64	4.78	4.74	0.43	3.10	6.50	3.40	0.11	1.10	0.09
MeanEducation	6	50	3.25	0.52	3.22	3.22	0.41	2.33	5.00	2.67	0.84	1.44	0.07
BornAgainProp	7	50	0.28	0.18	0.25	0.28	0.19	0.00	0.72	0.72	0.11	-0.62	0.02

## OLS:

```
> OLSfit<-with(StateData,lm(LGBTTherm~MeanCons))
> summary(OLSfit)
```

### Residuals:

Min	1Q	Median	3Q	Max
-39.01	-5.71	0.56	5.20	31.02

### Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	101.01	8.32	12.14	3.0e-16 ***
MeanCons	-10.76	2.06	-5.23	3.7e-06 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11.1 on 48 degrees of freedom

Multiple R-squared: 0.363, Adjusted R-squared: 0.349

F-statistic: 27.3 on 1 and 48 DF, p-value: 0.00000373

# WLS, Weighting by $\ln(N)$ of Respondents)

```
> WLSfit<-with(StateData,lm(LGBTTherm~MeanCons,weights=log(NResp)))  
> summary(WLSfit)
```

Call:

```
lm(formula = LGBTTherm ~ MeanCons, weights = log(NResp))
```

Weighted Residuals:

Min	1Q	Median	3Q	Max
-54.85	-6.96	2.18	7.34	30.17

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	98.13	8.30	11.82	1.1e-15 ***
MeanCons	-10.30	2.07	-4.98	8.9e-06 ***

---

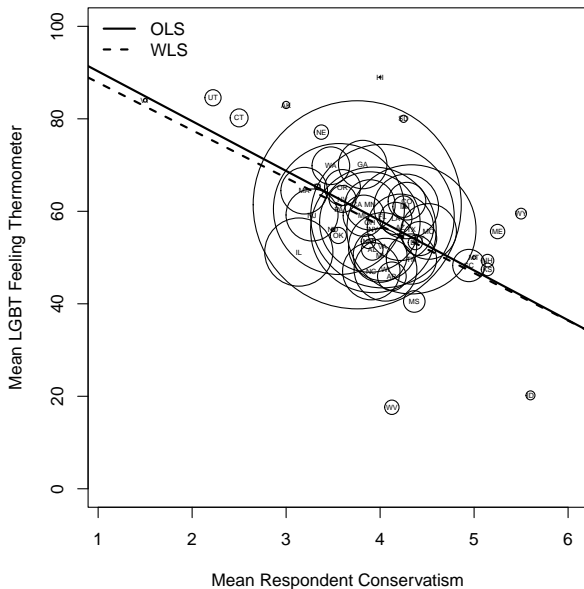
Signif. codes: 0 \*\*\* 0.001 \*\* 0.01 \* 0.05 . 0.1 1

Residual standard error: 15.1 on 47 degrees of freedom

Multiple R-squared: 0.346, Adjusted R-squared: 0.332

F-statistic: 24.8 on 1 and 47 DF, p-value: 0.00000888

# Plot of LGBTTherm Against MeanCons, Weighted by NResp





# A Multivariate Model: "Robust" SEs

```
> library(car)
> OLS<-lm(LGBTTherm~MeanCons+MeanAge+MeanEducation+BornAgainProp,data=StateData)
> summary(OLS)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	71.646	16.577	4.32	0.000084	***
MeanCons	-7.926	2.544	-3.12	0.0032	**
MeanAge	-2.669	2.557	-1.04	0.3022	
MeanEducation	9.477	3.384	2.80	0.0075	**
BornAgainProp	-0.227	10.560	-0.02	0.9829	

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.6 on 45 degrees of freedom

Multiple R-squared: 0.459, Adjusted R-squared: 0.411

F-statistic: 9.54 on 4 and 45 DF, p-value: 0.0000113

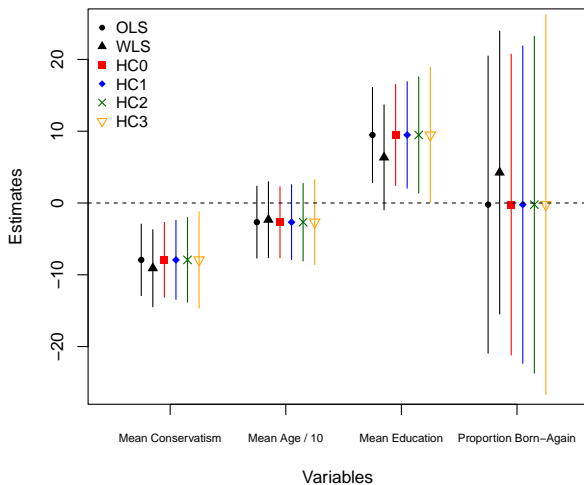
```
> hccm(OLS,type="hc0") # "HC0" var-cov matrix
```

	(Intercept)	MeanCons	MeanAge	MeanEducation	BornAgainProp
(Intercept)	328.54	-21.669	-22.309	-45.510	53.844
MeanCons	-21.67	7.094	-1.223	2.270	-23.004
MeanAge	-22.31	-1.223	6.373	-1.251	-1.276
MeanEducation	-45.51	2.270	-1.251	12.901	1.640
BornAgainProp	53.84	-23.004	-1.276	1.640	114.459

```
> sqrt(diag(hccm(OLS,type="hc0")))) # "HC0" robust SEs
```

(Intercept)	MeanCons	MeanAge	MeanEducation	BornAgainProp
18.126	2.664	2.525	3.592	10.699

# $\hat{\beta}$ s and 95% CIs: Various Models



# Cases, Variables, and Collinearity

OLS (and regression methods more generally) requires:

- $\mathbf{X}$  is full column rank.
- $N > K$ .
- “Sufficient” variability in  $\mathbf{X}$ .

# “Perfect” Multicollinearity

Formally: There cannot be any set of  $\lambda$ s such that:

$$\lambda_0 \mathbf{1} + \lambda_1 \mathbf{X}_1 + \dots + \lambda_K \mathbf{X}_K = \mathbf{0}$$

If there was, it would imply

$$\mathbf{X}_j = \frac{-\lambda_0}{\lambda_j} \mathbf{1} + \frac{-\lambda_1}{\lambda_j} \mathbf{X}_1 + \dots + \frac{-\lambda_K}{\lambda_j} \mathbf{X}_K$$

which means

$$\begin{aligned} Y &= \beta_0 \mathbf{1} + \beta_1 \mathbf{X}_1 + \dots + \beta_j \mathbf{X}_j + \dots + \beta_K \mathbf{X}_K + \mathbf{u} \\ &= \beta_0 \mathbf{1} + \beta_1 \mathbf{X}_1 + \dots + \beta_j \left( \frac{-\lambda_0}{\lambda_j} \mathbf{1} + \frac{-\lambda_1}{\lambda_j} \mathbf{X}_1 + \dots + \frac{-\lambda_K}{\lambda_j} \mathbf{X}_K \right) + \dots + \beta_K \mathbf{X}_K + \mathbf{u} \\ &= \left[ \beta_0 + \beta_j \left( \frac{-\lambda_0}{\lambda_j} \right) \right] \mathbf{1} + \left[ \beta_1 + \beta_j \left( \frac{-\lambda_1}{\lambda_j} \right) \right] \mathbf{X}_1 + \dots + \left[ \beta_K + \beta_j \left( \frac{-\lambda_K}{\lambda_j} \right) \right] \mathbf{X}_K + \mathbf{u} \\ &= \left( \beta_0 + \frac{\gamma_0}{\lambda_j} \right) \mathbf{1} + \left( \beta_1 + \frac{\gamma_1}{\lambda_j} \right) \mathbf{X}_1 + \dots + \left( \beta_K + \frac{\gamma_K}{\lambda_j} \right) \mathbf{X}_K + \mathbf{u} \end{aligned}$$

```
> Africa$newgdp<-(Africa$gdppppd-mean(Africa$gdppppd))*1000

> fit<-with(Africa, lm(adrate~gdppppd+newgdp+healthexp+subsaharan+
+                      muslperc+literacy))
> summary(fit)
```

Call:

```
lm(formula = adrate ~ gdppppd + newgdp + healthexp + subsaharan +
    muslperc + literacy)
```

Residuals:

Min	1Q	Median	3Q	Max
-15.291	-4.329	-1.412	2.723	20.682

Coefficients: (1 not defined because of singularities)

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-7.78020	10.33872	-0.753	0.4565
gdppppd	0.36142	0.58214	0.621	0.5385
newgdp	NA	NA	NA	NA
healthexp	1.87001	0.75667	2.471	0.0182 *
subsaharanSub-Saharan	3.64354	4.54163	0.802	0.4275
muslperc	-0.07908	0.05967	-1.325	0.1932
literacy	0.12445	0.09867	1.261	0.2151

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.665 on 37 degrees of freedom

Multiple R-squared: 0.4782, Adjusted R-squared: 0.4077

F-statistic: 6.782 on 5 and 37 DF, p-value: 0.0001407

- Perfect multicollinearity is terrible, but
- Perfect multicollinearity not a problem at all.

$$N > K \dots$$

Statistically,

- we lack sufficient degrees of freedom to identify  $\hat{\beta}$ .
- $\hat{\beta}$  is “overdetermined.”

Conceptually:

- Variables  $>$  Cases means
- ...no unique conclusion about explanatory / causal factors.



# $N = K$ in Practice

```
> smallAfrica<-subset(Africa,subsaharan=="Not Sub-Saharan")
> fit2<-with(smallAfrica,lm(adrate~gdppppd+healthexp+muslperc+
+                           literacy+war))
> summary(fit2)
```

Call:

```
lm(formula = adrate ~ gdppppd + healthexp + muslperc + literacy +
    war)
```

Residuals:

ALL 6 residuals are 0: no residual degrees of freedom!

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-0.12430	NA	NA	NA
gdppppd	-0.97906	NA	NA	NA
healthexp	-0.45166	NA	NA	NA
muslperc	0.01413	NA	NA	NA
literacy	0.09512	NA	NA	NA
war	-0.96429	NA	NA	NA

Residual standard error: NaN on 0 degrees of freedom

Multiple R-squared: 1, Adjusted R-squared: NaN

F-statistic: NaN on 5 and 0 DF, p-value: NA

# High (Non-Perfect) Multicollinearity

Recall that

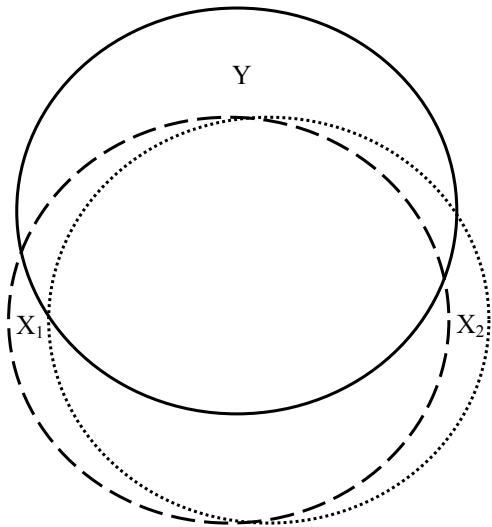
$$\widehat{\text{Var}(\hat{\beta})} = \hat{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1}$$

We can write the  $k$ th diagonal element of  $(\mathbf{X}'\mathbf{X})^{-1}$  as:

$$\frac{1}{(\mathbf{X}'_k \mathbf{X}_k)(1 - \hat{R}_k^2)}$$

where  $\hat{R}_k^2$  is the  $R^2$  from the regression of  $\mathbf{X}_k$  on all the other variables in  $\mathbf{X}$ .

# The Obligatory Venn Diagram



# High (Non-Perfect) Multicollinearity

## Things to understand:

1. Multicollinearity is a *sample problem*.
2. Multicollinearity is a matter of *degree*.

# Near-Perfect Collinearity: An Example

Consider:

$$\text{HIV}_i = \beta_0 + \beta_1(\text{Civil War}_i) + \beta_2(\text{Intensity}_i) + u_i$$

```
> with(Africa, table(internalwar,intensity))
```

	intensity			
internalwar	0	1	2	3
0	30	0	0	0
1	0	6	2	5

Table: Three Models

	<i>Dependent variable:</i>		
	adrate		
	(1)	(2)	(3)
internalwar	-4.459 (3.274)		-2.849 (6.682)
intensity		-1.955 (1.481)	-0.837 (3.018)
Constant	10.713*** (1.800)	10.502*** (1.734)	10.713*** (1.821)
Observations	43	43	43
R <sup>2</sup>	0.043	0.041	0.045
Adjusted R <sup>2</sup>	0.020	0.017	-0.003
Residual Std. Error	9.860 (df = 41)	9.873 (df = 41)	9.973 (df = 40)
F Statistic	1.855 (df = 1; 41)	1.743 (df = 1; 41)	0.945 (df = 2; 40)

Note:

\*p&lt;0.1; \*\*p&lt;0.05; \*\*\*p&lt;0.01

# (Near-Perfect) Multicollinearity: Detection

Symptoms:

1. *High  $R^2$ , but nonsignificant coefficients.*
2. *High pairwise correlations among independent variables.*
3. *High partial correlations among the **X**s.*
4. *VIF and Tolerance.*

If  $\hat{R}_k^2 = 0$ , then

$$\widehat{\text{Var}}(\hat{\beta}_k) = \frac{\hat{\sigma}^2}{\mathbf{X}'_k \mathbf{X}_k};$$

So:

$$\text{VIF}_k = \frac{1}{1 - \hat{R}_k^2}$$

$$\text{Tolerance} = \frac{1}{\text{VIF}_k}$$

Rule of Thumb:  $\text{VIF} > 10$  is a problem...



Don't:

- **Blindly drop covariates!!!**
- Restrict  $\beta$ s...

Do:

- **Add data.**
- **Transform the covariates**
  - Data reduction
  - First differences
  - Orthogonalize
- **Shrinkage / Regularization Methods**

# What To Do? Shrinkage Methods

OLS is:

$$\begin{aligned}\text{MSE} &= E\{[\mathbf{Y} - E(\mathbf{Y})]^2\} \\ &= E[(Y_i - \mathbf{X}_i\hat{\beta})^2] \\ &= [Y_i - E(\mathbf{X}_i\hat{\beta})]^2 + \{E[(\mathbf{X}_i\hat{\beta}) - E(\mathbf{X}_i\hat{\beta})]\}^2 \\ &= (\text{Bias})^2 + \text{Variance}\end{aligned}$$

“Ridge regression”:

$$\hat{\beta}^R = (\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}'\mathbf{Y}$$

- Biases  $\hat{\beta}$ , but
- Increases the (perceived) independent variability in  $\mathbf{X}$
- Yields:

$$\widehat{\text{Var}}(\hat{\beta}_\ell^R) = \frac{\hat{\sigma}^2}{(\mathbf{X}'_\ell \mathbf{X}_\ell + \lambda)(1 - R_\ell^2)}$$

# What To Do? Lasso...

“LASSO” = “Least Absolute Shrinkage and Selection Operator.”

- Formally:

$$\min_{\beta} \left\{ \frac{1}{N} \sum_{i=1}^N (Y_i - \mathbf{x}_i \beta)^2 \right\} \text{ subject to } \sum_{j=1}^p |\beta_j| \leq t.$$

- Combines variable selection and shrinkage...
- Like ridge regression, but with some  $\hat{\beta}$ s set to zero
- Reduces overfitting + makes the model more interpretable

# OLS, Ridge Regression, Lasso, & Elastic Net

## Objective / Loss Functions:

$$\text{OLS} = \sum_{i=1}^N \left( Y_i - \sum_{k=1}^k \mathbf{x}_{ik} \beta_k \right)^2$$

$$\text{LASSO} = \sum_{i=1}^N \left( Y_i - \sum_{k=1}^k \mathbf{x}_{ij} \beta_k \right)^2 + \lambda \sum_{k=1}^k |\beta_k|$$

$$\text{Ridge Regression} = \sum_{i=1}^N \left( Y_i - \sum_{k=1}^k \mathbf{x}_{ik} \beta_k \right)^2 + \lambda \sum_{k=1}^k \beta_k^2$$

$$\text{Elastic Net} = \sum_{i=1}^N \left( Y_i - \sum_{k=1}^k \mathbf{x}_{ik} \beta_k \right)^2 + \lambda \left[ (1 - \alpha) \sum_{k=1}^k \beta_k^2 + \alpha \sum_{k=1}^k |\beta_k| \right]$$

# Shrinkage Methods Using `glmnet`

The `glmnet` package fits (generalized) linear models with regularization.

- Model controlled by  $\alpha$ :
  - $\alpha = 0 \rightarrow$  ridge regression
  - $\alpha = 1 \rightarrow$  lasso
  - $0 < \alpha < 1 \rightarrow$  elastic net
- Fits multiple models over a range of values of  $\lambda$ , and
- Allows for selection of  $\lambda$  via  $k$ -fold cross-validation
- Plots, diagnostics, etc.

# Example: Impeachment

```
> summary(impeachment)
```

name		state	district		votesum
Length:433		Length:433	Min. : 1	Min. :0.00	
Class :character		Class :character	1st Qu.: 3	1st Qu.:0.00	
Mode :character		Mode :character	Median : 6	Median :2.00	
			Mean :10	Mean :1.85	
			3rd Qu.:13	3rd Qu.:4.00	
			Max. :52	Max. :4.00	

pctbl96		unionpct	clint96	GOPmember	ADA98
Min. : 0.0		Min. :0.0257	Min. :26.0	Min. :0.000	Min. : 0.0
1st Qu.: 2.0		1st Qu.:0.0930	1st Qu.:42.0	1st Qu.:0.000	1st Qu.: 5.0
Median : 5.4		Median :0.1690	Median :48.0	Median :1.000	Median : 30.0
Mean :11.9		Mean :0.1636	Mean :50.3	Mean :0.527	Mean : 46.3
3rd Qu.:14.0		3rd Qu.:0.2150	3rd Qu.:57.0	3rd Qu.:1.000	3rd Qu.: 90.0
Max. :74.0		Max. :0.3733	Max. :94.0	Max. :1.000	Max. :100.0

# Regression!

```
> # Standardize all the variables:
>
> ImpStd<-data.frame(scale(impeachment[,4:9]))
> cor(ImpStd)
```

	votesum	pctbl96	unionpct	clint96	GOPmember	ADA98
votesum	1.0000	-0.28765	-0.26199	-0.6408	0.9198	-0.9280
pctbl96	-0.2876	1.00000	-0.09394	0.6165	-0.3091	0.3029
unionpct	-0.2620	-0.09394	1.00000	0.3331	-0.1941	0.2756
clint96	-0.6408	0.61651	0.33305	1.0000	-0.6120	0.6703
GOPmember	0.9198	-0.30911	-0.19406	-0.6120	1.0000	-0.9392
ADA98	-0.9280	0.30288	0.27563	0.6703	-0.9392	1.0000

```
> # OLS w/o intercept:
>
> fit<-with(ImpStd,lm(votesum~pctbl96+unionpct+clint96+GOPmember+ADA98-1))
> summary(fit)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
pctbl96	0.0301	0.0233	1.29	0.199
unionpct	-0.0212	0.0193	-1.09	0.274
clint96	-0.0650	0.0301	-2.16	0.031 *
GOPmember	0.4367	0.0492	8.88	<2e-16 ***
ADA98	-0.4775	0.0530	-9.01	<2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.343 on 428 degrees of freedom

Multiple R-squared: 0.883, Adjusted R-squared: 0.882

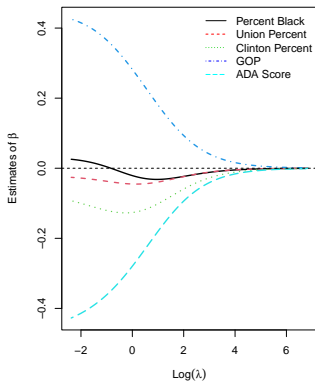
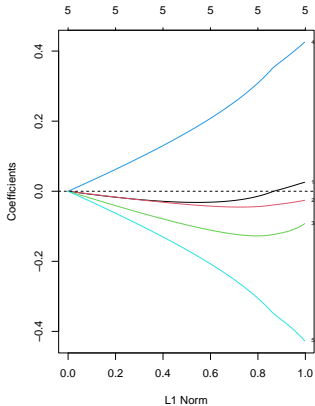
F-statistic: 648 on 5 and 428 DF, p-value: <2e-16

```
> vif(fit)
```

	pctbl96	unionpct	clint96	GOPmember	ADA98
	1.998	1.371	3.313	8.878	10.292

# Ridge Regression

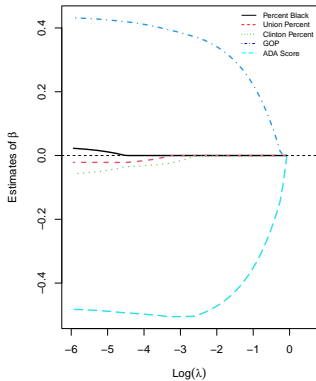
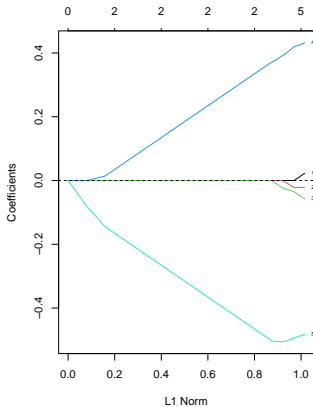
```
> # Ridge regression:  
>  
> X<-ImpStd[,2:6] # Predictors  
> Y<-ImpStd[,1]  # Response  
>  
> ridge.fit<-glmnet(X,Y,alpha=0)
```





# Lasso Regression

```
> # Lasso regression:  
>  
> lasso.fit<-glmnet(X,Y,alpha=1)
```



# Getting $\lambda$ Via Cross-Validation: Ridge Regression

```
> # Ridge regression:
>
> ridge.cv<-cv.glmnet(as.matrix(X),as.matrix(Y),alpha=0,intercept=FALSE)
> ridge.cv
```

Call: `cv.glmnet(x = as.matrix(X), y = as.matrix(Y), alpha = 0, intercept = FALSE)`

Measure: Mean-Squared Error

	Lambda	Index	Measure	SE	Nonzero
min	0.0927	100	0.122	0.0161	5
1se	0.3107	87	0.136	0.0150	5

```
> coef(ridge.cv,s="lambda.min")
6 x 1 sparse Matrix of class "dgCMatrix"
      s1
```

```
(Intercept) .
pctbl96      0.02561
unionpct     -0.02598
clint96      -0.09265
GOPmember    0.42533
ADA98        -0.42733
```

```
> coef(ridge.cv,s="lambda.1se")
6 x 1 sparse Matrix of class "dgCMatrix"
      s1
```

```
(Intercept) .
pctbl96      0.008112
unionpct     -0.035391
clint96      -0.117373
GOPmember    0.376793
ADA98        -0.372307
```

# Getting $\lambda$ Via Cross-Validation: Lasso

```
> # Lasso:
>
> lasso.cv<-cv.glmnet(as.matrix(X),as.matrix(Y),alpha=1,intercept=FALSE)
> lasso.cv
```

Call: `cv.glmnet(x = as.matrix(X), y = as.matrix(Y), alpha = 1, intercept = FALSE)`

Measure: Mean-Squared Error

	Lambda	Index	Measure	SE	Nonzero
min	0.0026	64	0.119	0.00906	5
1se	0.0825	27	0.127	0.00812	2

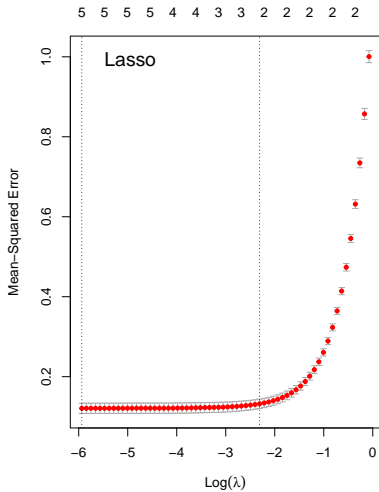
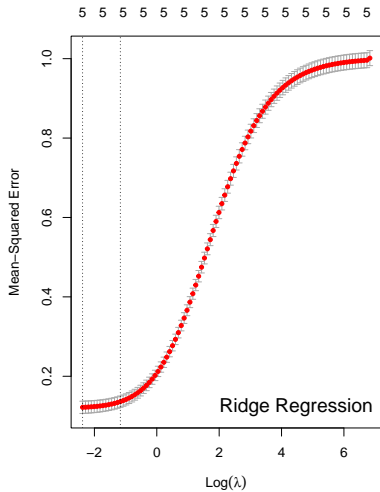
```
> coef(lasso.cv,s="lambda.min")
6 x 1 sparse Matrix of class "dgCMatrix"
      s1
```

```
(Intercept) .
pctbl96      0.02255
unionpct     -0.02141
clint96      -0.05732
GOPmember    0.43174
ADA98        -0.48234
```

```
> coef(lasso.cv,s="lambda.1se")
6 x 1 sparse Matrix of class "dgCMatrix"
      s1
```

```
(Intercept) .
pctbl96      .
unionpct     .
clint96      .
GOPmember    0.3686
ADA98        -0.4992
```

# Cross-Validation Plots



On regularization / shrinkage methods...

- Other useful R packages:
  - `caret` (will do all this, and more)
  - `grplasso`, `elasticnet`, etc.
  - More generally, see the CRAN Task View on machine learning
- Ridge regression / lasso / etc. are widely used for model selection in machine learning / prediction contexts, because
- ...they are automated ways of reducing model complexity and/or overfitting

On multicollinearity in general:

- Can often be ignored without issue
- Consider combining predictors when you can, or
- ...analyzing subsets of the data ( $\rightarrow$  interactions)