4.1: Penalized Regression

Dr. Bean - Stat 5100

1 Why Penalized Regression?

Recall linear regression model and predictive equation:

$$\begin{aligned} Y &= \beta_0 + \beta_1 X_1 + \ldots + \beta_{p-1} X_{p-1} + \varepsilon \\ \hat{Y} &= b_0 + b_1 X_1 + \ldots + b_{p-1} X_{p-1} \end{aligned}$$

IF the assumptions regarding residuals are satisfied, then ordinary least squares (OLS) provides the best (i.e. minimum variance) unbiased estimator for each β_k $(k = 1, \ldots, p-1)$ using the **loss function**

$$\sum_{i=1}^{n} \left(Y_i - \hat{Y}_i \right)^2 .$$

However, when multicollinearity is present, the variance of the estimates for the β_k are inflated. What we would like is a way to shrink the variance of our estimated coefficients, perhaps forcing some coefficients all the way to zero (i.e. variable selection). This will allow us to **stabilize** our coefficient estimates while at the same time provide an alternative approach for variable selection.

However, nothing in statistics comes free. Like the "soul stone" from the avengers series, we must sacrifice something we love in order to obtain smaller variance and a new approach for variable selection.

Our Solution: Sacrifice **unbiased** estimates of the β coefficients in order to reduce their variance.

(Individual) What does it mean to be unbiased?

$$E(b_k) = \beta_k$$

In other words, if I were to use multiple *different* samples to fit my regression line, the estimated coefficients will all be different, but will all be centered around the true (and unknown) coefficients. This is important because it means that as my sample size increases, I expect to get estimates that are closer and closer to the "truth".

(Why might we be OK with giving up unbiasedness in order to minimize variance?

- Coefficients are biased to have smaller magnitude compared to the "truth" so we can still interpret the sign of each estimator.
- Biased, yet stable, estimates of the coefficients can often provide greater predictive accuracy than an OLS model.

2 Penalized Regression Approaches

Alternative Loss Functions:

- Ridge regression

$$\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \quad + \quad \lambda \sum_{k=0}^{p-1} (\beta_k)^2$$

- LASSO

$$\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 + \lambda \sum_{k=1}^{p-1} |\beta_k|$$

- Adaptive LASSO

$$\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 + \lambda \sum_{k=1}^{p-1} \frac{|\beta_k|}{\tilde{b}_k}$$

  - Where $\tilde{b}_k$ represents some initial estimate of the model coefficients (perhaps using OLS or traditional LASSO).

- Elastic Net

$$\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 + \lambda_1 \sum_{k=0}^{p-1} (\beta_k)^2 + \lambda_2 \sum_{k=1}^{p-1} |\beta_k|$$

- Select values of λ that balances added bias with reduced variance.
- Our goal is impose the least amount of biasedness that we can in order to achieve an acceptable reduction in variance.
- One potential solution would be to select λ in such a way that minimizes the cross validation error.

Check out `https://ww2.amstat.org/meetings/csp/2014/onlineprogram/handouts/T3-Handouts.pdf` for additional info on these approaches.

Note that the explanatory variables MUST be standardized in order to use penalized regression techniques. Many functions perform this standardization automatically "under the hood."

2.1 Ridge Regression

Recall Linear Algebra Representation of OLS Regression:

$$Y = X\beta + \epsilon b \qquad = (X'X)^{-1} X'Yb \sim N(\beta, (X'X)^{-1}\sigma^2)$$

Recall also how we can standardize our X and Y variables producing:

$$\begin{aligned} Y^* &= X^*\beta^* + \varepsilon \\ b^* &= (X^{*\prime}X^*)^{-1}X^{*\prime}Y^* \\ &= (r_{XX})^{-1}r_{YX} \\ Cov(b^*) &= (r_{XX})^{-1}\sigma^2 \end{aligned} \qquad \begin{aligned} Y_i^* &= \frac{1}{\sqrt{n-1}} \cdot \frac{Y_i - \bar{Y}}{\text{SD of } Y} \\ X_{k,i}^* &= \frac{1}{\sqrt{n-1}} \cdot \frac{X_{k,i} - \bar{X}_k}{\text{SD of } X_k} \\ r_{XX} &= \text{correlation matrix of } X\text{'s} \\ r_{YX} &= \text{correlation vector between } Y \text{ and } X\text{'s} \end{aligned}$$

Ridge Regression introduces a small positive biasing constant $\lambda > 0$ so that

$$b^R = (r_{XX} + \lambda \cdot I)^{-1} r_{YX}$$

where I is the identity matrix (one's on the diagonal of the matrix and zeros elsewhere).

SAS Code:

```
proc reg data=<dataset> ridge=0 to <upper bound> by <step size>
     outvif outest=<named dataset of relevant ridge output>
     plots(only)=ridge(VIFaxis=log);
  model <model statement> / vif;
run;
```

Two graphical summaries to choose the "right" ridge parameter c:
(Note: these are guides; there is no "optimal" decision)

1. Ridge Trace Plot
   - (Need standardized data for this to be meaningful; SAS does internally)
   - Simultaneous plot of $b_1^R, \ldots, b_{p-1}^R$ (using standardized data) for different ridge parameters c (usually from 0 to 1 or 2)
   - As c increases from 0, the b_k^R may fluctuate wildly and even change signs
   - Eventually the b_k^R will move slowly toward 0
2. VIF Plot
   - Simultaneous plot of the variance inflation factor for the $p-1$ predictors for different ridge parameters

- As c increases from 0, the VIF drop toward 0

In general, choose smallest ridge parameter c:

1. where the b_k^R first become "stable" (their approach towards 0 has slowed)
2. and the VIF's have become "small enough" (close to 1 or less than 1)

2.1.1 Comments on Ridge Regression

- Choice of ridge parameter is somewhat subjective, but must be defendable (i.e. with a trace plot)
- given ridge parameter c, can get resulting parameter estimates b on the "unstandardized" (original data) scale
  - SAS gives these automatically, but need textbook equation 7.46b to get intercept b_0:

$$\beta_0 = \bar{Y} - \beta_1 \bar{X}_1 - \beta_2 \bar{X}_2 - \ldots - \beta_{p-1} \bar{X}_{p-1}$$

- ridge regression estimates b tend to be more robust against small changes to data than are OLS estimates
- predictors with very unstable ridge trace (tends toward zero without any plateau or slowing down) may be dropped from model, providing an alternative to stepwise variable selection techniques
- **major limitation**: traditional inference is not directly applicable to ridge regression estimates (part of our "soul stone" sacrifice)

2.2 LASSO (Least Absolute Shrinkage and Selection Operator)

Find b to minimize

$$\sum_{i=1}^{n} \left(Y_i - \hat{Y}_i \right)^2 + \lambda \sum_{k=1}^{p-1} |\beta_k|$$

Switching from $\lambda \sum_{k=1}^{p-1} \beta_k^2$ in ridge regression to $\lambda \sum_{k=1}^{p-1} |\beta_k|$ in LASSO, may seem minor, but this change causes b_k values to now shrink all the way to zero.

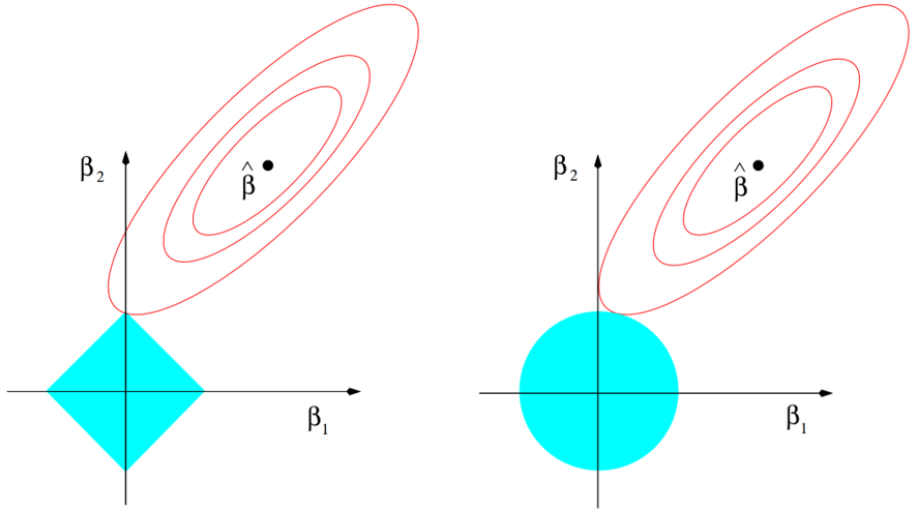


FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions $|\beta_1| + |\beta_2| \leq t$ and $\beta_1^2 + \beta_2^2 \leq t^2$, respectively, while the red ellipses are the contours of the least squares error function.*

Options exist for choosing λ We can use these because we now have models with different numbers of coefficients, not the case in ridge regression.

- likelihood function-based criteria (Adj. R^2, C_p, AIC, SBC, etc.)
- cross-validation
  - withhold some of the data, fit on the rest, then predict on withheld portion
  - select λ to minimize something like (others exist)

$$PRESS = \sum_{i=1}^{n} \left(\frac{Y_i - \hat{Y}_i}{1 - h_{ii}} \right)^2$$

SAS Code

```
proc glmselect data=<dataset> plots=(criterion <measure>);
 class <all qualitative variables>;
 model <your model>
        / selection=lasso(adaptive choose=<selection method> stop=none);
 output out=<output dataset> p=<lasso predictions>;
run;
```

One way to visualize progress of model is to show ASE as each variable is added

$$\text{ASE} = \frac{\text{SSE}}{n} \qquad MSE = \frac{\text{SSE}}{n-p}$$

.

2.3 Adaptive LASSO

- Problem: LASSO is known to give more biased estimates of nonzero coefficients
- Solution: Allow higher penalty for zero coefficients and lower penalty for nonzero coefficients

Find b to minimize

$$\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 + \lambda \sum_{k=1}^{p-1} \frac{|\beta_k|}{b_k}$$

"Adaptive" weights: $\frac{1}{b_k}$, where b_k is obtained from an initial model fit (using OLS or regular LASSO or something else)
– control shrinking of zero coefficients more than nonzero coefficients

2.4 Elastic Net

> *Similar to the lasso, the elastic net simultaneously does automatic variable selection and continuous shrinkage, and it can select groups of correlated variables. It is like a stretchable fishing net that retains 'all the big fish'"* - Zou and Hastie (2005)

Some limitations of LASSO:

- When number of predictors $(p-1)$ exceeds sample size (n), LASSO will select up to n predictor variables before it saturates.
- In the presence of high multicollinearity, LASSO tends to select only one variable from the group of correlated predictors.
- When sample size (n) exceeds number of predictors $(p-1)$ <u>and</u> there is high multicollinearity, LASSO is out-performed (prediction-wise) by ridge regression.

Elastic Net overcomes these limitations:

- can select more than n variables
- can select more than one variable from a group of highly collinear predictors
- can achieve better predictive performance

Find b to minimize

$$\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 + \lambda_1 \sum_{k=0}^{p-1} (\beta_k)^2 + \lambda_2 \sum_{k=1}^{p-1} |\beta_k|$$

4.1.1: SAS - Penalized Regression Methods
(Ridge Regression, LASSO, and Elastic Net)

Example 1: (Ridge Regression; recall Handout 2.6.1 example) A study seeks to relate (in females) amount of body fat (Y) to triceps skinfold thickness (X_1), thigh circumference (X_2), and midarm circumference (X_3). Amount of body fat is expensive to measure, requiring immersion of person in water. This expense motivates the desire for a predictive model based on these inexpensive predictors.

```
/* Input data */
data bodyfat;
   input triceps thigh midarm body @@; cards;
  19.5  43.1  29.1  11.9      24.7  49.8  28.2  22.8
  30.7  51.9  37.0  18.7      29.8  54.3  31.1  20.1
  19.1  42.2  30.9  12.9      25.6  53.9  23.7  21.7
  31.4  58.5  27.6  27.1      27.9  52.1  30.6  25.4
  22.1  49.9  23.2  21.3      25.5  53.5  24.8  19.3
  31.1  56.6  30.0  25.4      30.4  56.7  28.3  27.2
  18.7  46.5  23.0  11.7      19.7  44.2  28.6  17.8
  14.6  42.7  21.3  12.8      29.5  54.4  30.1  23.9
  27.7  55.3  25.7  22.6      30.2  58.6  24.6  25.4
  22.7  48.2  27.1  14.8      25.2  51.0  27.5  21.1
;
run;

/* Look at original fit */
proc reg data=bodyfat;
  model body = triceps thigh midarm / vif;
  title1 'Bodyfat Regression (original fit)';
run;
```

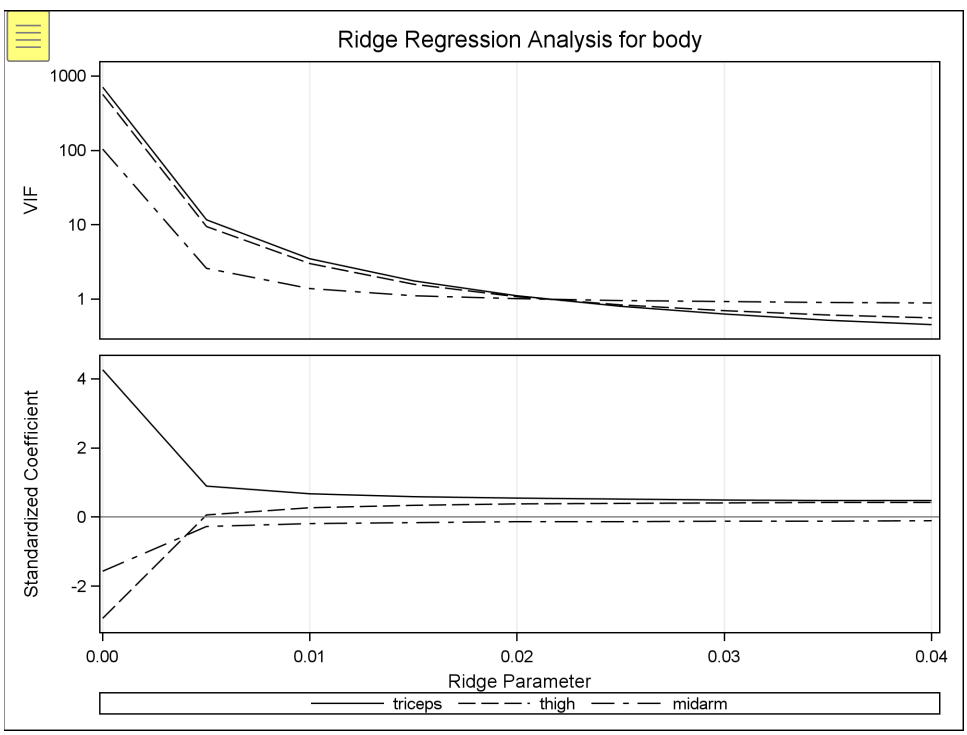
Bodyfat Regression (original fit)

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Variance Inflation
Intercept	**1**	117.08469	99.78240	1.17	0.2578	0
triceps	**1**	4.33409	3.01551	1.44	0.1699	708.84291
thigh	**1**	-2.85685	2.58202	-1.11	0.2849	564.34339
midarm	**1**	-2.18606	1.59550	-1.37	0.1896	104.60601

```
/* Try ridge regression as a remedial measure */
proc reg data=bodyfat ridge=0 to .04 by .005
     outvif outest=ridgests
     plots(only)=ridge(VIFaxis=log);
  model body = triceps thigh midarm / vif;
  title1 'Bodyfat Ridge Regression';
  title2 '(find adequate ridge parameter)';
run;
/* What these options do:

     ridge=0 to .04 by 0.005
        run a regression with each of these ridge parameter
       values

     outvif outest=ridgests
        ask for relevant output to be sent to a data set
       called ridgests (will include VIF and standardized
       coefficients for each ridge parameter)

     plots(only)=ridge(VIFaxis=log);
       make Ridge Trace and VIF plots only, with vertical axis
       in VIF plot on log scale
 */
```

Bodyfat Ridge Regression
(find adequate ridge parameter)



```
/* Now look at variable coeffs with ridge parameter 0.02 */
proc reg data=bodyfat outest=ridgenew outseb ridge=0.02
       outvif noprint;
    model body = triceps thigh midarm;
    title1 'Bodyfat Ridge Regression (c=.02)';
run;
proc print data=ridgenew;
  var _type_  _rmse_  triceps thigh midarm;
  title1 'Ridge Estimates for Variable Coefficients,';
  title2 'with ridge parameter c = 0.02';
run;
/* PARMS and SEB give the result of the regular OLS regression.
   RIDGE and RIDGESEB give the result of the ridge regression.
   -- Note no intercept is given; need to use textbook
      equation 7.46b to get intercept in ridge reg. (as below)
   Note substantial drop in SE for estimates in ridge reg.
   RIDGEVIF give the VIF after ridge regression.
 */
```

Ridge Estimates for Variable Coefficients,
with ridge parameter c = 0.02

Obs	_TYPE_	_RMSE_	triceps	thigh	midarm
1	PARMS	2.47998	4.33409	-2.85685	-2.18606
2	SEB	2.47998	3.01551	2.58202	1.59550
3	RIDGEVIF	.	1.10255	1.08054	1.01051
4	RIDGE	2.59924	0.55535	0.36814	-0.19163
5	RIDGESEB	2.59924	0.12465	0.11841	0.16436

```
/* Get intercept term in ridge regression */
proc means data=bodyfat mean;
  var body triceps thigh midarm;
  title1 'Summary Statistics';
run;
data temp;
 b0 = 20.195 - 0.55535*25.305 - 0.36814*51.17 + 0.19163*27.62;
proc print data=temp;
 var b0;
 title1 'Ridge Regression Intercept';
run;
```

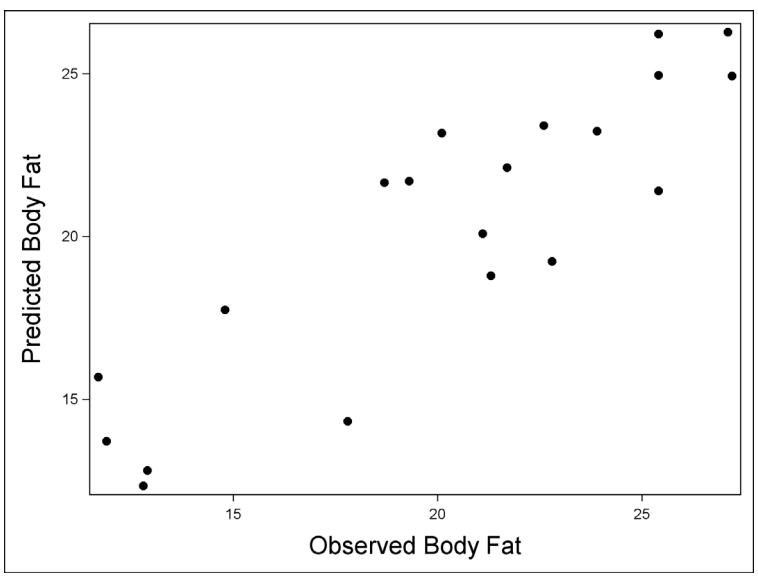
Summary Statistics

Variable	Mean
body	20.1950000
triceps	25.3050000
thigh	51.1700000
midarm	27.6200000

Ridge Regression Intercept

Obs	b0
1	-7.40303

```
/* Get predicted values in ridge regression */
data bodyfat; set bodyfat;
  predbody = -7.40303 + 0.55535*triceps
             + 0.36814*thigh - 0.19163*midarm;
proc sgplot data=bodyfat;
  scatter x=body y=predbody / markerattrs=(symbol=CIRCLEFILLED);
  xaxis label='Observed Body Fat' labelattrs=(size=15pt);
  yaxis label='Predicted Body Fat' labelattrs=(size=15pt);
  title1;
run;
```



Example 2: (Baseball) This data set (from the SAS Help) contains salary (for 1987) and performance (1986 and some career) data for 322 MLB players who played at least one game in both 1986 and 1987 seasons, excluding pitchers. How can salary be predicted from performance?

```
data baseball; set sashelp.baseball;
proc contents varnum data=baseball;
   ods select position;
run;
```

Variables in Creation Order

#	Variable	Type	Len	Label
1	Name	Char	18	Player's Name
2	Team	Char	14	Team at the End of 1986
3	nAtBat	Num	8	Times at Bat in 1986
4	nHits	Num	8	Hits in 1986
5	nHome	Num	8	Home Runs in 1986
6	nRuns	Num	8	Runs in 1986
7	nRBI	Num	8	RBIs in 1986
8	nBB	Num	8	Walks in 1986
9	YrMajor	Num	8	Years in the Major Leagues
10	CrAtBat	Num	8	Career Times at Bat
11	CrHits	Num	8	Career Hits
12	CrHome	Num	8	Career Home Runs
13	CrRuns	Num	8	Career Runs
14	CrRbi	Num	8	Career RBIs
15	CrBB	Num	8	Career Walks
16	League	Char	8	League at the End of 1986
17	Division	Char	8	Division at the End of 1986
18	Position	Char	8	Position(s) in 1986
19	nOuts	Num	8	Put Outs in 1986
20	nAssts	Num	8	Assists in 1986
21	nError	Num	8	Errors in 1986
22	Salary	Num	8	1987 Salary in $ Thousands
23	Div	Char	16	League and Division
24	logSalary	Num	8	Log Salary

```
/* lasso */
proc glmselect data=baseball plots=(criterion ase);
 class league division;
 model logSalary = nAtBat nHits nHome nRuns nRBI nBB
                   yrMajor crAtBat crHits crHome crRuns crRbi
                   crBB league division nOuts nAssts nError
        / selection=lasso(adaptive choose=sbc stop=none);
 output out=out1 p=predlasso;
run;
```

Data Set	WORK.BASEBALL
Dependent Variable	logSalary
Selection Method	Adaptive LASSO
Stop Criterion	None
Choose Criterion	SBC
Effect Hierarchy Enforced	None

Number of Observations Read	322
Number of Observations Used	263

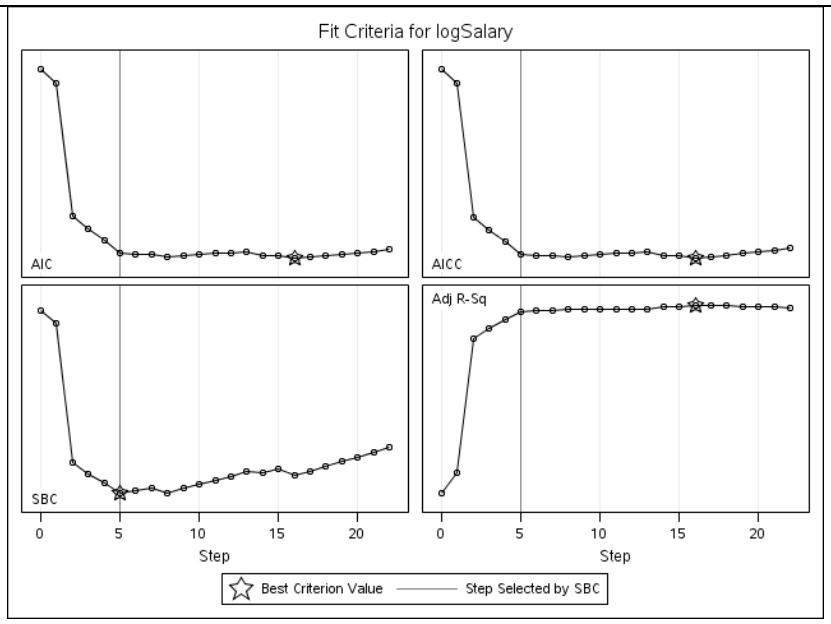
Class Level Information

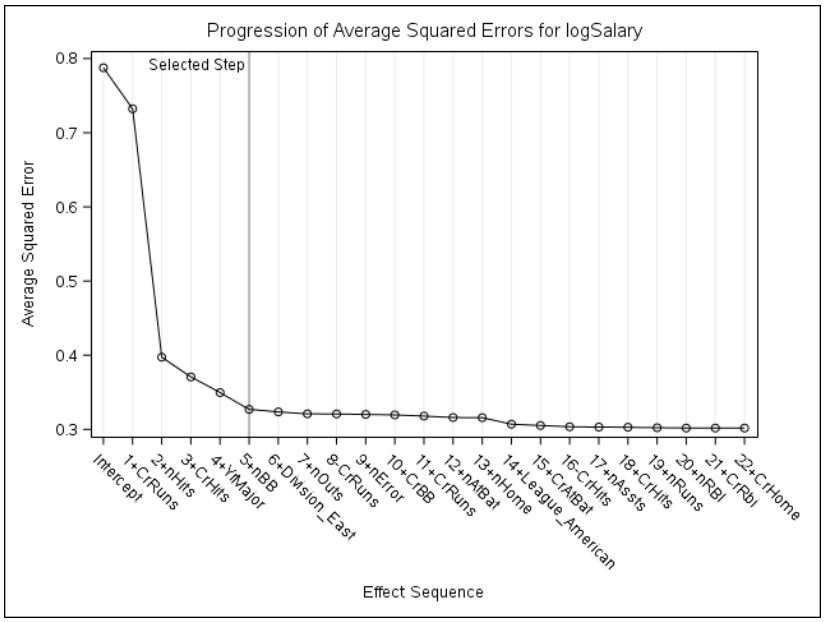
Class	*Levels*	*Values*
League	2	American National
Division	2	East West

LASSO Selection Summary

Step	*Effect Entered*	*Effect Removed*	*Number Effects In*	*SBC*
		** Optimal Value of Criterion*		
0	*Intercept*		1	-57.2041
1	*CrRuns*		2	-70.8348
2	*nHits*		3	-226.0696
3	*CrHits*		4	-238.6648
4	*YrMajor*		5	-248.4971
5	*nBB*		6	-260.5682*
6	*Division_East*		7	-257.7020
7	*nOuts*		8	-254.3352
8		*CrRuns*	7	-260.1040
9	*nError*		8	-254.9990
10	*CrBB*		9	-249.9243
11	*CrRuns*		10	-245.7008
12	*nAtBat*		11	-241.6564
13	*nHome*		12	-236.3245
14	*League_American*		13	-238.1068
15	*CrAtBat*		14	-234.0015
16		*CrHits*	13	-241.0870
17	*nAssts*		14	-235.9894
18	*CrHits*		15	-230.5456
19	*nRuns*		16	-225.5197
20	*nRBI*		17	-220.3634
21	*CrRbi*		18	-214.7952
22	*CrHome*		19	-209.2505

Selection stopped because all candidate effects for entry are linearly dependent on effects in the model.





Selected Model

The selected model, based on SBC, is the model at Step 5.

Root MSE	0.57845
Dependent Mean	5.92722
R-Square	0.5849
Adj R-Sq	0.5768
AIC	-17.00115
AICC	-16.56194
SBC	-260.56823

Parameter Estimates

Parameter	*DF*	*Estimate*
Intercept	1	4.229778
nHits	1	0.007194
nBB	1	0.005629
YrMajor	1	0.062808
CrHits	1	0.000222
CrRuns	1	0.000136

```
/* elastic net */
proc glmselect data=out1 plots=(criterion ase) seed=12;
 class league division;
 model logSalary = nAtBat nHits nHome nRuns nRBI nBB
                   yrMajor crAtBat crHits crHome crRuns crRbi
                   crBB league division nOuts nAssts nError
      / selection=elasticnet(stop=none choose=cv)
        cvmethod=random(20);
 output out=out2 p=predelasticnet;
run;
```

Data Set	WORK.OUT1
Dependent Variable	logSalary
Selection Method	ELASTICNET
Stop Criterion	None
Choose Criterion	Cross Validation
Cross Validation Method	Random
Cross Validation Fold	20
Effect Hierarchy Enforced	None
Random Number Seed	12

Number of Observations Read	322
Number of Observations Used	263

Class Level Information

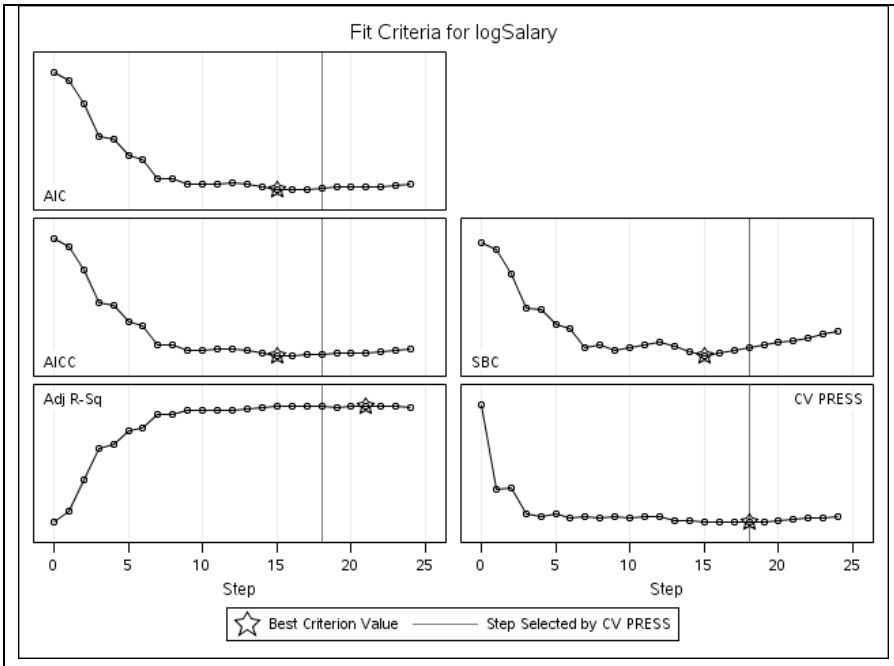
Class	*Levels*	*Values*
League	2	American National
Division	2	East West

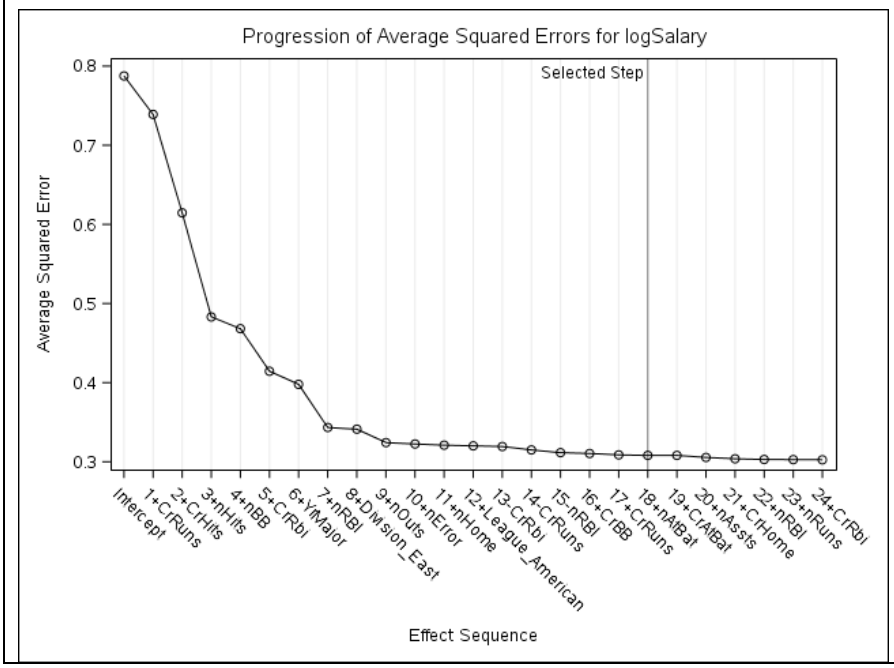
Elastic Net Selection Summary

Step	*Effect Entered*	*Effect Removed*	*Number Effects In*	*CV PRESS*
		** Optimal Value of Criterion*		
0	*Intercept*		1	209.2326
1	*CrRuns*		2	123.1776
2	*CrHits*		3	123.7433
3	*nHits*		4	97.6956
4	*nBB*		5	94.7216
5	*CrRbi*		6	98.1015
6	*YrMajor*		7	92.7082
7	*nRBI*		8	94.5500
8	*Division_East*		9	93.3921
9	*nOuts*		10	94.1530
10	*nError*		11	93.8913
11	*nHome*		12	94.2533
12	*League_American*		13	94.4968
13		*CrRbi*	12	90.7314
14		*CrRuns*	11	90.1957
15		*nRBI*	10	89.6571
16	*CrBB*		11	89.2733
17	*CrRuns*		12	89.4515
18	*nAtBat*		13	88.9017*
19	*CrAtBat*		14	89.2818
20	*nAssts*		15	89.7926
21	*CrHome*		16	91.8598
22	*nRBI*		17	92.6309
23	*nRuns*		18	93.1973
24	*CrRbi*		19	94.5881

** Optimal Value of Criterion*

Selection stopped because all candidate effects for entry are linearly dependent on effects in the model.





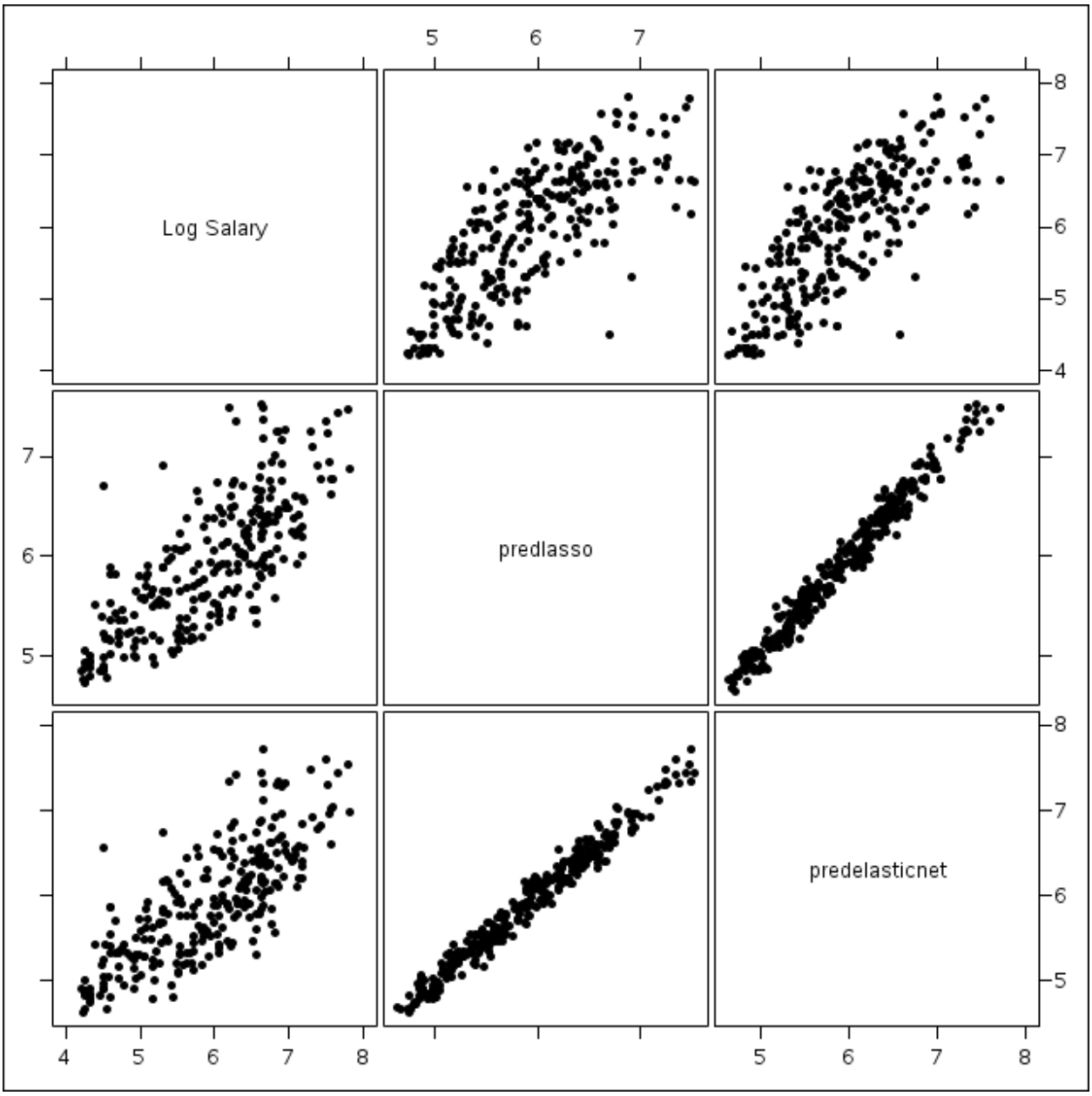
Selected Model
The selected model, based on Cross Validation, is the model at Step 18.

Root MSE	0.56923
Dependent Mean	5.92722
R-Square	0.6090
Adj R-Sq	0.5902
AIC	-18.72037
AICC	-17.02682
SBC	-237.28237
CV PRESS	88.90168

Parameter Estimates

Parameter	*DF*	*Estimate*
Intercept	1	4.195962
nAtBat	1	-0.000112
nHits	1	0.006807
nHome	1	0.003545
nBB	1	0.007082
YrMajor	1	0.070194
CrHits	1	0.000247
CrRuns	1	0.000212
CrBB	1	-0.000348
League_American	1	-0.092575
Division_East	1	0.144062
nOuts	1	0.000192
nError	1	-0.007767

```
proc sgscatter data=out2;
  matrix logSalary predlasso predelasticnet /
          markerattrs=(symbol=circlefilled size=6pt);
run;
```



4.2: Variations on OLS (Ordinary Least Squares)

Dr. Bean - Stat 5100

1 Why alternatives?

Remember this: when standard model assumptions are met, OLS is the "best" linear modeling approach.
No matter how good we are at performing variable transformations, there are some situations where we simply cannot satisfy linear model assumptions of constant variance, normality, or independence.

Fortunately, there are several OLS alternatives that address one or more of these issues.

The cost:

- Lose our ability to conduct inference on the coefficients.
- The models become harder to fit/harder to explain.

2 Weighted Least Squares (textbook §11.1)

Recall regression model $Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_{p-1} X_{p-1} + \varepsilon$ in matrix form:
(Ch. 5, Handout #12 p. 2)

$$Y = X\beta + \varepsilon$$

Model assumption: $\varepsilon \sim N(0, \sigma^2 I)$

- If constant variance, (i.e., $Cov(\varepsilon) = \sigma^2 I$), then use OLS:

$$b = (X'X)^{-1}X'Y$$

- If non-constant variance, then can estimate and account for it (WLS):

$$V = Cov(\varepsilon) \qquad \text{(typically assumed diagonal)}$$

$$W = V^{-1} \qquad \text{(i.e. the weights)}$$

$$b_w = (X'WX)^{-1}X'WY$$

Why give *smaller* weight to observations with *larger* variance when calculating the model coefficients?

Smaller variance is equivalent to greater certainty. Certain information should have greater "value" than uncertain information.

Typically, $Cov(\varepsilon)$ must be estimated

- can often relate variance of residuals (or squared residuals) to predictors or $\hat{Y}$ values
- example (as in Ex. 1 of Handout 4.2.1): residual vs. X_1 is megaphone-shaped (linear relationship between SD of residual and X_1)
  - regress absolute residuals on X_1 and get predicted values s (as function of X_1)
  - define weights $w = 1/s^2$
- see p. 425 for other examples
  – key is how to estimate w for given scenario, as a function of X's

Some things to remember:

- The *pattern* of the residuals against the other variables determines how we should estimate the weights.
- Its OK to see non-constant variance in weighted model.
- In **Spatial Statistics, weights are calculated using geographic similarity.**

3 Robust Regression (textbook §11.3)

Rather than remove influential observations and outliers, we may choose to reduce their influence by changing the way we measure "error".

3.1 IRLS (iteratively reweighted least squares)

1. Obtain (maybe from OLS) b, then calculate $\hat{Y} = Xb$ and $e = Y - \hat{Y}$
2. Calculate weights W, based on e (lots of weight functions available)
3. Calculate (WLS) $b_w = (X'WX)^{-1}X'WY$ and resulting $e = Y - Xb_w$
4. Iterate steps 2 & 3 to convergence of b_w

How to calculate weights?
- usually chosen to optimize some criterion
- the choice of criterion determines the method of weight calculation

3.2 M-estimation

- If $u_1, \ldots, u_n$ are *iid* from some distribution with parameter θ, then the type-M estimate of θ is of the form

$$\hat{\theta} = \arg\min \sum \rho(u_i; \theta)$$

  where ρ is some "scalar objective function"

- Example: $\rho(u;\theta) = -\frac{1}{n}\log f(u;\theta)$, f is pdf of distribution of $u_1, \ldots, u_n$. Then

$$\begin{aligned}\hat{\theta} &= \arg\max \sum \frac{1}{n}\log f(u_i;\theta) \\ &= \arg\max(\text{likelihood}) \\ &= (\text{what is this called?})\end{aligned}$$

- W-estimation approach in IRLS:
  1. Calculate robust estimate of σ, such as $s = \frac{MAD(e)}{0.6745}$
  2. Let $u_i = \frac{e_i}{s}$ be "scaled" (or standardized) residual
  3. Calculate (diagonal) weights $w_i = \frac{\psi(u_i)}{u_i}$
     – where $\psi(u) = \rho'(u)$ for some scalar objective function ρ

Example – Tukey Bisquare (sometimes called Tukey's Biweight):

$$\rho(u) = \begin{cases} \frac{c^2}{3}\left(1 - [1 - \left(\frac{u}{c}\right)^2]^3\right) & \text{if } |u| \leq c\text{; default } c = 4.685 \\ \frac{c^2}{3} & \text{otherwise} \end{cases}$$

Bisquare weight function: $w(u) = \left(1 - \left(\frac{u}{c}\right)^2\right)^2$ for $|u| \leq c$, 0 otherwise

Note: M-estimation works well for outliers; for leverage points, use MM-estimation (see SAS help)

3. Nonlinear Regression (textbook §13.1 – 13.2)

What if Y vs $X_1, \ldots X_{p-1}$ not linear (in β's)?
– Usually need mechanistic theory

Mechanistic Theory: the assumption that a natural phenomenon can be understood through the use of an equation.
Example: Population Growth

$$\frac{dN}{dt} = rN(1 - N/K)$$

`proc nlin` fits these nonlinear models

- Parameters estimated by an iterative process to reduce the SSE at each iteration, until convergence
- Keys to [useful] convergence:
  - form of nonlinear equation
  - initial parameter estimates

If you were dropped randomly on the side of a mountain with dense fog, how would you find your way down? How would you know when you have made it to the bottom (assuming the fog persists at the bottom)?

You would most likely take each step in a direction that would cause you to be lower than you were before. You would know that you (hopefully) arrived at the bottom of the mountain when you can no longer find a direction to take a step in which you could decrease your altitude. This approach is often called **gradient descent**.

Example 3.1: $Y = \beta_0 + \beta_1 X_1^{\beta_2} - \beta_3 \exp(\beta_4 X_2) \quad (+\epsilon)$
(with simulated data)

Example 3.2: a nonlinear curve to describe sand compression, from Lagioia et al. (1996) Computers and Geotechnics 19(3):171-191

$$f = \frac{p}{p_c} - \frac{\left(1 + \frac{q}{p \cdot M \cdot k_2}\right)^{\frac{k_2}{(1-\mu)(k_1-k_2)}}}{\left(1 + \frac{q}{p \cdot M \cdot k_1}\right)^{\frac{k_1}{(1-\mu)(k_1-k_2)}}},$$

where

$$\begin{aligned}
f &= \text{yield surface (response)} \\
q &= \text{deviatoric stress (predictor)} \\
p &= \text{mean effective stress (predictor)} \\
p_c &= \text{hardening / softening constant defining current size of surface (known)} \\
\eta &= \text{stress ratio } p/q \\
M &= \text{parameter defining value of } \eta \text{ with no strain increment} \\
\mu &= \text{parameter defining general slope of } d \text{ vs. } \eta \text{ curve} \\
\alpha &= \text{parameter defining how close to } \eta = 0 \text{ axis curve bends towards } d = \infty \\
d &= \text{dilatancy, } 2\mu M(1-\alpha)
\end{aligned}$$

Goal: find μ, α, and M to make $f \approx 0$, and look at the relationship between these three parameters

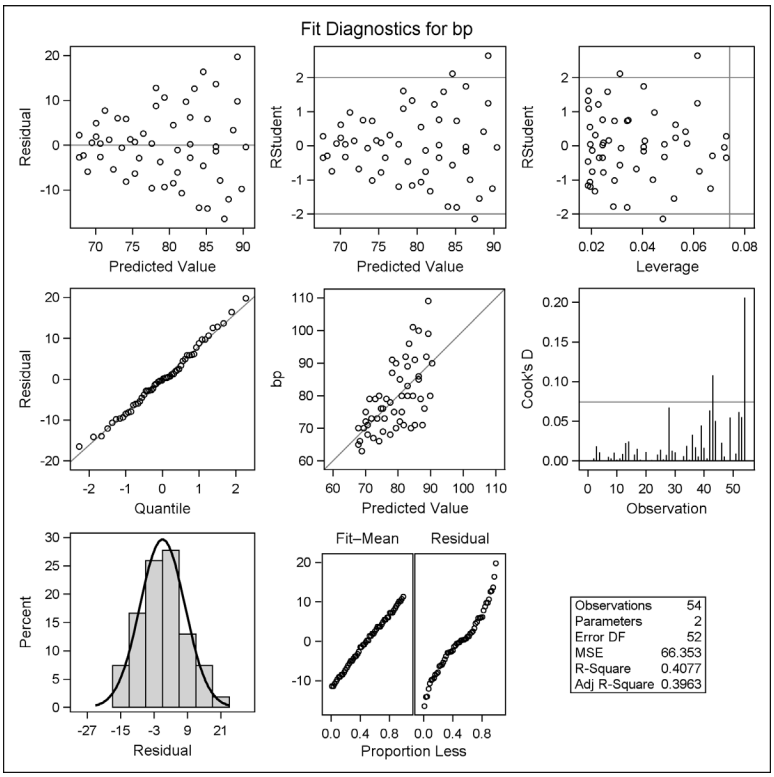
`proc model` estimates such nonlinear systems (can do multiple equations)

From playing with this in SAS, it appears that to achieve convergence of estimates in `proc model`, the most important thing is that at least one of the tails of the $q * p$ curve to be fit has data along most of it. To make the convergent estimates "good", it appears necessary to have data along both tails. It is also crucial that the initial starting estimates be good, especially for M (maybe within .2 or so).

Stat 5100 Handout #21 – SAS: Variations on Ordinary Least Squares (Weighted Least Squares, Robust Regression, Nonlinear Regression)

Example 1: (Weighted Least Squares) A health researcher is interested in studying the relationship between diastolic blood pressure (bp) and age in adult women. Data are reported on 54 healthy adult women.

```
/* Read in the data (Table 11.1) */
data bpexample; input age bp @@; cards;
  27   73   21   66   22   63   24   75   25   71   23   70
  20   65   20   70   29   79   24   72   25   68   28   67
  26   79   38   91   32   76   33   69   31   66   34   73
  37   78   38   87   33   76   35   79   30   73   31   80
  37   68   39   75   46   89   49  101   40   70   42   72
  43   80   46   83   43   75   44   71   46   80   47   96
  45   92   49   80   48   70   40   90   42   85   55   76
  54   71   57   99   52   86   53   79   56   92   52   85
  50   71   59   90   50   91   52  100   58   80   57  109
;

/* Try OLS */
proc reg data=bpexample;
  model bp = age;
  title1 'OLS model fit';
output out=out1 p=pred r=resid;
run;
```



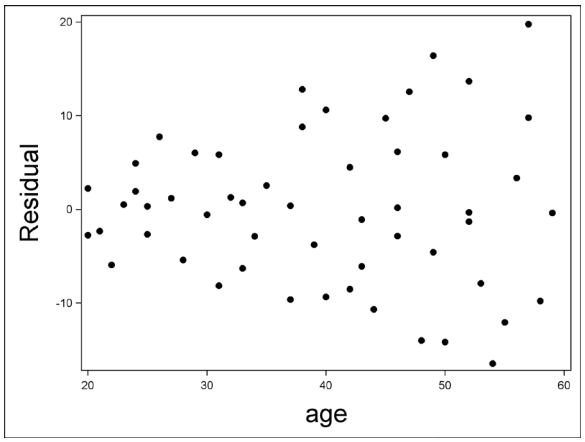

```
/* Use resid_num_diag macro from
   http://www.stat.usu.edu/jrstevens/stat5100/resid_num_diag_1line.sas
 */
 %macro resid_num_diag(dataset,datavar, ...

%resid_num_diag(dataset=out1, datavar=resid,
   label='residual', predvar=pred, predlabel='predicted');
run;
```

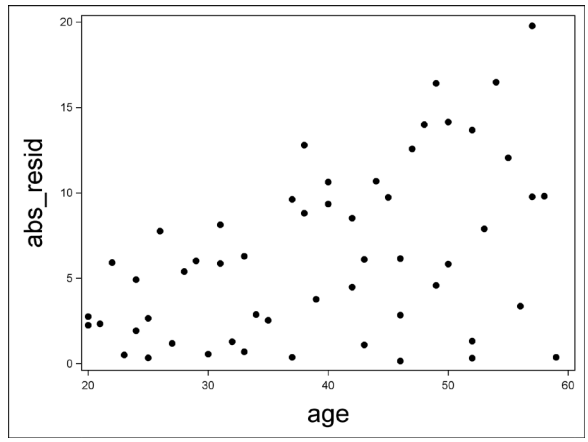
P-value for Brown-Forsythe test of constant variance in residual vs. predicted

Obs	t_BF	BF_pvalue
1	2.78547	.007440565

```
/* Look for relationship between SD of resid and X */
data out1; set out1;
  abs_resid = abs(resid);
proc sgplot data=out1;
  scatter x=age y=resid / markerattrs=(symbol=CIRCLEFILLED);
  xaxis labelattrs=(size=20pt);
```

```
  yaxis labelattrs=(size=20pt);
run;
proc sgplot data=out1;
  scatter x=age y=abs_resid / markerattrs=(symbol=CIRCLEFILLED);
  xaxis labelattrs=(size=20pt);
  yaxis labelattrs=(size=20pt);
run;
```



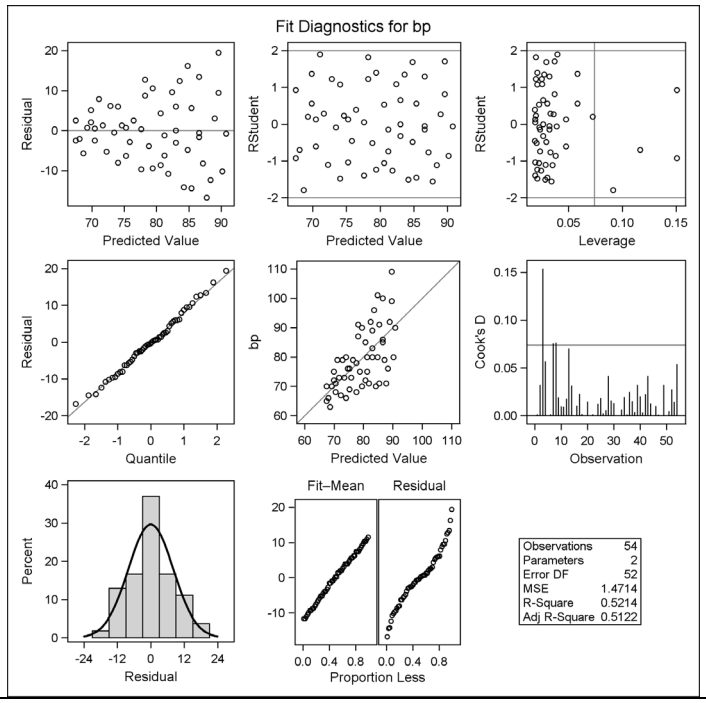



```
/* Get estimate of SD of resid based on X */
proc reg data=out1 noprint;
  model abs_resid = age;
  output out=out2 p=estSD;
run;

/* Define weight */
data out2; set out2;
  useWeight = 1/estSD**2;
run;

/* Fit WLS model */
proc reg data=out2;
  model bp = age;
  weight useWeight;
  title1 'WLS model fit';
run;
```

WLS model fit

Parameter Estimates					
Variable	**DF**	**Parameter Estimate**	**Standard Error**	**t Value**	**Pr > |t|**
Intercept	**1**	55.56577	2.52092	22.04	<.0001

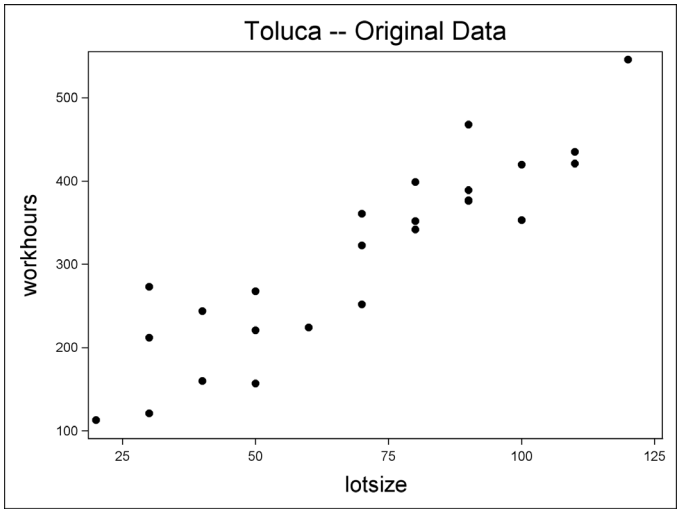
age	**1**	0.59634	0.07924	7.53	<.0001

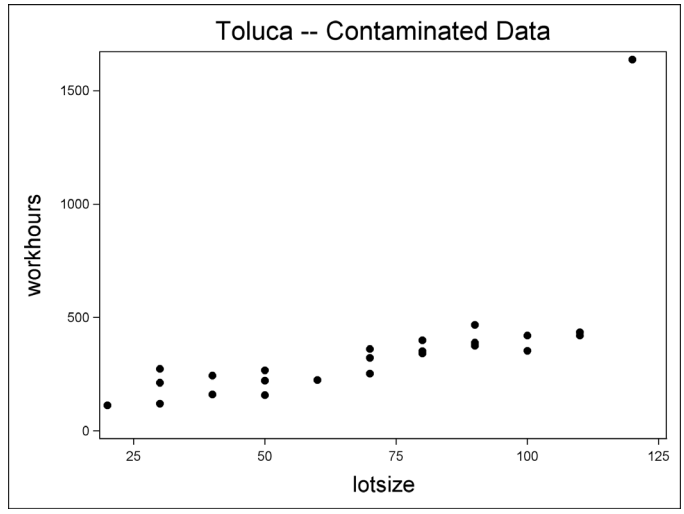


Example 2: (IRLS; recall Handout #2 example) As part of a cost improvement program, the Toluca company wished to better understand the relationship between the lot size (X) and the total work hours (Y).

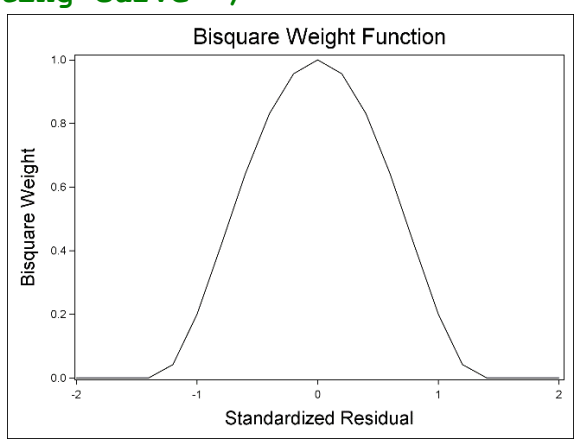
```
/* Input data -- recall Ch. 1 example */
data toluca; input lotsize workhours @@; cards;
   80  399   30  121   50  221   90  376   70  361   60  224
  120  546   80  352  100  353   50  157   40  160   70  252
   90  389   20  113  110  435  100  420   30  212   50  268
   90  377  110  421   30  273   90  468   40  244   80  342
   70  323
  ;
run;

/* Look at original data */
proc sgplot data=toluca;
  scatter x=lotsize y=workhours / markerattrs=(symbol=CIRCLEFILLED);
  xaxis labelattrs=(size=15pt);
  yaxis labelattrs=(size=15pt);
  title1 height=2 'Toluca -- Original Data';
run;

/* To show effect of robust regression, look at
   'contaminated' data */
data contam; set toluca;
  if workhours > 500 then workhours = workhours*3;
```

```
proc sgplot data=contam;
  scatter x=lotsize y=workhours / markerattrs=(symbol=CIRCLEFILLED);
  xaxis labelattrs=(size=15pt);
  yaxis labelattrs=(size=15pt);
  title1 height=2 'Toluca -- Contaminated Data';
run;
```






```
/* Look at shape of bisquare weighting curve */
data temp; input u @@; cards;
 -2.0 -1.8 -1.6 -1.4 -1.2
 -1.0 -0.8 -0.6 -0.4 -0.2
    0  0.2  0.4  0.6  0.8
  1.0 1.2 1.4 1.6 1.8 2.0
 ;
data temp; set temp;
 c = 1.345;
 w = (1-(u/c)**2)**2;
 if abs(u) >= c then w = 0;
proc sgplot data=temp;
 series x=u y=w;
 yaxis label='Bisquare Weight' labelattrs=(size=15pt);
 xaxis label='Standardized Residual' labelattrs=(size=15pt);
 title1 height=2 'Bisquare Weight Function';
run;
```




```
/* OLS regression on original data */
proc reg data=toluca;
 model workhours = lotsize;
 output out=out2 p=pred2;
 title1 'Regression on original data';
run;
```

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	62.36586	26.17743	2.38	0.0259
lotsize	1	3.57020	0.34697	10.29	<.0001

```
/* OLS regression on response-contaminated data */
proc reg data=contam;
 model workhours = lotsize;
 output out=out3 p=pred3;
 title1 'Regression on response-contaminated data';
run;
```

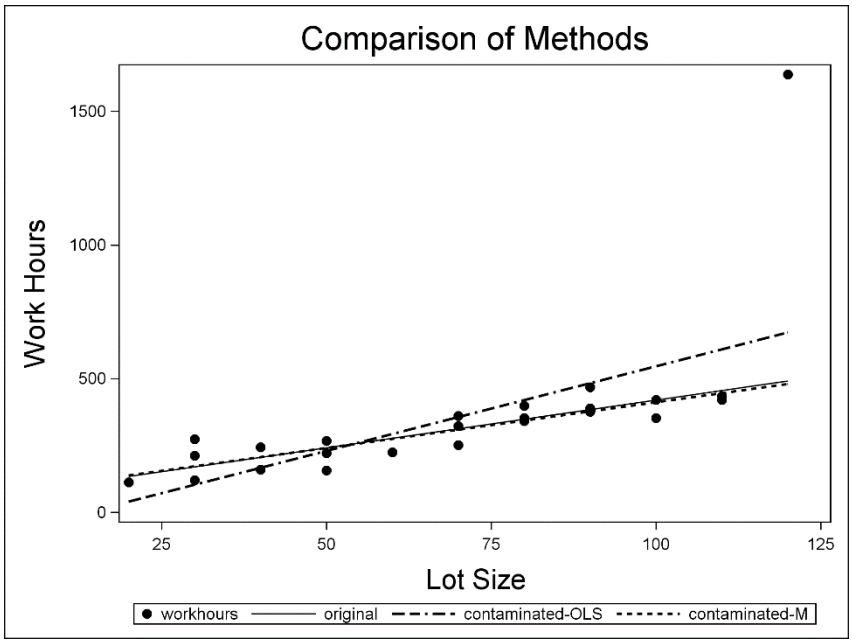
Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	-86.98444	120.90818	-0.72	0.4791
lotsize	1	6.32778	1.60259	3.95	0.0006

```
/* Robust (M) regression on response-contaminated data */
proc robustreg data=contam method=M (wf=bisquare);
 model workhours = lotsize;
 output out=out4 p=pred4;
 title1 'Robust (M) regression on response-contaminated data';
run;
```

Parameter Estimates							
Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	69.2426	27.3941	15.5511	122.9340	6.39	0.0115
lotsize	1	3.4207	0.3631	2.7091	4.1324	88.75	<.0001
Scale	1	56.2335					

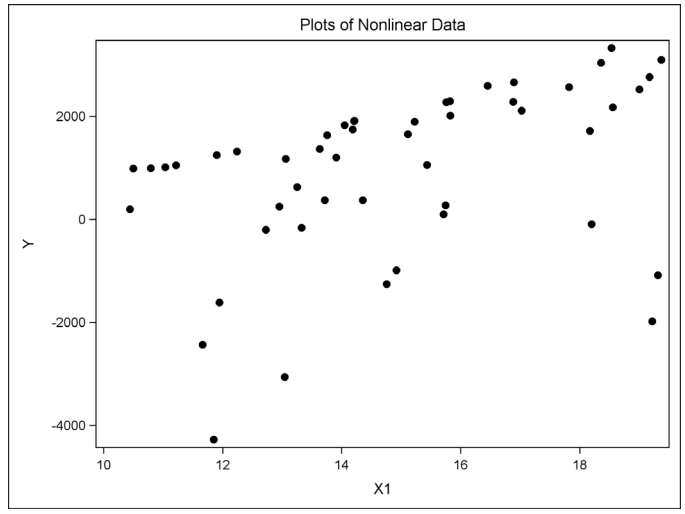
```
/* Visualize comparison of methods */
data out2; set out2; keep pred2;
data out3; set out3; keep pred3;
data out4; set out4; keep pred4;
data comp; merge contam out2 out3 out4;
  label pred2 = 'original'
```

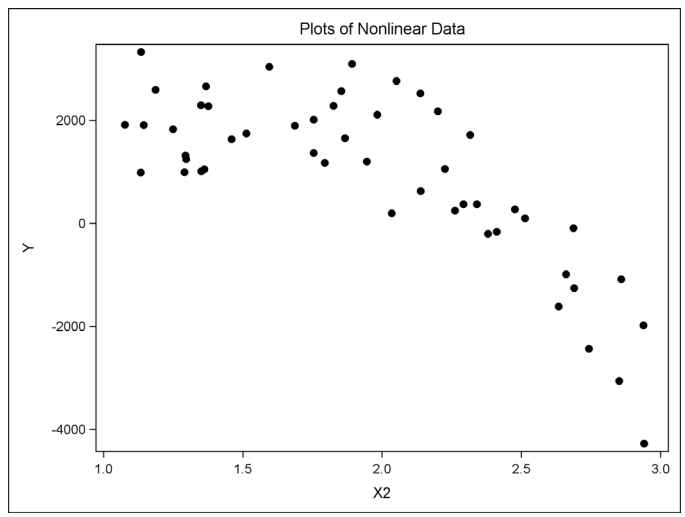
```
        pred3 = 'contaminated-OLS'
        pred4 = 'contaminated-M';
proc sort data=comp;  by lotsize;
proc sgplot data=comp;
 scatter x=lotsize y=workhours /
   markerattrs=(symbol=CIRCLEFILLED);
 series x=lotsize y=pred2 /
   lineattrs=(pattern=1
             thickness=1);
 series x=lotsize y=pred3 /
   lineattrs=(pattern=14
             thickness=2);
 series x=lotsize y=pred4 /
   lineattrs=(pattern=2
             thickness=2);
 xaxis label='Lot Size'
   labelattrs=(size=15pt);
 yaxis label='Work Hours'
   labelattrs=(size=15pt);
 title1 height=2
  'Comparison of Methods';
run;
```



Example 3.1: (Nonlinear Regression) Suppose $Y = \beta_0 + \beta_1 X_1^{\beta_2} - \beta_3 \exp(\beta_4 X_2) + \varepsilon$

```
/* Generate random data */
data temp;
 do i=1 to 50;
   X1 = 10+10*uniform(i);
   X2 = 1+2*uniform(i+2);
   error = 10*normal(2*i);
   output;
 end;
run;
/* uniform(A) --> U[0,1]
   normal(A) --> N(0,1)
   with seed A
 */

/* Define relation */
data temp1; set temp;
 Y=50+10*X1**2-16*exp(2*X2)+error;
run;

/* Look at plots */
proc sgplot data=temp1;
 scatter x=X1 y=Y /
markerattrs=(symbol=CIRCLEFILLED);
 title1 'Plots of Nonlinear Data';
run;
proc sgplot data=temp1;
 scatter x=X2 y=Y /
markerattrs=(symbol=CIRCLEFILLED);
run;
```






```
 /* Try proc nlin using the default loss function.
    The result would be the same if the pred and _LOSS_
    lines were deleted from the code. */
 proc nlin data=temp1 noitprint maxiter=500;
  pred = b0 + b1*X1**b2 + b3*exp(b4*X2);
  _LOSS_ = (Y-pred)**2;
  model Y = b0 + b1*X1**b2 + b3*exp(b4*X2);
  parameters b0=100 b1=8 b2=3 b3=-20 b4=4;
  title1 'proc nlin with [default] squared error loss function';
  title2 'truth: b0=50, b1=10, b2=2, b3=-16, b4=2';
  output out=out1 r=resid p=pred;
 run;
 /* What if we wanted better fits for smaller predicted values? */
  *_LOSS_ = ((Y-pred)/pred)**2;
```

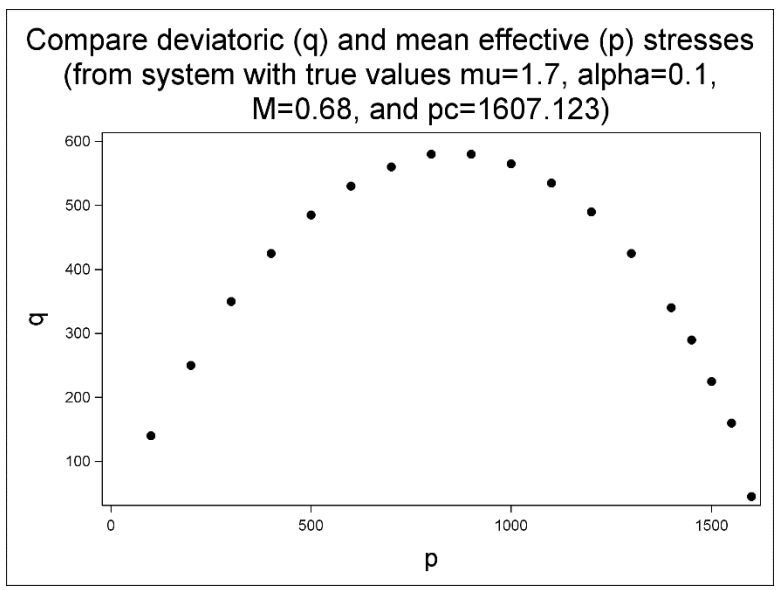
proc nlin with [default] squared error loss function
truth: b0=50, b1=10, b2=2, b3=-16, b4=2

NOTE: Convergence criterion met.

Parameter	Estimate	Approx Std Error	Approximate 95% Confidence Limits	
b0	32.9411	23.1548	-13.6950	79.5773
b1	10.1254	0.6771	8.7617	11.4891
b2	1.9970	0.0207	1.9554	2.0387
b3	-15.5777	0.2049	-15.9904	-15.1650
b4	2.0090	0.00450	1.9999	2.0180

Example 3.2: (Nonlinear Regression) A nonlinear curve to describe sand compression

```
data ex2; input p q @@; cards;
  100 140 200 250 300 350 400 425 500 485 600 530 700
  560 800 580 900 580 1000 565 1100 535 1200 490 1300
  425 1400 340 1450 290 1500 225 1550 160 1600 45
 ;
proc sgplot data=ex2;
 scatter x=p y=q / markerattrs=(symbol=CIRCLEFILLED);
 xaxis labelattrs=(size=15pt);
 yaxis labelattrs=(size=15pt);
 title1 h=2 'Compare
   deviatoric (q) and
   mean effective (p)
   stresses';
 title2 h=2 '(from system
   with true values mu=1.7,
   alpha=0.1,';
 title3 h=2 'M=0.68,
   and pc=1607.123)';
run;
```



```
proc model data=ex2;
 parms mu 1.7 alpha .2 M .7 ;
 bounds M mu > 0;
 control pc 1607.123;
 k1 = mu*(1-alpha)/(2*(1-mu)) *
                        (1+sqrt(1-4*alpha*(1-mu)/(mu*(1-alpha)**2)));
 k2 = mu*(1-alpha)/(2*(1-mu)) *
                        (1-sqrt(1-4*alpha*(1-mu)/(mu*(1-alpha)**2)));
 eq.f = p/pc - ((1+q/p/M/k2)**(k2/(1-mu)/(k1-k2)) /
                (1+q/p/M/k1)**(k1/(1-mu)/(k1-k2)));
 fit f  / method=marquardt prl=lr corrb;
 title1 'Sand stress example';
 title2 '(truth: mu=1.7, alpha=0.1, M=0.68)';
run;

/*
   parms -- sets initial starting estimates of parameters
            to be estimated in model

   bounds -- sets boundaries on parameter values

   control -- define fixed [known] constants

   k1, k2  -- functions of parameters to be estimated

   eq.f  -- expression that equals 0 (i.e., want to find
            parameter values to make eq.f=0)

   method -- specify estimation routine

   prl=lr -- requests CI on parameter estimates

   corrb  -- requests correlation matrix among parameter estimates

 */
```

Sand stress example
(truth: mu=1.7, alpha=0.1, M=0.68)

Nonlinear OLS Parameter Estimates				
Parameter	**Estimate**	**Approx Std Err**	**t Value**	**Approx Pr > |t|**
mu	1.67184	0.0181	92.49	<.0001
alpha	0.110909	0.00762	14.56	<.0001
M	0.677976	0.00215	314.83	<.0001

Parameter Likelihood Ratio 95% Confidence Intervals			
Parameter	**Value**	**Lower**	**Upper**
mu	1.6718	1.6352	1.7061
alpha	0.1109	0.0967	0.1267
M	0.6780	0.6736	0.6821

Correlations of Parameter Estimates			
	mu	**alpha**	**M**
mu	1.0000	-0.9117	0.7978
alpha	-0.9117	1.0000	-0.8644
M	0.7978	-0.8644	1.0000

4.3: Nonparametric Regression

Dr. Bean - Stat 5100

1 Why nonparametric regression?

For most of this course, we have assumed models of the form:

$$Y_i = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + \cdots + +\beta_{p-1} X_{i,p-1} + \epsilon.$$

Such models assume the following:

- Each explanatory variable shares a linear relationship with the response variable (perhaps aided by transformations).
  - In other words, after transformations, the rate of increase or decrease in Y is independent of the actual values of X.
- The effect of each explanatory variable can be isolated from the rest (assuming no interaction terms).
  - In other words, each explanatory variable is independent of all other explanatory variables.

What are some consequences associated with inappropriately assuming a linear model?

- If residual distributional assumptions are violated, there can be no meaningful model inference.
- Our accuracy will likely be poor if we assume the wrong model form.

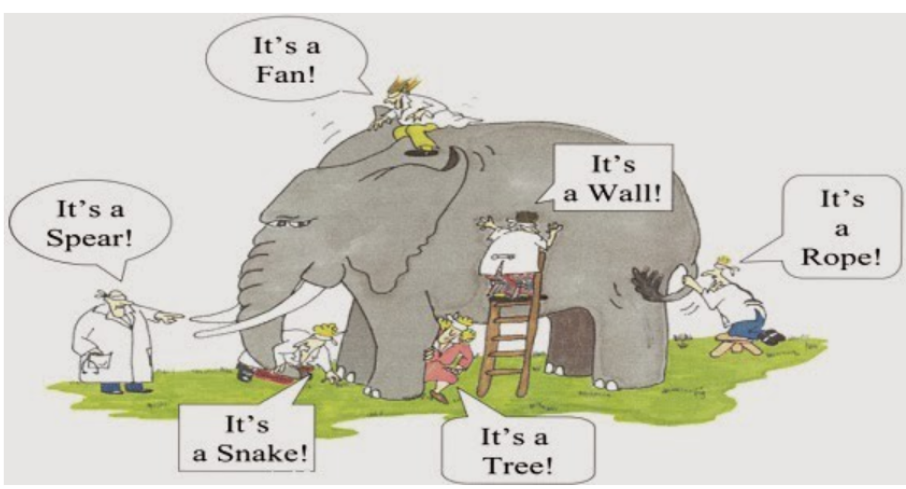


Figure 1: `https://medium.com/betterism/the-blind-men-and-the-elephant-596ec8a72a7d`

Nonparametric methods make far less (if any) assumptions about the form of the relationship between the explanatory and response variables.

The cost: Methods are often much more "data hungry" and harder to explain.

2 LOESS (local regression)

Close relative, lowess (local weighted regression scatter plot smoothing)

2.1 Assumptions

- Predictor variables are pre-selected
- The response function is "smooth." (i.e. small changes in any X_i, lead to relatively small changes in Y).
- Error terms are normal with constant variance.

2.2 Process

In order to make a prediction $\hat{Y}$ for a particular "X-profile" (i.e. combination of unique values for each explanatory variable)

1. (optional) standardize predictor variables X_i
2. For each observation i, calculate the distance to the current X-profile $X_{h,j}$

$$d_i = \sum_{j=1}^{p-1} (X_{i,j} - X_{h,j})^2$$

3. Let q = proportion of observations nearest to the current X-profile ($q \in (0, 1)$)
4. Let d_q = distance from X-profile to the furthest observation in the neighborhood as defined by q
5. For each observation i within that neighborhood, define weight

$$w_i = \begin{cases} \left(1 - \left(\frac{d_i}{d_q}\right)^3\right)^3 & d_i < d_q \\ 0 & otherwise \end{cases}$$

6. Using these weights, fit a weighted least squares (WLS) regression model based on polynomials of all predictors.
7. Use the WLS model to estimate $\hat{Y}$
   - Polynomial degree:
     - 0 - moving average
     - 1 - connected lines
     - 2 - smooth curves
     - (don't typically go higher than degree 2 as this can lead to unstable fits)

2.3 Implementation

LOESS requires the user to select the smoothing parameter q. (See Figure 2.)

- Larger q $\rightarrow$ smoother fit
- Smaller q $\rightarrow$ "choppy fit"

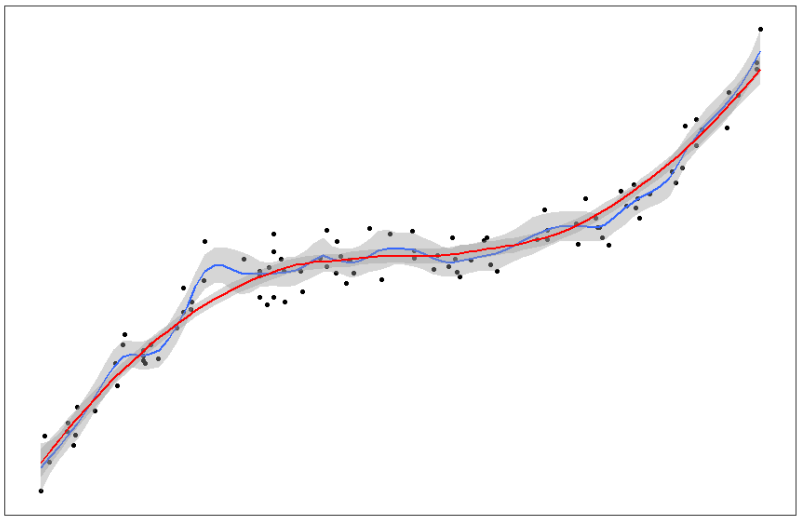


Figure 2: Example LOESS smoothing curves with only one X-variable and two levels of smoothness.

- Advantages
  - Flexible response surface - do not have to worry about whether or not the data share a linear relationship.
- Disadvantages:
  - Requires "dense" data to get good predictions.
    - Method extremely sensitive to outliers in "sparse" data regions.
  - No "model" to report - no inference.

In general, the less our model *assumes*, the more data we must *consume*.

3 Regression Trees

Simple, yet powerful way to handle high-ordered interactions between variables.

3.1 Process

- Separate the data into two **branches** by splitting the data in a way that minimizes the sum of squares error $\sum_i \left(Y_i - \hat{Y}_i\right)^2$ (or a similar metric).
  - Predictions $\hat{Y}_i$ in this case is the average of the values in each **terminal node or leaf** (i.e. the group of values that fall into each branch at the end of the tree).
- Keep splitting the subgroups over and over until all nodes are completely **pure** ($\sum_i \left(Y_i - \hat{Y}_i\right)^2 = 0$).
  - This may mean that each terminal node in the **fully grown** tree will be single observations.
- Because a model that perfectly predicts the training data is obviously overfit, we will **prune** the tree back to a set of cuts that balances accuracy with simplicity.
  - Typically picked using a **cost complexity parameter**:

$$CC(T) = R(T) + \alpha|T|$$

    * $CC(T)$ - cost complexity
    * $R(T)$ - error rate (such as average squared error)
    * α - user selected cost parameter (controls size of tree).
    * $|T|$ - number of nodes in the tree
  - Alternatively, complexity can be defined using restrictions on the tree such as:
    * Minimum number of observations in a terminal node.
    * Minimum percentage increase in the percent variance explained in order for a split to be conducted.

Example: predicting snow density using climate reanalysis data.
Variables

- maxv_SNWD - the depth of the snowpack (mm)
- TD - difference in the mean annual temperature between the coldest and the warmest month of the year (degrees Celsius)
- PPTWT - total winter (Dec to Feb) precipitation.

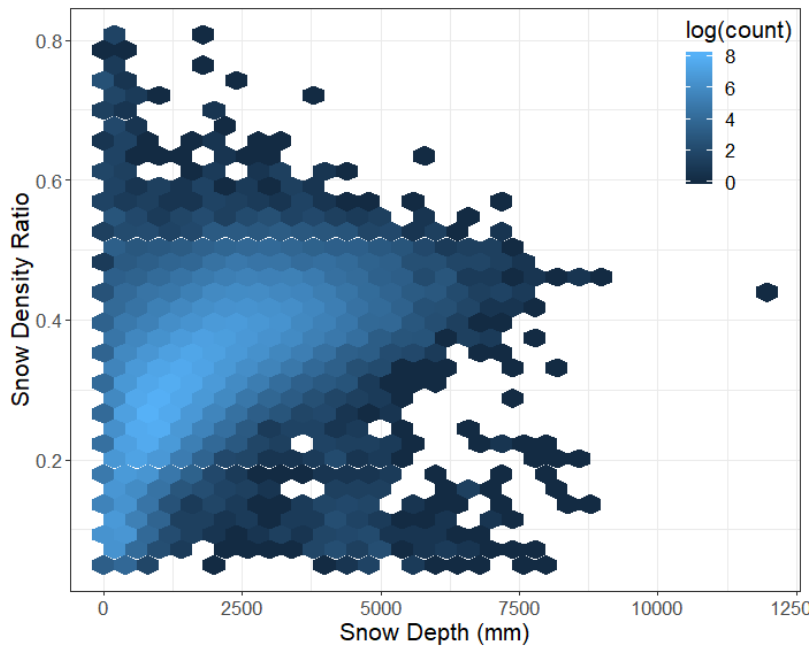


Figure 3: Plot of the snow density ratio in relation to its depth for locations across North America.

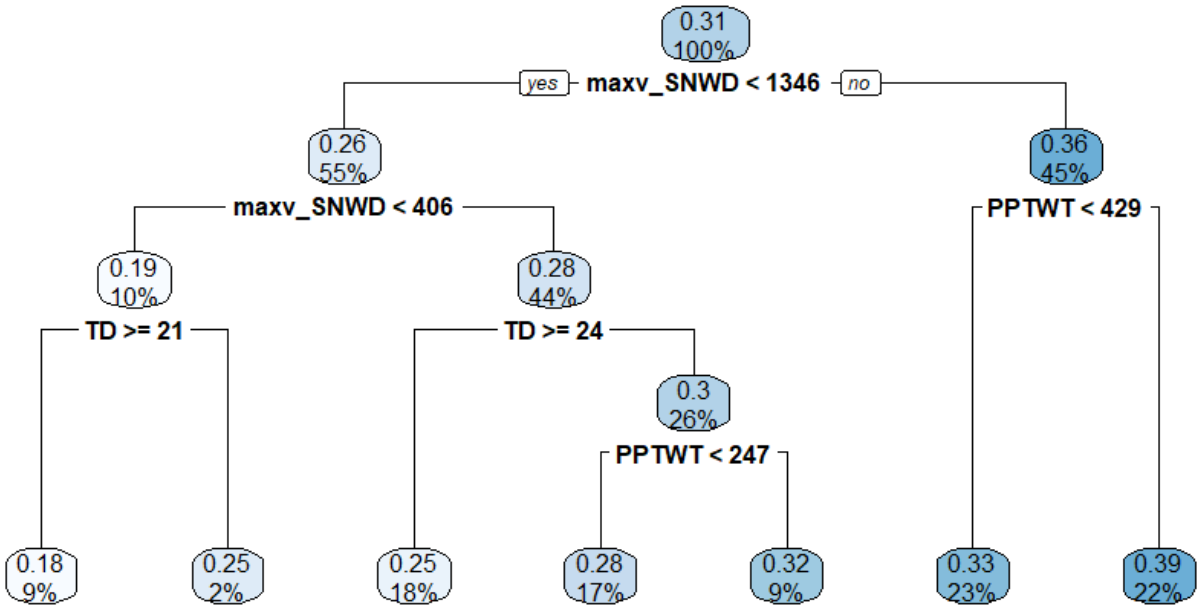


Figure 4: Sample tree (pruned) for predicting snow ratio using climate variables.

3.2 Variable Importance

There are several ways in which we can explore the importance of variables in a regression tree.

- **Count:** Variables that are used *more often* for splitting are more important.
- **Error Reduction:** The greater reduction in the SSE resulting from splitting on a variable, the more important a variable.

3.3 Extensions of Regression Trees

- Boosting - fit tree in an iterative fashion, re-weighting the observations for the next split depending on the values of the residuals from the previous split.

  Essentially, a combination of "weak" trees that together provide a stronger prediction.

- Bagging - fit many trees, with each tree using a bootstrap sample of the training data.
  - Final predictions for an observation are simply the average prediction from each tree.

- Methods that combine/average predictions from a group of simpler models are called "ensemble methods."

Why might ensemble-based approaches provide better (more accurate) predictions when compared to a single regression tree?

Taking the **average** of a set of predictions has the effect of reducing the **variance** of the overall prediction. Reductions in variance lead to an overall increase in accuracy.

4 Random Forest

A clever ensemble based method that was created by Leo Breiman and USU's own Adele Cutler.

An extension of bagging that, in addition to taking bootstrap samples of the original data for each tree, also only considers a random subset of the variables when deciding how to split the tree at each node.

- The random sub-setting of the variables helps differentiate the trees, which further reduces the variance of the predictions.

SAS:

```
proc hpforest data=<dataset> seed=<random seed> scoreprole=oob;
input <all my explanatory variables>
target <my response variable>;
ods output <outputs you want printed to screen>
run;
```

4.0.1 Model Accuracy

- Bootstrap samples are samples with replacement from the original data, which means some observations show up more than once in each sample, and other observations do not show up at all.

- This means that each observation will have been ignored when creating some subset of the trees.

- We can determine the out of bag (OOB) error rate by making predictions using only the trees from which a particular observation was not included in the fitting.

4.0.2 Variable Importance

Random Forest includes a powerful measure of variable importance:

- For each tree, look at the OOB and random permute (scramble) the values of a single predictor variable X_j.
- Pass the OOB data with the scrambled X_j information down the tree - obtain the OOB error rate.
- Compare this error with the OOB error obtained when X_j was not scrambled.
- The worse the error rate is with the scrambled X_j information, the more important X_j is to the model.

4.0.3 Limitations

- Random forests is an extremely powerful method, but is often referred to as a "black box" algorithm because it does not produce a model.
- The lack of model makes random forest more difficult to interpret.
- Random forests does offer **partial dependence plots**, which visualize the effect of each predictor holding all others constant, but these are not implemented in SAS.
- Alternatively, one can get a **generalized additive model** to try and visualize the effect of each predictor.

$$Y_i = s_0 + s_1\left(X_{i,1}\right) + \cdots + s_{p-1}\left(X_{i,p-1}\right) + \epsilon_i$$

5 Helpful Resources

(both from USU's Dr. Richard Cutler):

- "What Statisticians Should Know about Machine Learning"' (2017 SAS Global Forum proceedings) `https://support.sas.com/resources/papers/proceedings17/0883-2017.pdf`
- "Prediction and Interpretation for Machine Learning Regression Methods" (2018 SAS Global Forum proceedings) `https://pdfs.semanticscholar.org/eade/6d9e5a9e5e3667cb2f88c665638735c8` `pdf`

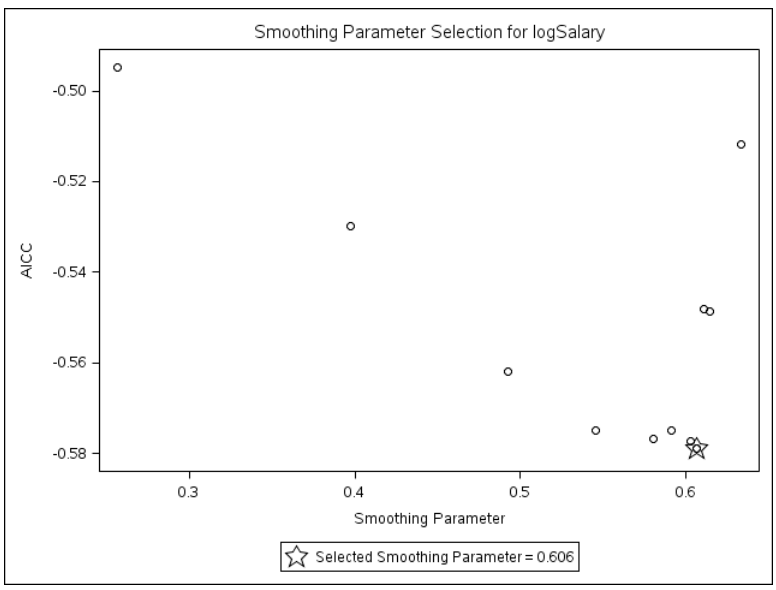
Remember, the less our model *assumes*, the more data we must *consume*.

4.3.1- SAS: Nonparametric Regression Methods (LOESS, Regression Trees, and Random Forests)

Example: (Baseball, same as Handout 4.1.1)

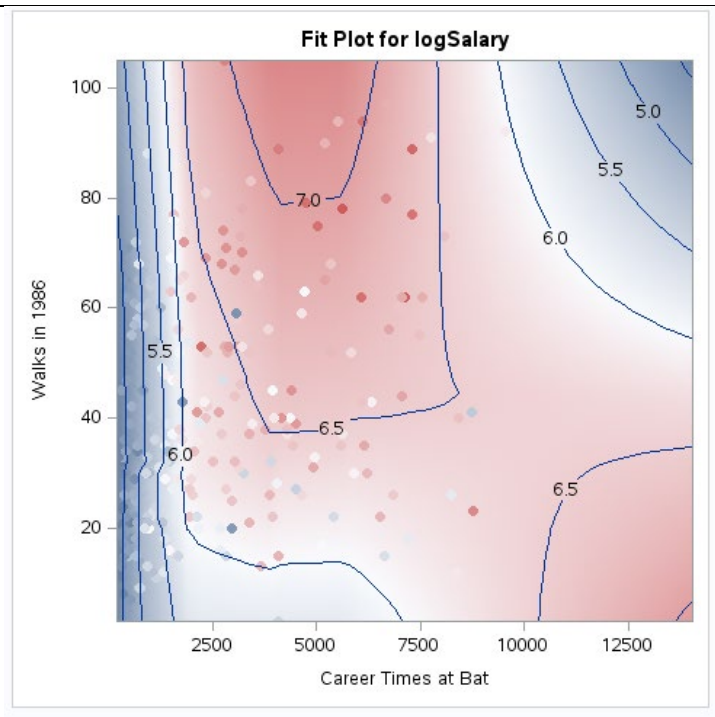
```
data baseball; set sashelp.baseball;
  AmerLg = (League="American");
  EastDv = (Division="East");
run;


/* loess */
proc loess data=baseball plots=(fitpanel fitplot contourfitpanel
contourfit);
 model logSalary =  crAtBat nBB
                  / degree=2 select=AICC scale=sd;
 output out=out1 p=predloess;
run;
```

The LOESS Procedure

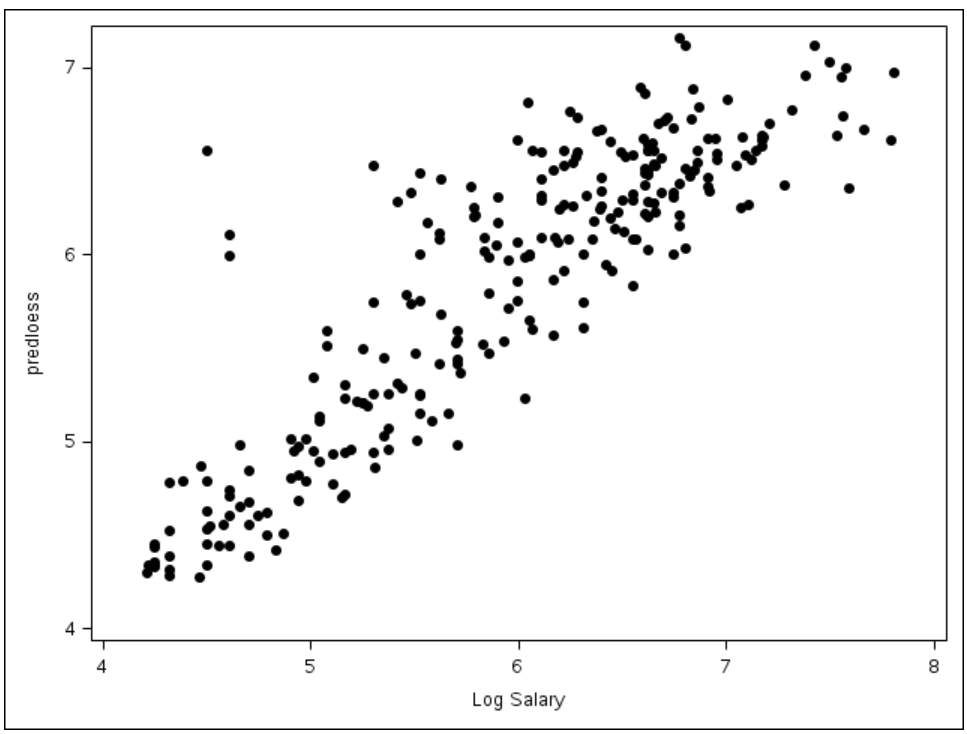


Optimal Smoothing Criterion

AICC	*Smoothing Parameter*
-0.57876	0.60646

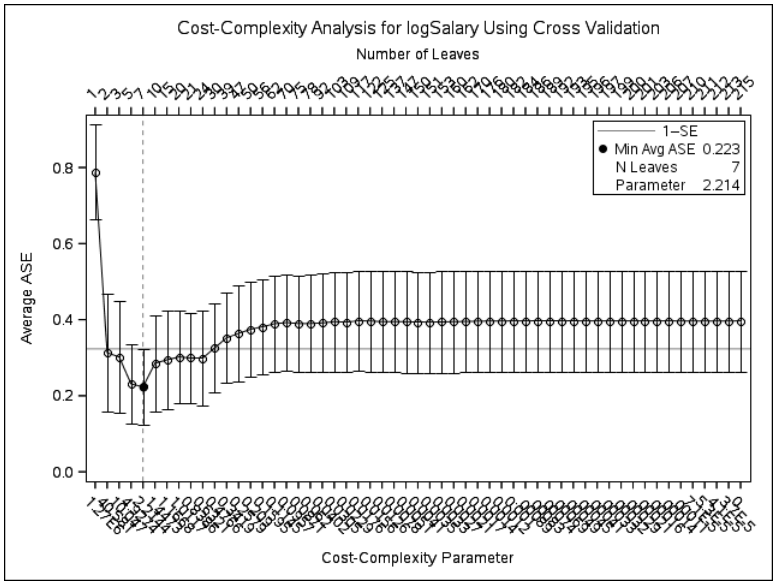


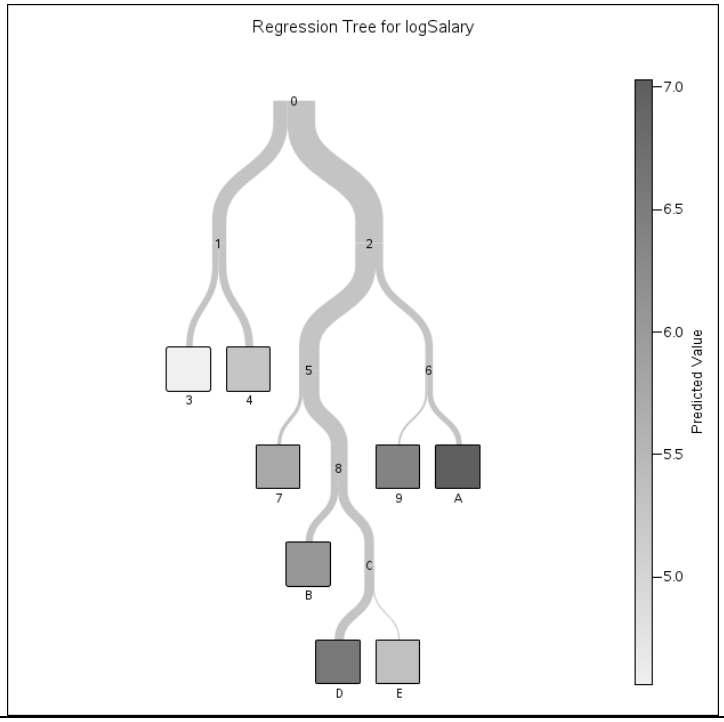

```
proc sgplot data=out1;
  scatter x=logSalary y=predloess /
markerattrs=(symbol=circlefilled size=6pt);
run;
```

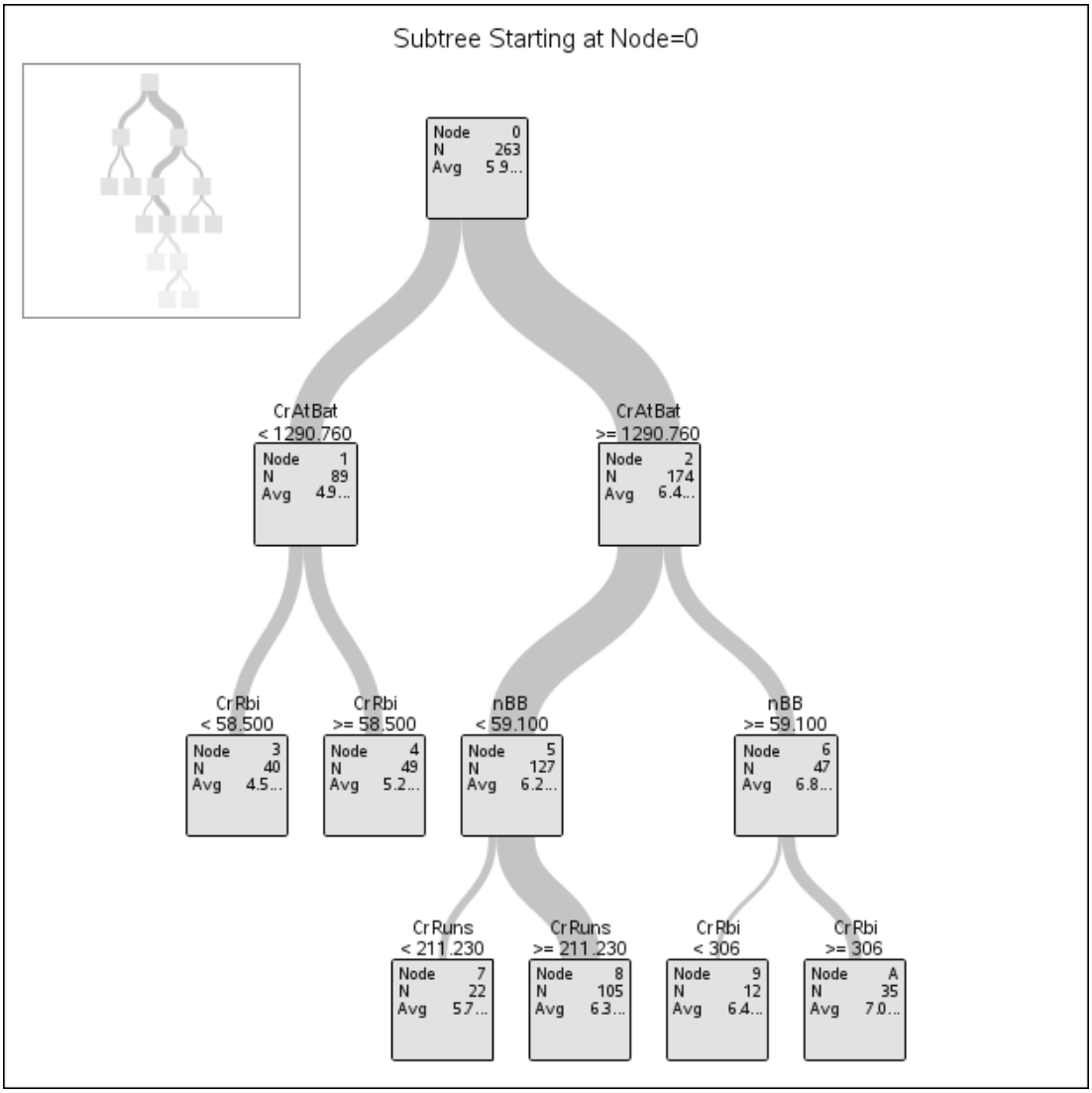


```
/* regression tree */
proc hpsplit data=baseball seed=123 maxdepth=15 maxbranch=2;
 class league division;
 model logSalary = nAtBat nHits nHome nRuns nRBI nBB
                   yrMajor crAtBat crHits crHome crRuns crRbi
                   crBB league division nOuts nAssts nError;
 output out=out2;
run;
```

The HPSPLIT Procedure







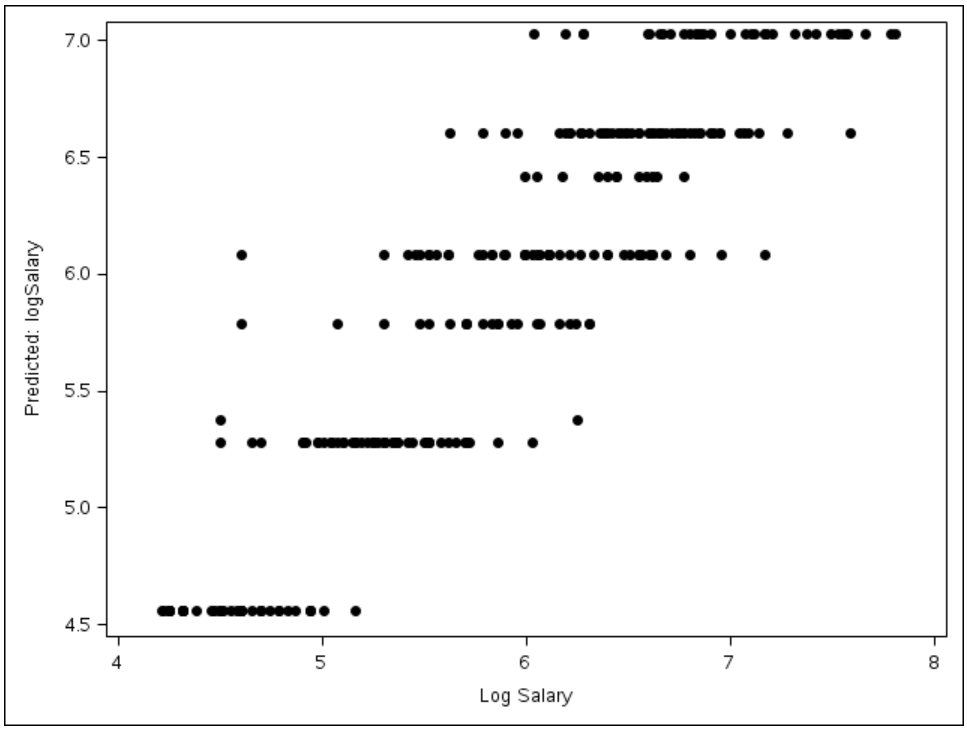
Model-Based Fit Statistics for Selected Tree

N Leaves	*ASE*	*RSS*
8	0.1443	37.9587

Variable Importance

Variable	*Variable Label*	*Training Relative*	*Training Importance*	*Training Count*
CrAtBat	Career Times at Bat	1.0000	11.2539	1
nBB	Walks in 1986	0.3546	3.9905	2
CrRbi	Career RBIs	0.3414	3.8415	2
nAtBat	Times at Bat in 1986	0.2168	2.4397	1
CrRuns	Career Runs	0.2161	2.4316	1

```
proc sgplot data=out2;
  scatter x=logSalary y=p_logSalary /
markerattrs=(symbol=circlefilled size=6pt);
run;
```



Question: What is going on in this plot? Do these patterns in the prediction make sense? If yes, why do they make sense?

Question: Recalling Output in Handout 4.1.1, what do the "important" variables have in common?

```
/* random forest */
proc hpforest data=baseball seed=134 scoreprole=oob;
 input nAtBat nHits nHome nRuns nRBI nBB
                  yrMajor crAtBat crHits crHome crRuns crRbi
                  crBB league division nOuts nAssts nError;
 target Salary;
 ods output FitStatistics=fitstats
   VariableImportance=varimp;
run;
```

The HPFOREST Procedure

Model Information

Parameter	Value	
Variables to Try	4	(Default)
Maximum Trees	100	(Default)
Missing Value Handling	.	Valid value

Number of Observations

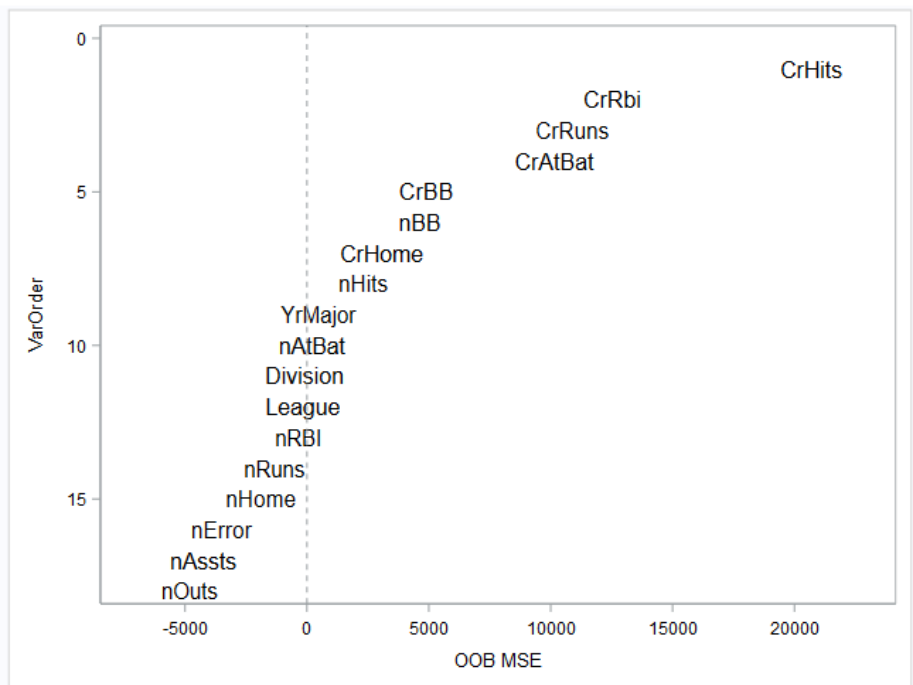
Type	N
Number of Observations Read	322
Number of Observations Used	263

Loss Reduction Variable Importance

Variable	Number of Rules	MSE	OOB MSE	Absolute Error	OOB Absolute Error
CrHits	907	27941.87	20687.57	48.803825	34.608172
CrRbi	1160	22995.54	12521.15	35.533126	19.290786
CrRuns	1072	23108.48	10892.41	39.211686	18.379497
CrAtBat	751	18859.52	10140.97	32.764124	20.230476
CrBB	1364	16893.90	4896.42	31.277359	11.410166
nBB	606	12942.85	4625.19	14.772798	3.751437
CrHome	804	13002.18	3062.38	18.501506	4.823677
nHits	439	10636.46	2314.45	14.907649	3.961956
YrMajor	455	5866.65	471.24	11.912504	2.927752
nAtBat	414	10120.05	199.98	14.692048	0.552953
Division	9	355.44	-102.12	0.373370	-0.103367
League	15	117.50	-174.16	0.244754	-0.153395
nRBI	572	11899.64	-352.58	15.151606	-0.354135
nRuns	497	8491.47	-1336.94	11.766502	-0.471976
nHome	423	5302.24	-1882.58	8.979994	-0.764283
nError	1755	4534.88	-3505.17	13.465747	-3.311704
nAssts	1582	3494.33	-4257.11	12.493737	-3.871985
nOuts	1802	9530.72	-4815.96	21.164897	-4.546558

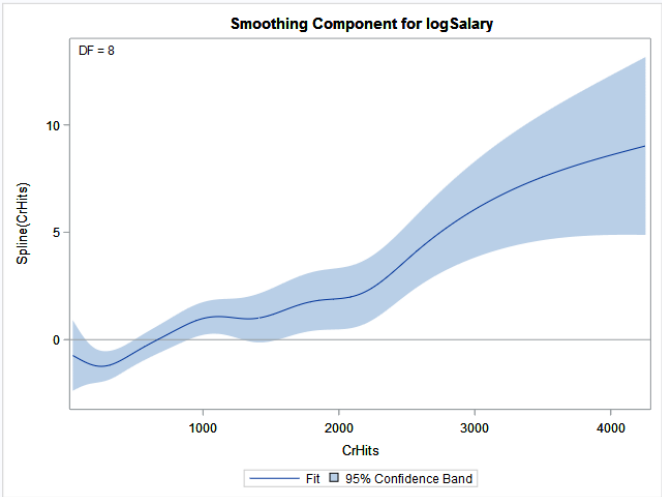
Question: What does it mean to have a negative out of bag mean square error? What does this provide evidence for?

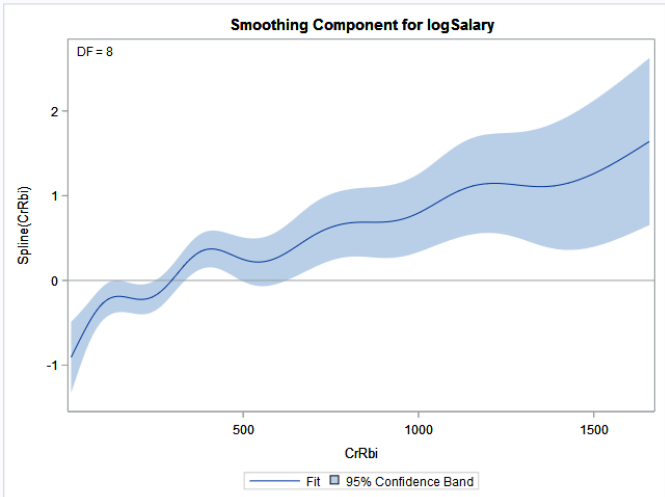
```
data varimp; set varimp;
 VarOrder=_n_;
proc sgplot data=varimp;
 scatter x=MSEOOB y=VarOrder / markerchar=Variable
markercharattrs=(size=12);
 yaxis reverse;
 refline 0 / axis = x LINEATTRS=(pattern=2);
run;
```

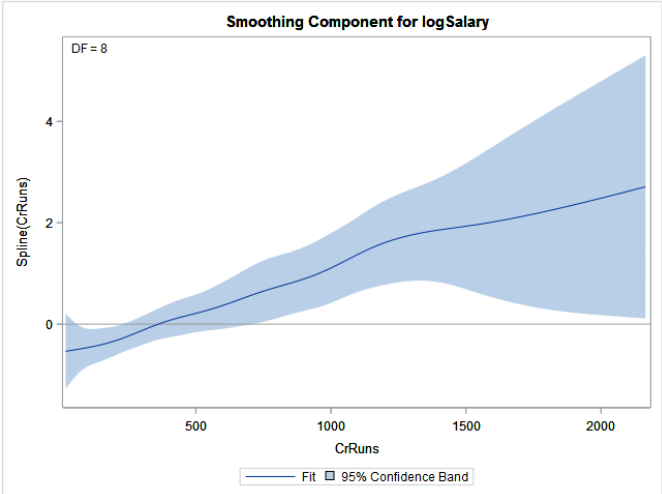


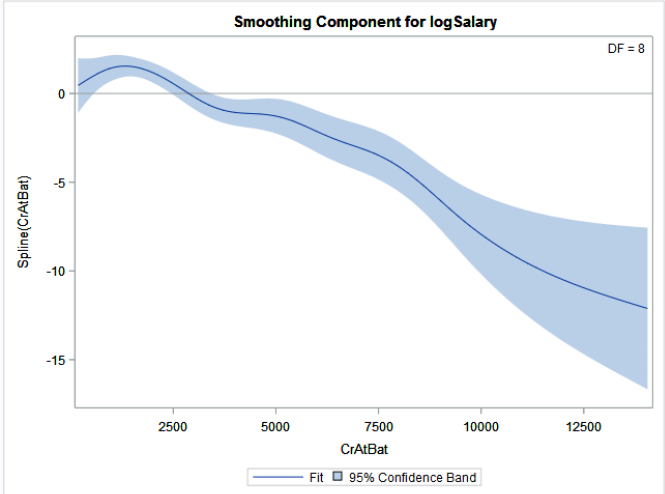
```
/* Visualize effects of top predictors using a generalized
additive model */
proc gampl data=baseball plots(unpack)=all;
  model logSalary = s(crHits) s(CrRbi) s(CrRuns) s(CrAtBat)
    / dist=norm;
run;
```

The GAMPL Procedure









```
/* Compare with simple scatter plot */
proc sgscatter data=baseball;
 matrix logSalary crHits crRBI
    crRuns crAtBat /
    markerattrs=(
       symbol=CIRCLEFILLED
       size=6pt);
run;
```