3.1 Alternate Variable Types and Interactions

Dr. Bean - Stat 5100

1 Why Interactions?

Example (HO 3.1.1): $Y =$ `cycles`, $X_1 =$ `charge_rate`, $X_2 =$ `temperature`

All models we have discussed in this class assume that the effects of the explanatory variables are **additive**.

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \varepsilon_i$$

In other words, the effect of each explanatory variable can be considered **separate** from all other explanatory variables.

What if the **real** effect of X_1 on Y actually depends on X_2 as well?

What would it mean for the effect of `charge_rate` on `cycles` to depend on `temperature`?

- We "know": higher `charge_rate` $\rightarrow$ lower `cycles`, and
  higher `temperature` $\rightarrow$ higher `cycles`
- But maybe: higher `charge_rate` **and** higher `temperature` $\rightarrow$ **much** higher `cycles`
- "**much**" higher here: significantly more than could be attributed to the sum of the effects of `charge_rate` and `temperature` only (often called **synergy**)

Whenever the effect of an explanatory variable (X_k) on the response (Y) *depends on* the values of other explantory variables, you have an **interaction effect**.

Metaphor: The bachelorette - the relationship of each potential suitor (X_k) with the bachelorette (Y) is partially depends upon the other potential suitors.

How is an interaction effect different from multicollinearity?

Muticollinearity only has to do with relationships among the X_k and has nothing to do with Y. Interactions have everything to do with the relationship between the X_k's and Y.

Define an interaction term as a new predictor variable:

$$\begin{aligned} X_3 &= X_1 \cdot X_2 \\ Y_i &= \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i3} + \varepsilon_i \\ &= \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i1} X_{i2} + \varepsilon_i \end{aligned}$$

Note: sometimes β_{12} instead of β_3

1.1 How to interpret interaction terms?

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \varepsilon$$

- if X_1 increases by 1 unit, then we expect an average change of $\beta_1 + \beta_3 X_2$ in Y
  - the effect of X_1 on Y depends on X_2
  - if the interaction term is non-zero, we *cannot* separate the effect of X_1 from the effect of X_2. We must consider them jointly (unless X_1 or $X_2 = 0$).

1.2 Best Practices

- Don't check all possible interactions. Only include an interaction term in a linear model if its output is interpretable.
- Include all lower-ordered terms that compose an interaction term, regardless of the significance of the lower interaction term.
  - Prevents forcing lower ordered coefficients to zero.
  - Maintains a flexible response surface and facilitates interpretation.

1.3 Things to remember about interactions:

- Unless the X_k are standardized, the interaction term $X_3 = X_1 * X_2$ is likely to be collinear with either X_1 or X_2.
  - This will ruin inference for the "lower order" terms, but not the interaction term.
- Two-way interactions are often interpretable, but higher order interactions (ex: $X_4 = X_1 * X_2 * X_3$) become difficult to interpret.
  - A plot of residuals from a non-interaction model against the potential interaction term may help to determine inclusion (if a trend is apparent).
- If your problem is best solved by including multiple, high-ordered, interaction terms, then regression trees/random forests is likely a better approach (more in Module 4).

1.4 Polynomial Predictors

- Up to this point, we have limited ourselves to modeling variables that share a linear relationship.
- If a variable X_k shares a quadratic, or higher-order (often called "curvilinear") relationship with Y, then that means that the effect of X_k on Y *depends upon itself* (i.e. interacts with itself).

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \beta_4 X_1^2 + \beta_5 X_2^2 + \varepsilon$$

- Handle higher-ordered terms the same way we handle other interaction terms:
  - include lower-order terms
  - standardize to reduce multicollinearity

- coefficient interpretations important: – if X_1 increases by 1 unit (and X_2 held constant), then we expect an average change in Y of $\beta_1 + \beta_3 X_2 + \beta_4 \cdot (2X_1 + 1)$

For those who have taken calculus, you may see a relationship between one unit increase in X_k with the $\frac{\partial Y}{\partial X_k}$.

2 Alternate Variable Types

Up to this point we have only focused on **quantitative variables**:

- Values are represented as numbers where number *order* and *magnitude* matters.
- Quantiative variables can be either:
  - Continuous: can take on any value (theoretically infinite number of decimal places) within a range.
  - Discrete: can only take on a discrete (countable) set of values.

We now wish to also consider **qualitative variables**

- Cannot be measured/ordered on a numerical scale.
- SAS can't recognize words/letters in a regression model, and it will treat a set of numbered factored levels as quantitative (and thus order the levels).
- Because of this, we use **dummy/indicator variables** to include qualitative predictors in a model.

2.1 Dummy Variables

Consider the following student demographic variables (qualitative in bold): (age, height, **Utah residency status**, weight, **major college**)

Use an indicator variable to include residency status in model

$$X = I_{\text{resident}} = \begin{cases} 1 & \text{if student is resident of Utah} \\ 0 & \text{otherwise} \end{cases}$$

Things get a little more complicated for major college as we have to create multiple dummy variables to represent a single categorical variable:

$$X_1 = I_{\text{College of Science}} = \begin{cases} 1 & \text{if student's major is within the college of science} \\ 0 & \text{otherwise} \end{cases}$$

$$X_2 = I_{\text{College of Engineering}}$$

$$\vdots$$

$$X_7 = I_{\text{School of Business}}$$

If there are eight colleges in the University, why would I only have seven dummy variables?

Values of 0 for all seven indicator variables means the person is a member of the eighth college. This college would be referred to as the base class on which all things are compared.

3 Example (See HO 3.1.1)

Y = months, X_1 = size, X_2 = type of firm

Note that $X_2 = I_{[\text{firm = stock}]} = \begin{cases} 1 & \text{if firm = stock} \\ 0 & \text{otherwise} \end{cases}$

Model with only qualitative predictor:

$$Y = \beta_0 + \beta_2 X_2 + \varepsilon$$

- equivalent to a two-sample t-test
- special case of one-way ANOVA model (`proc glm`, STAT 5200)

$$\begin{aligned} Y_{i,j} &= \mu_i + \epsilon_{i,j}, \qquad i = 1, 2; j = 1, \ldots, n_i \\ &= \mu + \alpha_i + \epsilon_{i,j}, \qquad \sum_{i=1}^{2} \alpha_i = 0 \\ \epsilon_{i,j} &\; iid \;\; N(0, \sigma^2) \end{aligned}$$

Model with both qualitative and quantitative predictor:

- Additive

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$$

- Interaction

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \varepsilon$$

Note how the additive and interaction models differ:
(in the size (X_1) vs. months (Y) relationship for each firm type)

- Additive:
  - stock ($X_2 = 1$): $Y = (\beta_0 + \beta_2) + \beta_1 X_1 + \varepsilon$
  - mutual ($X_2 = 0$): $Y = \beta_0 + \beta_1 X_1 + \varepsilon$
- Interaction

- stock ($X_2 = 1$): $Y = (\beta_0 + \beta_2) + (\beta_1 + \beta_3)X_1 + \varepsilon$
- mutual ($X_2 = 0$): $Y = \beta_0 + \beta_1 X_1 + \varepsilon$

Note that the additive model results in *two parallel lines*, where the difference between stock and mutual firms are separated by a constant distance β_2. Whereas in the interaction model, both the slope *and* the intercept are different.

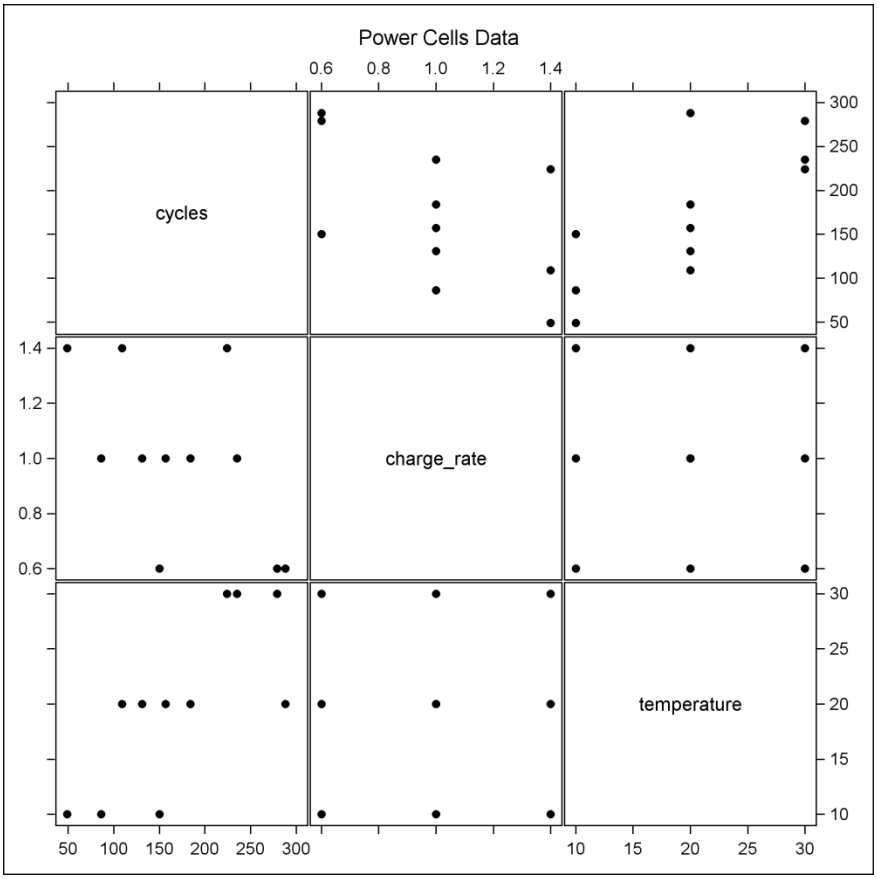
3.1 Note on interactions between qualitative predictors.

- possibly very interesting
- numerically much easier in [two-way] ANOVA setting (`proc glm`, STAT 5200), as ANOVA doesn't require the use of dummy variables.

Stat 5100 Handout 3.1.1 – SAS: Alternative Predictor Variable Types

Example 1: (Table 8.1) Study looks at the effects of the charge rate and temperature on the life of a new type of power cell. A small-scale preliminary study was conducted using 11 power cells. Variables reported are the charge rate (X1, in amperes), the ambient temperature (X2, in degrees Celsius), and the life of the power cell (Y, in the number of discharge-charge cycles before failure).

```
/* Input data -- see Table 8.1 in text */
data powercells;
   input cycles charge_rate temperature; cards;
  150   0.6   10
   86   1.0   10
   49   1.4   10
  288   0.6   20
  157   1.0   20
  131   1.0   20
  184   1.0   20
  109   1.4   20
  279   0.6   30
  235   1.0   30
  224   1.4   30
;
run;
```




```
/* Look at shape of relationships with Y */
proc sgscatter data=powercells;
  matrix cycles charge_rate temperature /
    markerattrs=(symbol=CIRCLEFILLED size=2pt);
  title1 'Power Cells Data';
run;
```

```
/* Define higher-order predictors */
data powercells; set powercells;
  cr_temp = charge_rate*temperature;
  cr2 = charge_rate**2;
  temp2 = temperature**2;
run;

proc reg data=powercells;
  model cycles = charge_rate temperature cr_temp / vif;
  title1 'Check for interaction';
run;
```

Check for interaction

Analysis of Variance					
Source	**DF**	**Sum of Squares**	**Mean Square**	**F Value**	**Pr > F**
Model	3	53435	17812	17.39	0.0013
Error	7	7171.33333	1024.47619		
Corrected Total	10	60606			

Parameter Estimates						
Variable	**DF**	**Parameter Estimate**	**Standard Error**	**t Value**	**Pr > |t|**	**Variance Inflation**
Intercept	**1**	218.08333	90.80890	2.40	0.0474	0
charge_rate	**1**	-197.08333	86.42997	-2.28	0.0566	7.00000
temperature	**1**	4.67500	4.20891	1.11	0.3034	10.37500
cr_temp	**1**	2.87500	4.00093	0.72	**0.4957**	16.37500

```
proc reg data=powercells;
  model cycles = charge_rate temperature cr_temp cr2 temp2
     / vif;
  highercheck: test cr_temp=cr2=temp2=0;
  title1 'Check for higher-order predictors';
run;
```

Check for higher-order predictors

Analysis of Variance					
Source	**DF**	**Sum of Squares**	**Mean Square**	**F Value**	**Pr > F**
Model	5	55366	11073	10.57	0.0109
Error	5	5240.43860	1048.08772		
Corrected Total	10	60606			

Parameter Estimates						
Variable	**DF**	**Parameter Estimate**	**Standard Error**	**t Value**	**Pr > |t|**	**Variance Inflation**
Intercept	**1**	337.72149	149.96163	2.25	0.0741	0
charge_rate	**1**	-539.51754	268.86033	-2.01	0.1011	66.21053
temperature	**1**	8.91711	9.18249	0.97	0.3761	48.26974
cr_temp	**1**	2.87500	4.04677	0.71	**0.5092**	16.37500
cr2	**1**	171.21711	127.12550	1.35	**0.2359**	60.28708
temp2	**1**	-0.10605	0.20340	-0.52	**0.6244**	38.97129

Test highercheck Results for Dependent Variable cycles				
Source	**DF**	**Mean Square**	**F Value**	**Pr > F**
Numerator	3	819.96491	0.78	**0.5527**
Denominator	5	1048.08772		

```
proc reg data=powercells;
  model cycles = charge_rate temperature;
  title1 'Lower-order model';
run;
```

Lower-order model

Analysis of Variance					
Source	**DF**	**Sum of Squares**	**Mean Square**	**F Value**	**Pr > F**
Model	2	52906	26453	27.48	0.0003
Error	8	7700.33333	962.54167		
Corrected Total	10	60606			

Parameter Estimates						
Variable	**DF**	**Parameter Estimate**	**Standard Error**	**t Value**	**Pr > |t|**	**Variance Inflation**
Intercept	**1**	160.58333	41.61545	3.86	0.0048	0
charge_rate	**1**	-139.58333	31.66461	-4.41	0.0023	1.00000
temperature	**1**	7.55000	1.26658	5.96	0.0003	1.00000

```
/* Now look at higher-order variables with standardized
data */
proc stdize data=powercells out=std_powercells
    method=std mult=.3162;
run; /* Note that mult = 1/sqrt(n-1) */

data std_powercells; set std_powercells;
  cr_temp = charge_rate*temperature;
  cr2 = charge_rate**2;
  temp2 = temperature**2;
run;

proc reg data=std_powercells;
  model cycles = charge_rate temperature cr_temp / vif;
  title1 'Check for interaction (standardized scale)';
run;
```

Check for interaction (standardized scale)

Parameter Estimates						
Variable	**DF**	**Parameter Estimate**	**Standard Error**	**t Value**	**Pr > |t|**	**Variance Inflation**
Intercept	**1**	-1.431E-17	0.03920	-0.00	1.0000	0
charge_rate	**1**	-0.55553	0.13001	-4.27	0.0037	1.00000
temperature	**1**	0.75122	0.13001	5.78	0.0007	1.00000
cr_temp	**1**	0.28030	0.39008	0.72	**0.4957**	1.00000

```
proc reg data=std_powercells;
  model cycles = charge_rate temperature cr_temp cr2 temp2
   / vif;
  highercheck: test cr_temp=cr2=temp2=0;
  title1 'Check for higher-order predictors (standardized
scale)';
run;
```

Check for higher-order predictors (standardized scale)

Parameter Estimates						
Variable	**DF**	**Parameter Estimate**	**Standard Error**	**t Value**	**Pr > |t|**	**Variance Inflation**
Intercept	**1**	-0.03720	0.06745	-0.55	0.6051	0
charge_rate	**1**	-0.55553	0.13150	-4.22	0.0083	1.00000
temperature	**1**	0.75122	0.13150	5.71	0.0023	1.00000
cr_temp	**1**	0.28030	0.39455	0.71	**0.5092**	1.00000
cr2	**1**	0.66773	0.49577	1.35	**0.2359**	1.07656
temp2	**1**	-0.25850	0.49577	-0.52	**0.6244**	1.07656

Test highercheck Results for Dependent Variable cycles				
Source	**DF**	**Mean Square**	**F Value**	**Pr > F**
Numerator	3	0.01353	0.78	**0.5527**
Denominator	5	0.01729		

```
/* NOTE: You don't need to standardize predictors to look at
         higher-order predictors like this.  Instead, you can
         include a higher-order predictor and test it; if
         not significant, drop it; if significant, don't worry
         about significance of lower-order term. If higher-order
         term is significant and you really need to look at
         significance of lower-order term, or if the context of
         the data would allow the lower-order and higher-order
         terms to be 'stand-alone' interpretable, then
         standardize.

         Tests for higher-order terms are the same whether
         data are standardized or not.

*/
```

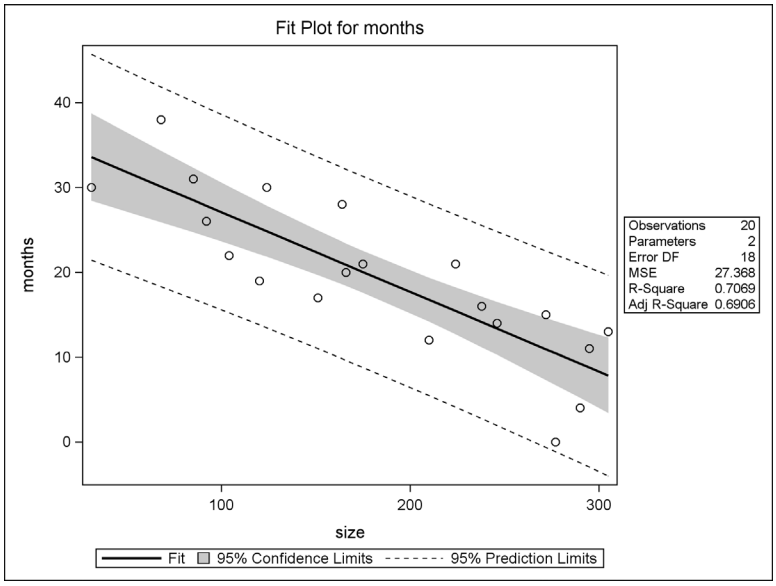
Example 2: An economist wishes to relate the speed with which a particular insurance innovation is adopted (Y, in months) to the size of the insurance firm (X1, in millions of dollars) and the type of firm (X2, either mutual (0) or stock firms (1)).

```
/* Input data -- see Table 8.2 of text */
data insurance; input months size type @@; cards;
  17   151   0      26    92   0      21   175   0      30    31   0
  22   104   0       0   277   0      12   210   0      19   120   0
   4   290   0      16   238   0      28   164   1      15   272   1
  11   295   1      38    68   1      31    85   1      21   224   1
  20   166   1      13   305   1      30   124   1      14   246   1
  ;

/* Model with only quantitative predictor */
proc reg data=insurance;
  model months = size;
  title1 'Single quantitative predictor';
  output out=out1 p=pred;
run;
```

Single quantitative predictor

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	**1**	36.48211	2.84425	12.83	<.0001
size	**1**	-0.09394	0.01426	-6.59	<.0001



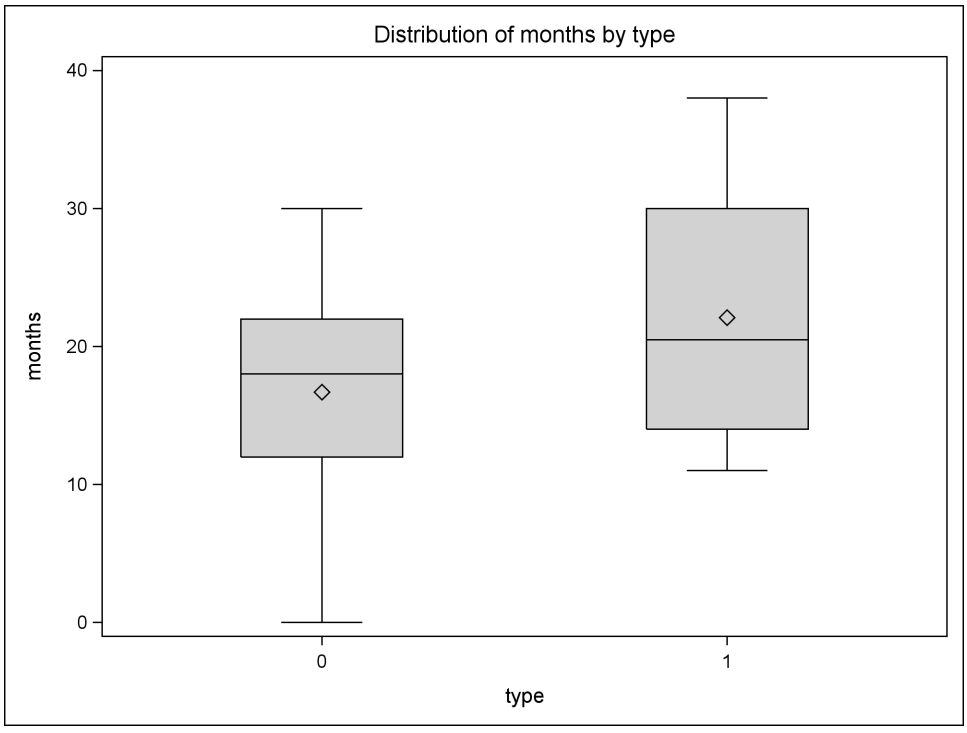
```
/* Model with only qualitative predictor */
proc reg data=insurance;
  model months = type;
  title1 'Single qualitative predictor';
run;
```

Single qualitative predictor

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	16.70000	2.92024	5.72	<.0001
type	1	5.40000	4.12984	1.31	0.2075

```
proc sort data=insurance out=sort_ins; by type;
proc boxplot data=sort_ins;
 plot months*type /
    boxstyle=schematic boxwidth=30 haxis=axis1
    cboxfill=yellow cboxes=blue;
 axis1 order=(.5 to 1.5 by .5);
run;
```



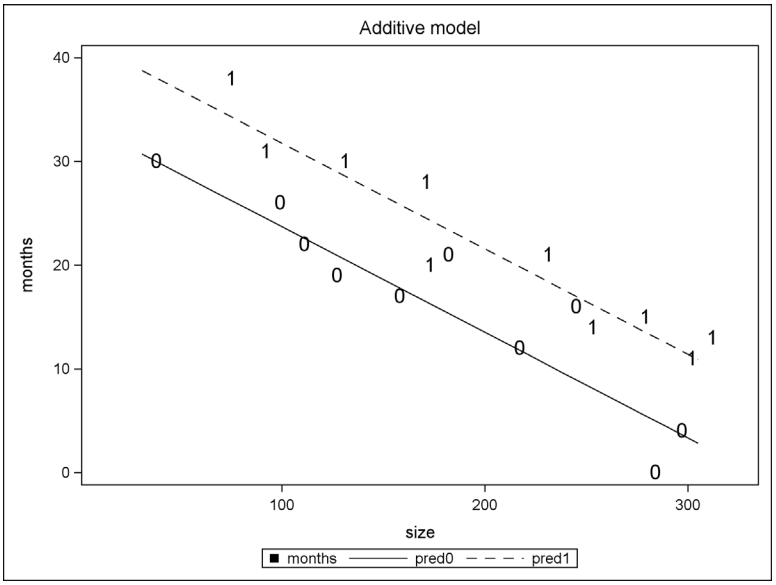
```
/* Additive model */
proc reg data=insurance;
  model months = size type;
  title1 'Additive model';
run;
```

Additive model

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	33.87407	1.81386	18.68	<.0001
size	1	-0.10174	0.00889	-11.44	<.0001
type	1	8.05547	1.45911	5.52	<.0001

```
/* Define predicted values for each type level, by hand,
   and look at fitted lines */
data insurance; set insurance;
  pred0 = 33.87407 - .10174*size;
  pred1 = 33.87407 - .10174*size + 8.05547;
proc sort data=insurance;
  by size type;
proc sgplot data=insurance;
  scatter x=size y=months /
    markerchar=type markercharattrs=(size=12pt);
  series x=size y=pred0 / lineattrs=(pattern=solid);
  series x=size y=pred1 / lineattrs=(pattern=dash);
run;
```



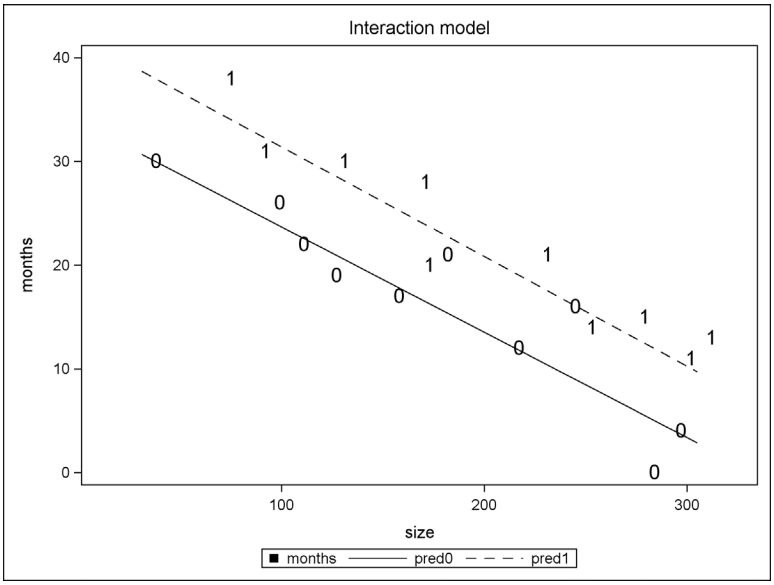
```
/* Interaction model */
data insurance; set insurance;
  size_type = size*type;
proc reg data=insurance;
  model months = size type size_type;
  title1 'Interaction model';
run;
```

Interaction model

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	33.83837	2.44065	13.86	<.0001
size	1	-0.10153	0.01305	-7.78	<.0001
type	1	8.13125	3.65405	2.23	0.0408
size_type	1	-0.00041714	0.01833	-0.02	0.9821

```
data insurance; set insurance;
  pred0 = 33.83837 - .10153*size;
  pred1 = 33.83837 - .10153*size + 8.13125 - .0041714*size;
proc sort data=insurance;  by size type;
proc sgplot data=insurance;
  scatter x=size y=months /
    markerchar=type markercharattrs=(size=12pt);
  series x=size y=pred0 / lineattrs=(pattern=solid);
  series x=size y=pred1 / lineattrs=(pattern=dash);
run;
```



3.2: Variable Selection

Dr. Bean - Stat 5100

1 Why Variable Selection

- Up until now, we have focused on trying to make predictions/inference using all the potential explanatory variables we have available to us.
- We now wish to consider several candidate models, ultimately making a judgment as to which model is "best."
  - Selection is more than an art than it is a science: no "right" decisions, several *wrong* decisions, several "reasonables."
  - This is an iterative process, that makes it difficult to know when we are "done" (see Figure 1 on last page).
- One element of the model building process involves **selecting a subset** of potential explanatory variables for use in the final model.
  - Follows the Ockham's razor principle: *entia non sunt multiplicanda praeter necessitatem*

"Entities should not be multiplied without necessity" (i.e. all else equal: simpler answers are better).

2 Methods of Variable Selection

How to pick the "best" subset of variables?

- Whenever possible, remove variables based on **context**, which comes with **expertise**.
- Automatic Methods:
  - **All possible regressions:** Consider all possible combinations of predictor variables, select the "best" model according to some measurement criteria.
  - **Stepwise methods:** Take a structured approach that takes a (semi) intelligent search through a subset of all possible models.
  - **Penalized regression:** more in Module 4.

2.1 All Possible Regressions

Consider all subsets of predictor variables $X_1, \ldots, X_{p-1}$.

- Number of subsets of size $p-1 = \binom{P-1}{p-1} = \frac{(P-1)!}{(p-1)!(P-p)!}$.
- Number of subsets of all possible sizes: $\sum_{p=1}^{P} \binom{P-1}{p-1} = 2^{P-1}$.

2.1.1 Measures of "goodness"

- R-square - but which model will always have the highest R^2?

$$R_p^2 = 1 - \frac{SS_{Error,p}}{SS_{Total}}$$

- Adjusted R-square - balances against # of predictors

$$R_{a,p}^2 = 1 - \frac{n-1}{n-p} \cdot \frac{SS_{Error,p}}{SS_{Total}}$$

  As p increases, $R_{a,p}^2$ first increases, then decreases

- Mallow's C_p - for a certain subset of $p-1$ predictors:

$$C_p = \frac{SS_{Error} \text{ from model with } p-1 \text{ predictors}}{MSE \text{ from model with } P-1 \text{ predictors}} + 2p - n$$

  When a subset of $p-1$ predictors gives unbiased $\hat{Y}$'s, $E[C_p] \approx p$.
  – so look for model with smallest p such that $C_p \approx p$,
  i.e., want $C_p \approx$ # predictors + 1.

- Akaike's information criteria & Schwarz's Bayesian criterion
  – both penalize larger numbers of predictors (want small):

$$\begin{aligned} AIC_p &= n \log SS_{Error,p} - n \log n + 2p \\ SBC_p &= n \log SS_{Error,p} - n \log n + p \log n \end{aligned}$$

- Prediction sum of squares – based on leave-one-out philosophy ($\hat{Y}_{i(i)}$)

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

  – look for models with small $PRESS_p$

2.2 Stepwise Selection

Stepwise methods:

- automatically select a model based on some criterion (convenient)
- less satisfactory, do not "guarantee" the "right" model
- best used as "confirmatory" approaches
- three main: backward (okay), forward (worst), stepwise (hybrid)

Backward Elimination – basic algorithm

1. Fit model with all $P-1$ predictors
   (a) Compare each predictor's individual P-value to some threshold (`slstay`; default in SAS is 0.10)

(b) If any predictor's P-value > `slstay`, drop predictor with largest P-value

2. Repeat with $P - 2$ predictors
3. Continue until all predictors remaining have P-values below `slstay`

Forward Selection – basic algorithm

1. Find predictor with highest correlation with response
   (a) Regress response on this predictor
   (b) Leave predictor in model if P-value is below some threshold (`slentry`; default in SAS is 0.50)
2. Given the previously entered predictor, find the predictor with the highest partial correlation with response
   (a) Add this predictor to the model
   (b) Leave in model if P-value is below `slentry`
3. Continue until no more predictors warrant inclusion
   (P-value of "next" predictor above threshold)

Big problem here: best 2-variable model does not necessarily contain best 1-variable model (first step(s) can throw everything off)
Stepwise Selection – basic algorithm:

1. Take a "forward" step: add "best" predictor with P-value below `slentry` (default 0.15)
2. Take a "backward" step: evaluate all predictors in model and drop the variable with the highest P-value above `slstay` (default 0.15)
3. Iterate "forward" and "backward" steps until model stays the same

Note: in all these automatic stepwise procedures (backward, forward, stepwise), the `slentry` and `slstay` thresholds are deceptive. After the first step (really a hypothesis test), they are <u>not</u> significance levels (α), but "conditional" significance levels, which are harder to interpret.

2.3 Remember this...

- In order to have reliable results, we need $n >> P$ (often 6*10 times larger).
- Each described technique measures how well your models fit the data you already have, which might not translate to new data (in production).

We get a sense of how our models perform on new data by:

- Splitting our data into "training" and "test" sets.
- Fit each model using only the training data, then use the model to predict on the test data.
- Calculate the mean square prediction error:

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

FIGURE 9.1
Strategy for Building a Regression Model.

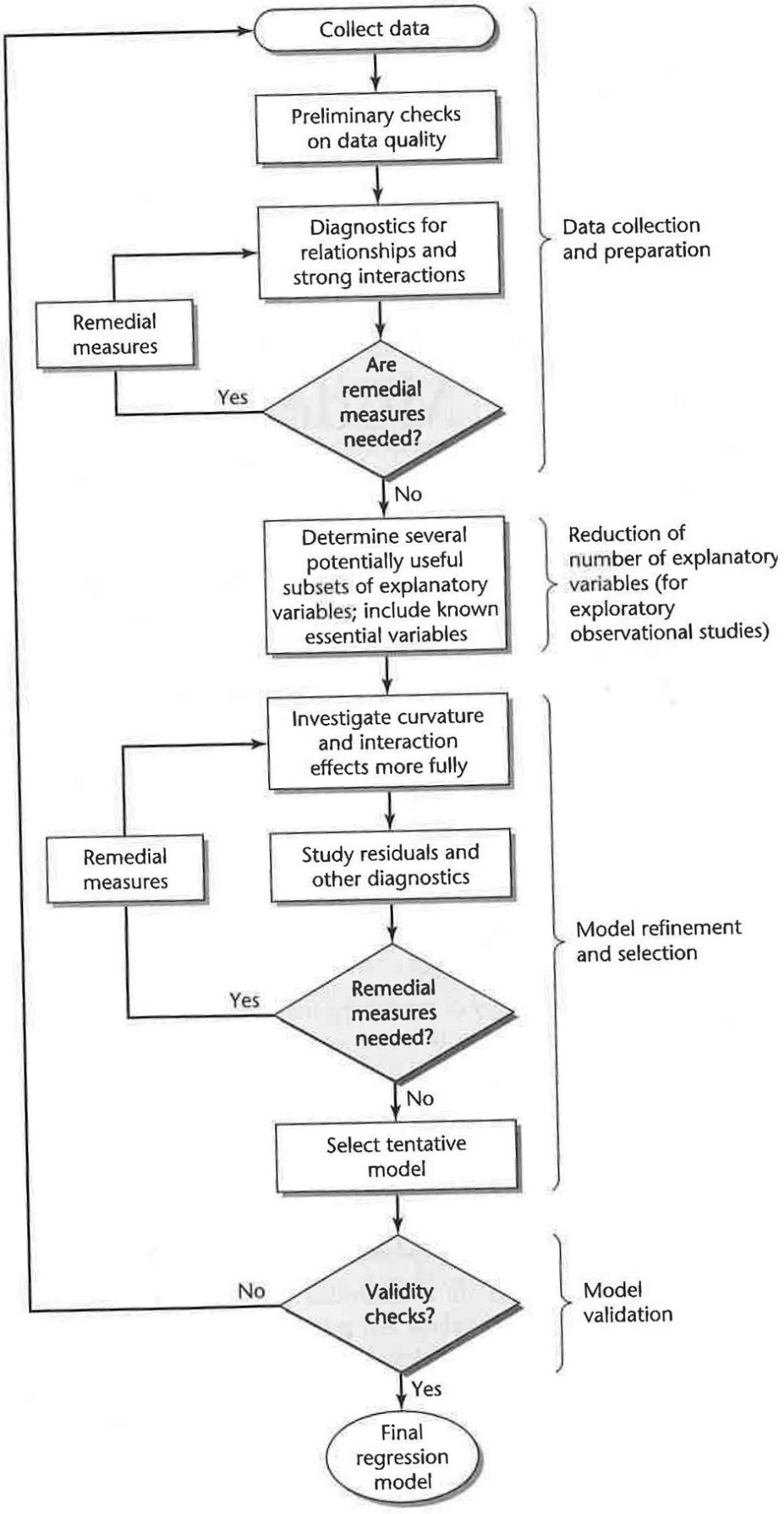


Figure 1: General model for multiple regression model selection (taken from Kutner et. al. (2004)).

Stat 5100 Handout 3.2.1 – SAS: Variable Selection

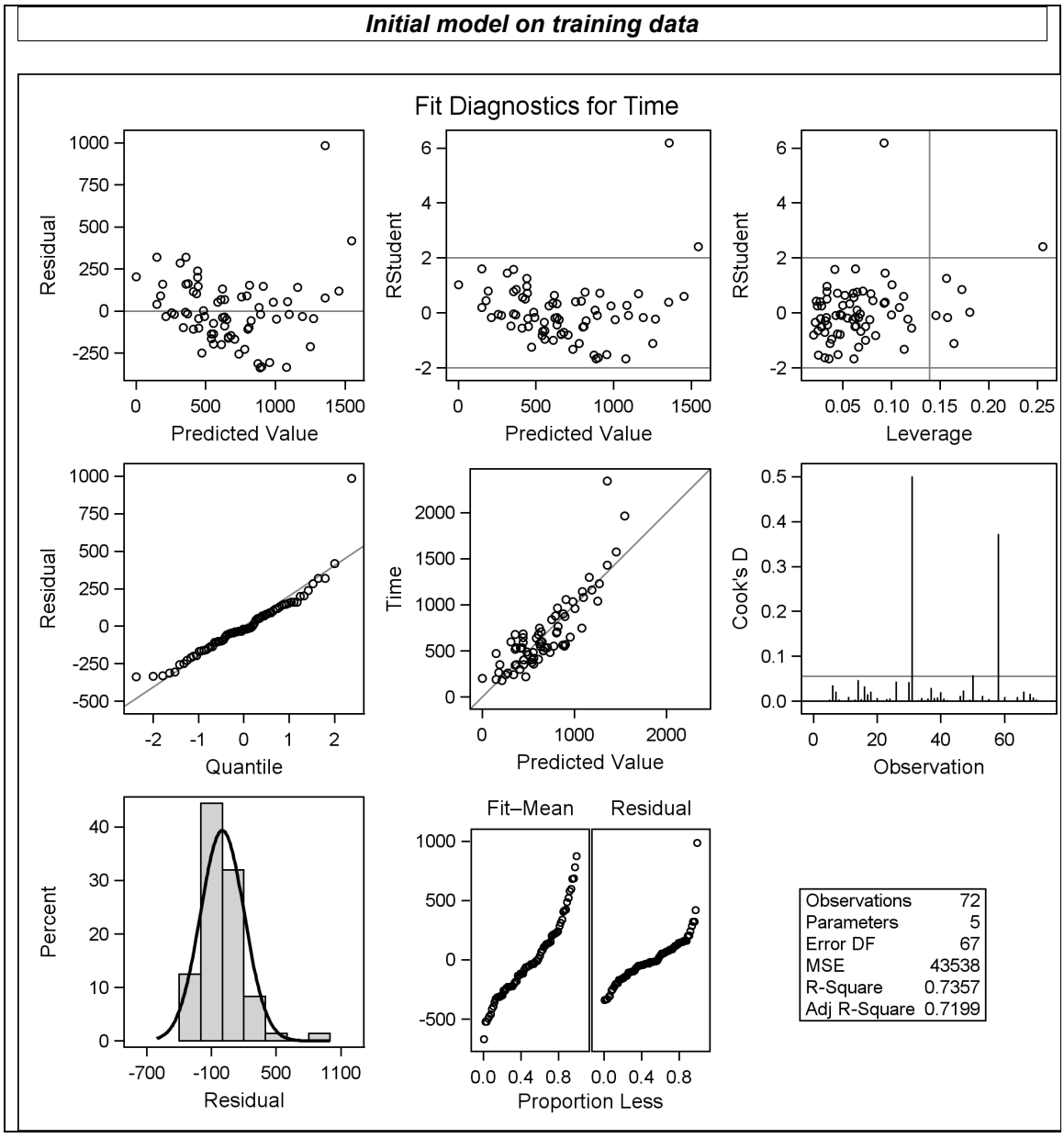
Example: (Textbook tables 9.1 & 9.5) A hospital surgical unit was interested in predicting survival time for patients who undergo a particular liver operation. Data are reported for 108 patients on the following variables: blood-clotting score, prognostic index, enzyme function test score, liver function test score, age (in years), gender (0=male, 1=female), indicators of alcohol use (none, moderate, heavy), and survival time (in days). Which (if any) of these predictors should be used in a linear model?

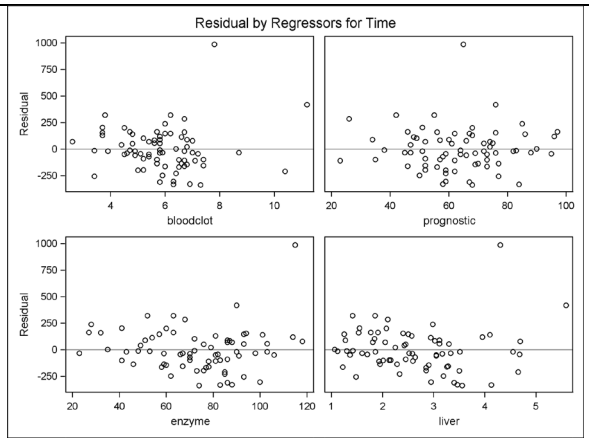
```
/* Input data -- see Table 9.1 in text */
data surgical;
  infile '<filename>' delimiter = '09'x;
       /* '09'x indicates tab-delimited .txt file */
  input bloodclot prognostic enzyme liver age gender
        modAlcohol heavyAlcohol Time;
run;

/* Randomly select training and test sets */
data surgical; set surgical;
  U = uniform(1234);
  ID = _n_;
proc sort data=surgical;
  by U;
proc print data=surgical;
  var U ID Time;
  title1 'Sorted Surgical Data (by U)';
run;
```

Sorted Surgical Data (by U)

Obs	U	ID	Time
1	0.00276	27	545
2	0.00722	101	1158
…			
107	0.97760	38	362
108	0.98587	84	881

```
data train; set surgical;
  if _n_ <= 72;
data test; set surgical;
  if _n_ > 72;
run;

/*************************************************************/

/* Check initial residual assumptions */
proc reg data=train;
  model Time = bloodclot prognostic enzyme liver;
  output out=out1 r=resid p=pred;
  title1 'Initial model on training data';
run;
```






```
/* Define shortcut macro, using line copied from
Canvas page
*/
%macro resid_num_diag(dataset,...
/* Call shortcut macro */
%resid_num_diag(dataset=out1, datavar=resid,
   label='Residual', predvar=pred, predlabel='Predicted');
run;
```

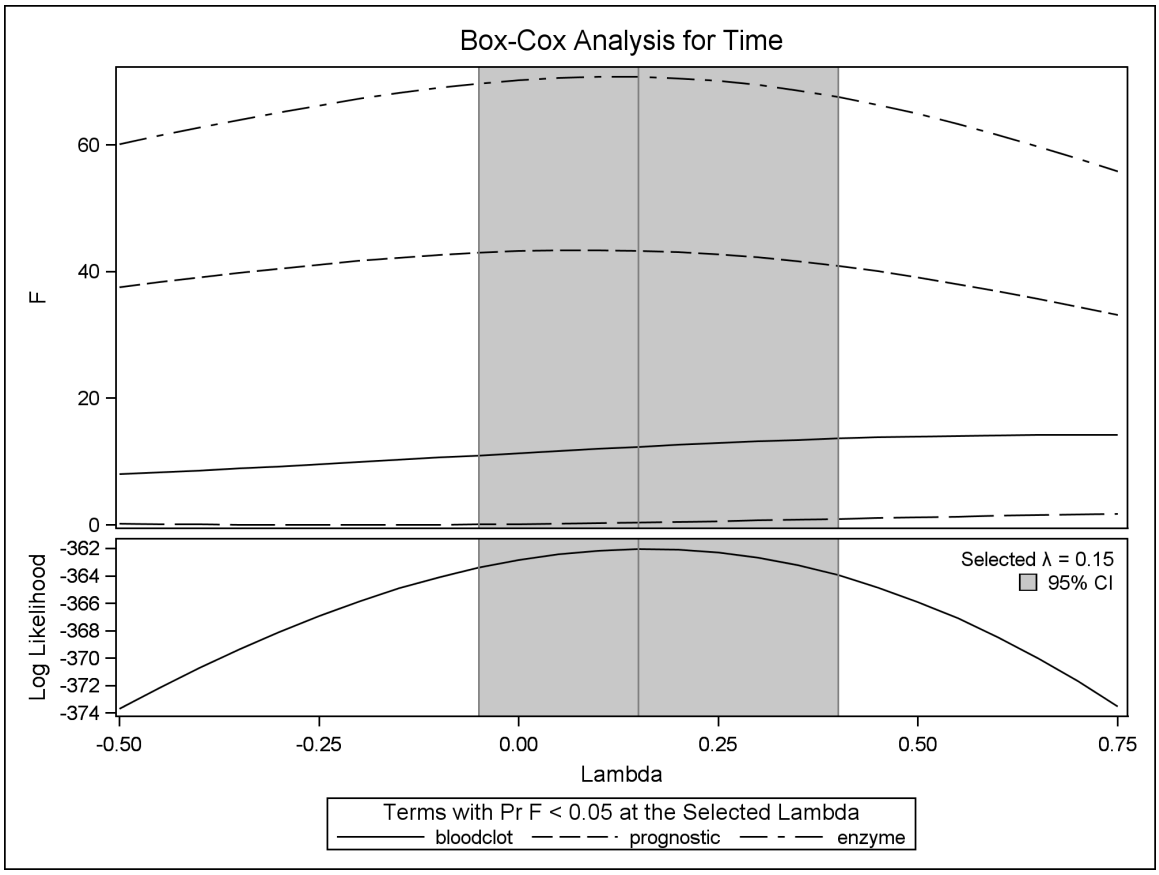
P-value for Brown-Forsythe test of constant variance in Residual vs. Predicted

Obs	t_BF	BF_pvalue
1	1.10680	0.27217

Output for correlation test of normality of Residual (Check text Table B.6 for threshold)

Pearson Correlation Coefficients, N = 72 Prob > |r| under H0: Rho=0		
	resid	**expectNorm**
resid Residual	1.00000	0.94169 <.0001
expectNorm	0.94169 <.0001	1.00000

```
/* Check possible transformation */
proc transreg data=train;
 model boxcox(Time / lambda = -.5 to .75 by .05)
       = identity(bloodclot prognostic enzyme liver);
 title1 'Box-Cox transformation on training data';
run;
```




```
/* Make transformation */
data train; set train;
  logTime = log(Time);
run;
```

```
/*************************************************************/
```

```
/* Look at some 'all possible regressions' approaches: */

proc reg data=train;
  model logTime = bloodclot prognostic enzyme liver
                  / selection=Rsquare;
  title1 'R-square Selection';
run;
```

R-square Selection

Number in Model	R-Square	Variables in Model
1	0.5474	enzyme
1	0.4175	liver
1	0.2690	prognostic
1	0.0307	bloodclot
2	0.7040	prognostic enzyme
2	0.6166	enzyme liver
2	0.5808	bloodclot enzyme
2	0.5265	prognostic liver
2	0.4249	bloodclot liver
2	0.3407	bloodclot prognostic
3	0.7688	bloodclot prognostic enzyme
3	0.7303	prognostic enzyme liver
3	0.6203	bloodclot enzyme liver
3	0.5273	bloodclot prognostic liver
4	0.7692	bloodclot prognostic enzyme liver

```
proc reg data=train;
  model logTime = bloodclot prognostic enzyme liver
                  / selection=AdjRSq;
  title1 'Adjusted R-square Selection';
run;
```

Adjusted R-square Selection

Number in Model	Adjusted R-Square	R-Square	Variables in Model
3	**0.7586**	0.7688	bloodclot prognostic enzyme
4	**0.7554**	0.7692	bloodclot prognostic enzyme liver
3	**0.7184**	0.7303	prognostic enzyme liver
2	**0.6954**	0.7040	prognostic enzyme
2	**0.6055**	0.6166	enzyme liver
3	**0.6036**	0.6203	bloodclot enzyme liver
2	**0.5686**	0.5808	bloodclot enzyme
1	**0.5409**	0.5474	enzyme
2	**0.5128**	0.5265	prognostic liver
3	**0.5064**	0.5273	bloodclot prognostic liver
1	**0.4092**	0.4175	liver
2	**0.4082**	0.4249	bloodclot liver
2	**0.3216**	0.3407	bloodclot prognostic
1	**0.2586**	0.2690	prognostic
1	**0.0168**	0.0307	bloodclot

```
proc reg data=train;
  model logTime = bloodclot prognostic enzyme liver
                  / selection=Cp;
  title1 'Mallows Cp Selection';
run;
```

Mallows Cp Selection

Number in Model	C(p)	R-Square	Variables in Model
3	**3.1274**	0.7688	bloodclot prognostic enzyme
4	**5.0000**	0.7692	bloodclot prognostic enzyme liver
3	**14.3147**	0.7303	prognostic enzyme liver
2	**19.9321**	0.7040	prognostic enzyme
2	**45.3107**	0.6166	enzyme liver
3	**46.2329**	0.6203	bloodclot enzyme liver
2	**55.7184**	0.5808	bloodclot enzyme
1	**63.4064**	0.5474	enzyme
2	**71.4633**	0.5265	prognostic liver
3	**73.2405**	0.5273	bloodclot prognostic liver
2	**100.9613**	0.4249	bloodclot liver
1	**101.1208**	0.4175	liver
2	**125.4071**	0.3407	bloodclot prognostic
1	**144.2297**	0.2690	prognostic
1	**213.4195**	0.0307	bloodclot

```
proc reg data=train;
  model logTime = bloodclot prognostic enzyme liver
                  / selection=AdjRSq Cp AIC SBC;
  title1 'Compare Selection Criteria';
run;
```

Compare Selection Criteria

Number in Model	Adjusted R-Square	R-Square	C(p)	AIC	SBC	Variables in Model
3	**0.7586**	0.7688	3.1274	-187.9550	-178.84833	bloodclot prognostic enzyme
4	**0.7554**	0.7692	5.0000	-186.0918	-174.70842	bloodclot prognostic enzyme liver
3	**0.7184**	0.7303	14.3147	-176.8567	-167.75005	prognostic enzyme liver
2	**0.6954**	0.7040	19.9321	-172.1735	-165.34349	prognostic enzyme
2	**0.6055**	0.6166	45.3107	-153.5422	-146.71221	enzyme liver
3	**0.6036**	0.6203	46.2329	-152.2428	-143.13611	bloodclot enzyme liver
2	**0.5686**	0.5808	55.7184	-147.1065	-140.27651	bloodclot enzyme
1	**0.5409**	0.5474	63.4064	-143.5924	-139.03909	enzyme
2	**0.5128**	0.5265	71.4633	-138.3479	-131.51793	prognostic liver
3	**0.5064**	0.5273	73.2405	-136.4647	-127.35805	bloodclot prognostic liver
1	**0.4092**	0.4175	101.1208	-125.4255	-120.87213	liver
2	**0.4082**	0.4249	100.9613	-124.3507	-117.52075	bloodclot liver
2	**0.3216**	0.3407	125.4071	-114.5126	-107.68262	bloodclot prognostic
1	**0.2586**	0.2690	144.2297	-109.0774	-104.52411	prognostic
1	**0.0168**	0.0307	213.4195	-88.7607	-84.20735	bloodclot

```
/*************************************************************/
```

```
/* Now look at three stepwise approaches: */


proc reg data=train;
  model logTime = bloodclot prognostic enzyme liver
                  / selection=backward slstay=.10;
  title1 'Backward Elimination';
run;
```

Backward Elimination

All variables left in the model are significant at the 0.1000 level.

Summary of Backward Elimination

Step	Variable Removed	Number Vars In	Partial R-Square	Model R-Square	C(p)	F Value	Pr > F
1	liver	3	0.0004	0.7688	3.1274	0.13	0.7223

```
proc reg data=train;
  model logTime = bloodclot prognostic enzyme liver
                  / selection=forward slentry=.10;
  title1 'Forward Selection';
run;
```

Forward Selection

No other variable met the 0.1000 significance level for entry into the model.

Summary of Forward Selection

Step	Variable Entered	Number Vars In	Partial R-Square	Model R-Square	C(p)	F Value	Pr > F
1	enzyme	1	0.5474	0.5474	63.4064	84.66	<.0001
2	prognostic	2	0.1566	0.7040	19.9321	36.51	<.0001
3	bloodclot	3	0.0648	0.7688	3.1274	19.05	<.0001

```
proc reg data=train;
  model logTime = bloodclot prognostic enzyme liver
              / selection=stepwise slentry=.10 slstay=.10;
  title1 'Stepwise Selection';
run;
```

Stepwise Selection

All variables left in the model are significant at the 0.1000 level.

No other variable met the 0.1000 significance level for entry into the model.

Summary of Stepwise Selection

Step	Variable Entered	Variable Removed	Number Vars In	Partial R-Square	Model R-Square	C(p)	F Value	Pr > F
1	enzyme		1	0.5474	0.5474	63.4064	84.66	<.0001
2	prognostic		2	0.1566	0.7040	19.9321	36.51	<.0001
3	bloodclot		3	0.0648	0.7688	3.1274	19.05	<.0001

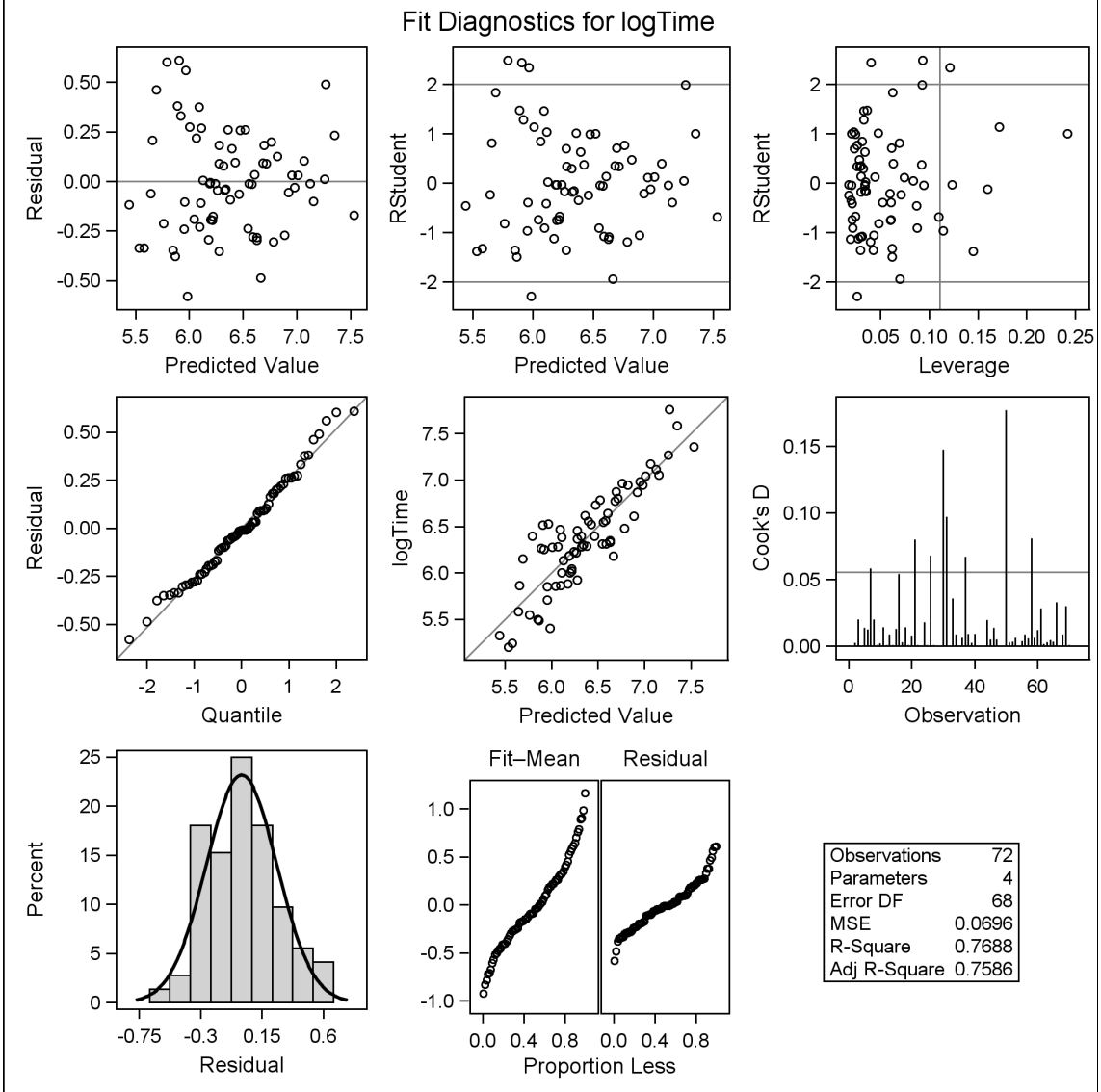
```
/*************************************************************/

/* Validity check of tentative model */

proc reg data=train;
  model logTime = bloodclot prognostic enzyme;
  output out=out2 r=resid p=pred;
  title1 'Tentative Model';
run;
```

Tentative Model

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	15.74523	5.24841	75.37	<.0001
Error	68	4.73541	0.06964		
Corrected Total	71	20.48065			

Parameter Estimates					
Variable	**DF**	**Parameter Estimate**	**Standard Error**	**t Value**	**Pr > |t|**
Intercept	**1**	3.62880	0.21572	16.82	<.0001
bloodclot	**1**	0.09656	0.02212	4.36	<.0001
prognostic	**1**	0.01523	0.00205	7.44	<.0001
enzyme	**1**	0.01649	0.00147	11.22	<.0001



```
%resid_num_diag(dataset=out2, 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	2.39814	0.019148

Output for correlation test of normality of Residual (Check text Table B.6 for threshold)

Pearson Correlation Coefficients, N = 72 Prob > |r| under H0: Rho=0		
	resid	expectNorm
resid Residual	1.00000	0.99273 <.0001
expectNorm	0.99273 <.0001	1.00000

```
data test; set test;
  logTime = log(Time);
  logTimehat = 3.62880 + 0.09656*bloodclot
               + 0.01523*prognostic + 0.01649*enzyme;
  SqPredError = (logTime - LogTimehat)**2;
proc means data=test mean;
  var SqPredError;
  title1 'MSPR for test set';
run;
```

MSPR for test set

Mean
0.0763624

3.3: Influential Observations and Outliers

Dr. Bean - Stat 5100

1 Why Care About Influential Observations/Outliers?

When we specify a model form of

$$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_{p-1} X_{p-1} + \varepsilon$$

we assume that all observations in the data are generated from the same source (i.e. the theoretical line).

If we have observations that are **not** from the same source as the rest, OLS regression will try to **force** the model to fit the data, perhaps compromising the estimated coefficients and or inference.

Two things to watch for (not mutually exclusive):

- **Outliers** - observations with values of Y that are not well-explained by the model.
- **Influential Points** - observations that unduly influence the estimated coefficients b_k or predicted values $\hat{Y}$.

Is it possible for a model outlier to not be reflected in a boxplot of Y? Explain why or why not.

Yes. A value of why can be exceptionally far away from the line *given its X-values*, while still being in a reasonable range for Y overall.

2 Ways to detect outliers or influential points

- (Primary) Scatterplots of X_k vs Y
- Other Diagnostics for Influential Observations
  - Hat matrix diagonals
  - DFBETAS
  - DFFITS
  - Cooks Distance
- Other Diagnostics for Outliers
  - Residuals
  - Studentized Residuals
  - Studentized Deleted Residuals

2.1 Hat Matrix Diagonals

Recall the linear algebra representation of the OLS regression model:

$$Y = X\beta + \varepsilon \qquad b = (X'X)^{-1}X'Y$$

$$\hat{Y} = Xb = X(X'X)^{-1}X'Y = HY$$

In other words, the predicted values of Y are simply linear combinations of the actual values of Y where each observation "weight" is determined by the X matrix.
Let $h_{i,l}$ be the element in row i and column l of H
- sometimes called "leverage" (influence of obs. i on its fitted value)

Since $\hat{Y} = HY$, then $\hat{Y}_i = \sum_{l=1}^{n} h_{i,l}Y_l$

(Individual) What would a "larger" diagonal element $h_{i,i}$ mean?

It means that the value of Y_i is more influential in its own prediction ($\hat{Y}_i$). We care about this because if the influence of a particular point is large enough, then the model is likely fitting that particular point at the sacrifice of the rest of the data.

We usually consider a point to be influential if:

- rule of thumb: $h_{i,i} > \frac{2p}{n}$ or $h_{i,i} > \frac{3p}{n}$
- can plot $h_{i,i}$ against observation number, with reference lines at $2p/n$ (SAS default) and/or $3p/n$

Another graphical diagnostic with $h_{i,i}$:

- leverage plots/partial regression/added variable plots); for X_1:
  1. Regress X_1 on $X_2, \ldots, X_{p-1}$ and obtain residuals $e_{X_1|X_2,\ldots,X_{p-1}}$
  2. Regress Y on $X_2, \ldots, X_{p-1}$ and obtain residuals $e_{Y|X_2,\ldots,X_{p-1}}$
  3. Plot $e_{Y|X_2,\ldots,X_{p-1}}$ vs. $e_{X_1|X_2,\ldots,X_{p-1}}$, and add regression line
     - slope will be b_1 from multiple regression model
     - Helps to visualize the marginal effect of adding X_1 in the model after already including all other X variables.
     - Influential points fall significantly farther away from the line than other points.
- (possible) modification here: point-size in leverage plot proportional to corresponding $h_{i,i}$ NOT shown in the SAS output provided in HO 3.3.1.
  - then this is called a proportional leverage plot
  - influential observations will be the points with big "bubbles" that appear to "pull" the regression line in their direction

2.2 DFBETAS

Provide a measure of how **different** ("DF") an estimate of β_k would be if we removed one observation from the data.

$$\begin{aligned} b_k &= \text{estimate of } \beta_k \text{ using full data} \\ b_{k(i)} &= \text{estimate of } \beta_k \text{ when observation } i \text{ is ignored} \\ MSE_{(i)} &= \text{Mean SS for error when observation } i \text{ is ignored} \\ C_{kk} &= k^{th} \text{ diagonal element of } (X'X)^{-1} \\ DFBETAS_{k(i)} &= \frac{b_k - b_{k(i)}}{\sqrt{MSE_{(i)} C_{kk}}} \end{aligned}$$

Interpreting DFBETAS:

- DFBETAS$_{k(i)}$ positive: obs. i "pulls" b_k up
- DFBETAS$_{k(i)}$ negative: obs. i "pulls" b_k down

How "large" to declare observation i "influential" on b_k?

- *Rough* rule of thumb:

  $|DFBETAS_{k(i)}| > 1$ for $n \leq 30$

  $|DFBETAS_{k(i)}| > 2/\sqrt{n}$ for $n > 30$ (SAS)

- Graphical diagnostics probably better for DFBETAS:
  - Histograms or boxplots for each k
  - Proportional leverage plot with "bubble" size prop. to DFBETAS$_{k(i)}$
  - Plot DFBETAS$_{k(i)}$ against obs. number for each k (Provided by SAS, unlike the others)

2.3 DFFITS

Similar to DFBETAS: how different would $\hat{Y}_i$ be
if observation i were not used to fit the model

$$DFFITS_i = \frac{\hat{Y}_i - \hat{Y}_{i(i)}}{\sqrt{MSE_{(i)} h_{i,i}}}$$

How large DFFITS to declare obs. i as influential on $\hat{Y}_i$?

- *Rough* rule of thumb:

  $|DFFITS_i| > 1$ for $n \leq 30$

  $|DFFITS_i| > 2\sqrt{\frac{p}{n}}$ for $n > 30$ (SAS)

- Good graphical diagnostics for DFFITS:
  - Plot DFFITS vs. Observation Number

- Plot Residuals vs. Predicted Values, with point sizes proportional to corresponding DFFITS_i

(DFBETAS_{ij} vs. DFFITS_i) vs. $h_{i,i}$

- somewhat related, so "conclusions" will quite often agree
- BUT: if two or more points exert "influence" together then the drop-one diagnostics (DFBETAS and DFFITS) may not detect them
  - these are leverage points - need to look at $h_{i,i}$

2.4 Cooks Distance

Kind of an overall measure of effect of obs. i on all of the $\hat{Y}_l$ values:

$$D_i = \frac{\sum_{j=1}^{n} \left(\hat{Y}_j - \hat{Y}_{j(i)}\right)^2}{p \cdot \text{MSE}}$$

Diagnostics:

- Numerical:
  - simple: compare D_i with $4/n$ (SAS)
  - more useful: compare D_i with the $F_{p,n-p}$ distribution (See 3.3.1 pg 8 for example of how to do this "by hand")
    - percentile 10-20: little influence
    - percentile 50+: major influence
- Graphical: plot D_i (or percentile from $F_{p,n-p}$) vs. observation number i

2.5 Residuals

$e_i = Y_i - \hat{Y}_i$

Sometimes a large $|e_i|$ indicates an outlier

- not well-explained by fitted model
- but how "large" it needs to be depends on the residuals:
  - Recall $\varepsilon \sim N(0, \sigma^2)$, so $e_i \sim N(0, \sigma^2(1 - h_{ii}))$
    - because $\hat{Y} = HY$ results in $e = Y - HY = (I - H)Y$
  - Could compare e_i with the normal critical values, but need to estimate variance (including σ^2) $\Rightarrow$ normal approx. not appropriate; need Student's t

2.6 Studentized Residuals

$$r_i = \frac{e_i}{\sqrt{MSE \cdot (1 - h_{ii})}} \qquad (MSE = \hat{\sigma}^2)$$

If ε_i iid $N(0, \sigma^2)$, then the r_i follow the t_{n-p} distribution; diagnostics:

- Numerical: compare $|r_i|$ with upper $\alpha/2$ critical value of t_{n-p}
- Graphical: plot $\hat{Y}_i$ vs. r_i, with ref. lines at upper $\alpha/2$ critical value of t_{n-p}

2.7 Studentized Deleted Residuals

If obs. i really is an outlier, then including it in the data will inflate MSE
- So consider dropping it and re-calculating the studentized residual:

$$e_i^* = \frac{e_i}{\sqrt{MSE_{(i)}(1 - h_{ii})}} \qquad \text{(Text uses } t_i \text{ instead of } e_i^*)$$

2.8 Other Diagnostics (similar to studentized residuals)

- plot $\hat{Y}_i$ vs. e_i^*
- compare to $|e_i^*|$ to some critical value of t_{n-p} (for each of $i = 1, \ldots, n$)

  BUT: α = probability of type I error (calling obs. i outlier when it's not)

  - actually want α to be probability of *at least one* type I error in all n tests – a family-wise error rate
  - many ways to adjust the critical value; here, we'll use Bonferroni correction:

    compare $|e_i^*|$ to upper $\alpha/(2n)$ critical value of t_{n-p}

3 Remedial Measures for Influential Observations or Outliers

1. Look for:
   - typos in data (more common than would like to think)
   - fundamental differences in observations
     - drop obs. if from a different "population"
   - very skewed distributions of predictors
     - remember that in general, there is no assumption regarding the distribution of X's
     - sometimes transforming X will reduce influence of obs. with extreme values
2. Look at potential changes to model:
   - will a transformation "bring in" the observations?
   - should a curvilinear or other predictor be added?
     - look at leverage plot for the possible predictor

- any trend suggests adding it to model

3. Could obtain estimates differently (instead of OLS, robust regression - more in Module 4):
    - LAD (least absolute deviation) regression
    - IRLS (iteratively reweighted least squares) regression

Stat 5100 Handout 3.3.1 – SAS: Influential Observations and Outliers

Example: Data collected on 50 countries relevant to a cross-sectional study of a life-cycle savings hypothesis, which states that the response variable

- SavRatio: aggregate personal saving divided by disposable income

can be explained by the following four predictor variables:

- AvIncome: per-capita disposable income, in USD (yearly average over decade)
- GrowRate: percentage growth rate in per-capita disposable income (over decade)
- PopU15: percentage of the population less than 15 years old (yearly average over decade)
- PopO75: percentage of the population over 75 years old (yearly average over decade)

The decade is 1960-1970. These data are published in section 2.2 of *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity* (1980) by Belsley, Kuh, and Welsch (limited excerpt available through Google books).

```
/* Define options */
ods html image_dpi=300 style=journal;

/* Read in the data */
proc import out=work.savings dbms=csv replace
    datafile =
   "<file path here>"
  getnames=yes;
  datarow=2;
run;


/* Look at a regression model to predict SavRatio,
   with diagnostics for influential obs. and outliers */

proc reg data = savings
   plots(label)=(CooksD RStudentByLeverage DFFITS DFBETAS);
 id Country;
 model SavRatio = PopU15 PopO75 AvIncome GrowRate /
            partial partialdata;
 output out=out1 r=resid p=pred;
 title1 'Predict SavRatio';
run;
```

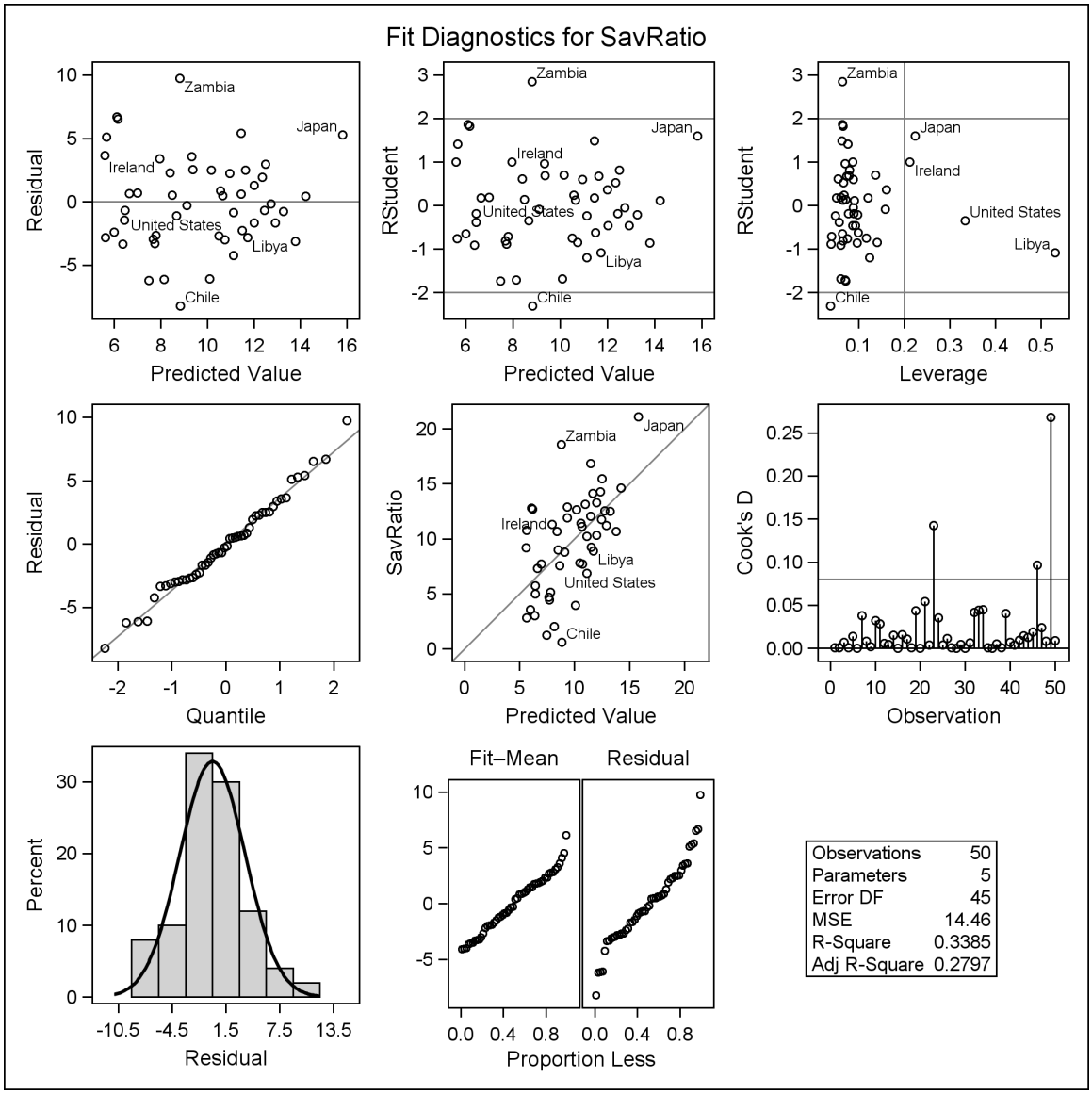
Predict SavRatio

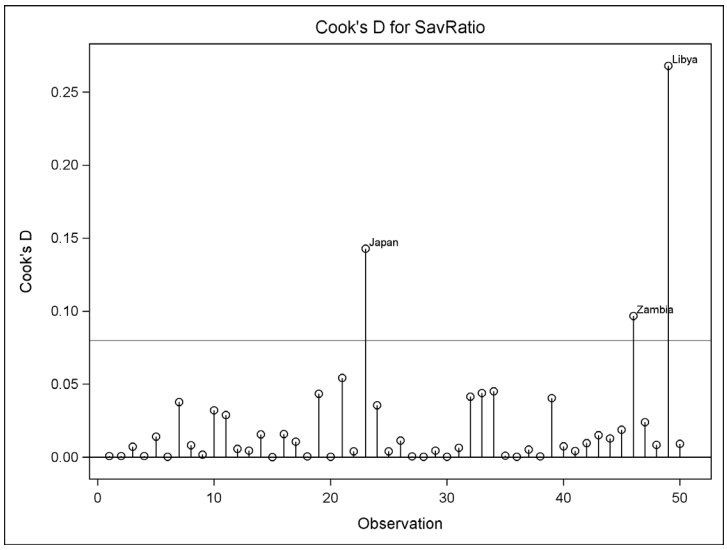
Number of Observations Read	50
Number of Observations Used	50

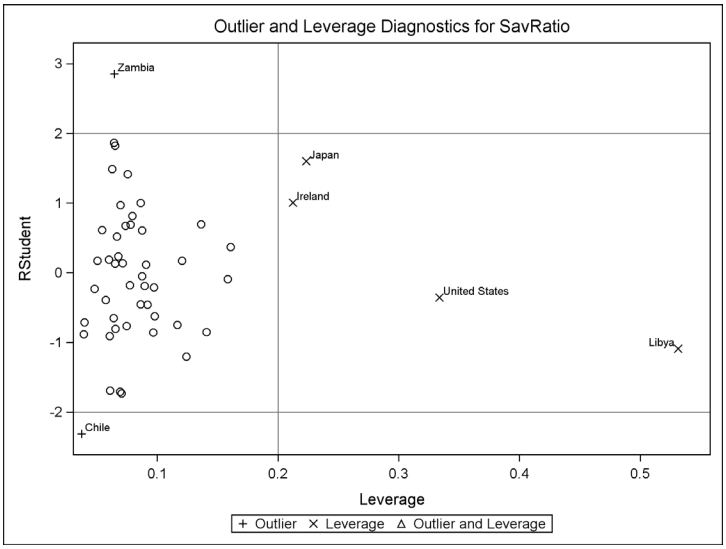
Analysis of Variance					
Source	**DF**	**Sum of Squares**	**Mean Square**	**F Value**	**Pr > F**
Model	4	332.91525	83.22881	5.76	0.0008
Error	45	650.71300	14.46029		
Corrected Total	49	983.62825			

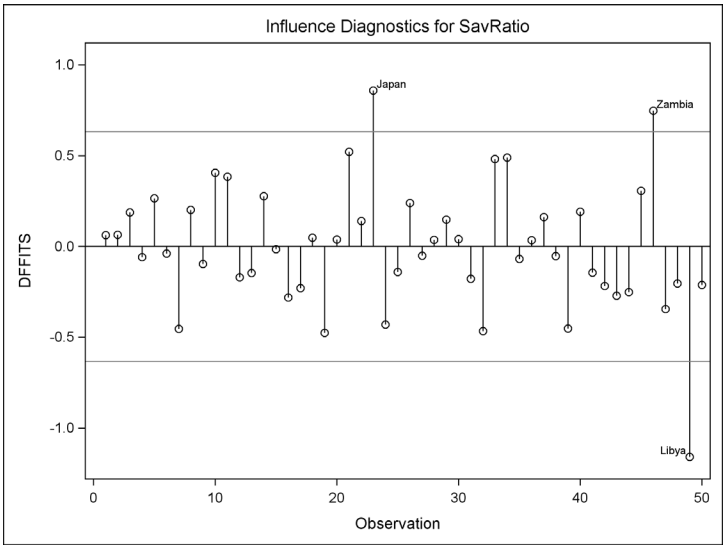
Root MSE	3.80267	**R-Square**	0.3385
Dependent Mean	9.67100	**Adj R-Sq**	0.2797
Coeff Var	39.32033		

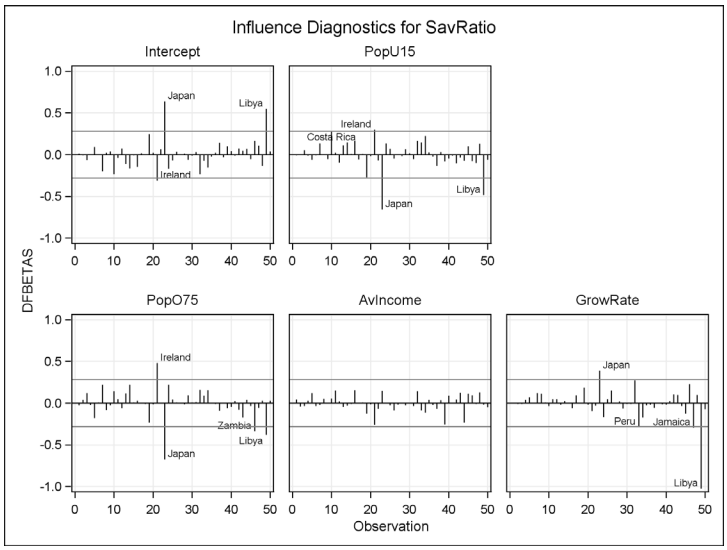
Parameter Estimates					
Variable	**DF**	**Parameter Estimate**	**Standard Error**	**t Value**	**Pr > |t|**
Intercept	**1**	28.56609	7.35452	3.88	0.0003
PopU15	**1**	-0.46119	0.14464	-3.19	0.0026
PopO75	**1**	-1.69150	1.08360	-1.56	0.1255
AvIncome	**1**	-0.00033690	0.00093111	-0.36	0.7192
GrowRate	**1**	0.40969	0.19620	2.09	0.0425

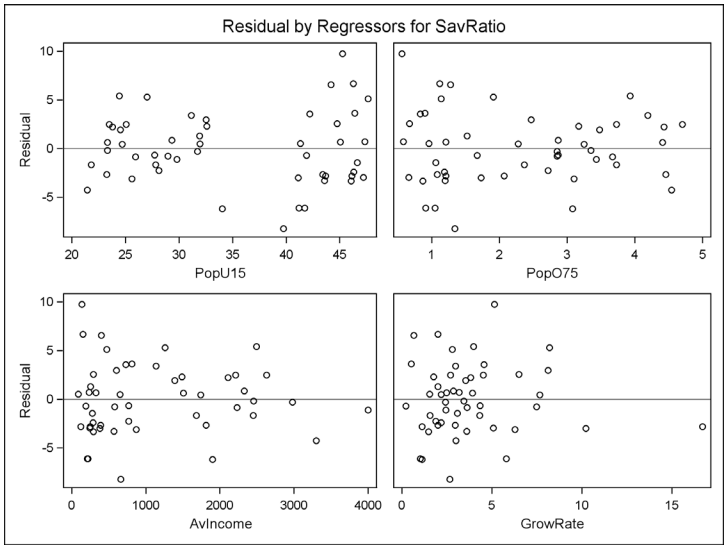

Fit Diagnostics for SavRatio
Residual
Predicted Value
Zambia
Japan
Ireland
United States
Libya
Chile
RStudent
Leverage
Quantile
SavRatio
Cook's D
Observation
Percent
Fit–Mean
Residual
Proportion Less
Observations 50
Parameters 5
Error DF 45
MSE 14.46
R-Square 0.3385
Adj R-Square 0.2797

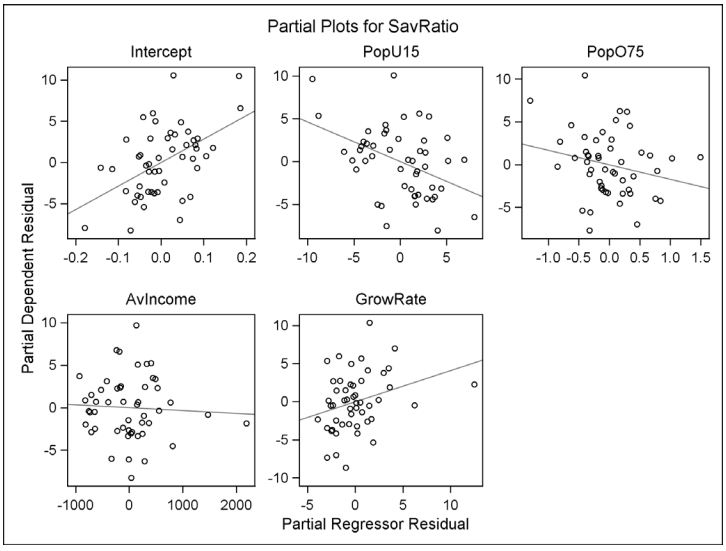

Cook's D for SavRatio
Cook's D
Observation
Libya
Japan
Zambia
0.25
0.20
0.15
0.10
0.05
0.00
0
10
20
30
40
50


Outlier and Leverage Diagnostics for SavRatio
RStudent
Leverage
Zambia
Japan
Ireland
United States
Libya
Chile
3
2
1
0
-1
-2
0.1
0.2
0.3
0.4
0.5
+ Outlier × Leverage Δ Outlier and Leverage


Influence Diagnostics for SavRatio
DFFITS
Observation
Japan
Zambia
Libya
1.0
0.5
0.0
-0.5
-1.0
0
10
20
30
40
50


Influence Diagnostics for SavRatio
Intercept
PopU15
PopO75
AvIncome
GrowRate
DFBETAS
Observation
Japan
Libya
Ireland
Costa Rica
Zambia
Peru
Jamaica


Residual by Regressors for SavRatio
Residual
PopU15
PopO75
AvIncome
GrowRate


Partial Plots for SavRatio
Intercept
PopU15
PopO75
AvIncome
GrowRate
Partial Dependent Residual
Partial Regressor Residual

Obs	Country	partial PopU15	SavRatio partial PopU15	partial PopO75	SavRatio partial PopO75	partial AvIncome	SavRatio partial AvIncome	partial GrowRate	SavRatio partial GrowRate
1	**Australia**	-1.13831	1.38856	-0.38628	1.51696	768.25943	0.60475	-0.01280	0.85833
…									
21	**Ireland**	6.87903	0.21857	1.49268	0.86626	-928.27477	3.70387	-1.59754	2.73663
22	**Italy**	-3.46588	3.52520	0.02001	1.89291	-530.90583	2.10562	-1.03136	1.50421
23	**Japan**	-9.48247	9.65473	-1.30002	7.48046	328.00826	5.17098	4.14032	6.97775
24	**Korea**	-2.03533	-5.16830	-0.44035	-5.36212	-11.96560	-6.10295	1.88266	-5.33567
…									
44	**United States**	4.41082	-3.14583	-0.30252	-0.59987	**2191.50614**	-1.84991	1.46020	-0.51335
45	**Venezuela**	2.53497	2.46341	-0.11300	3.82366	444.00821	3.48293	-2.30789	2.68698
46	**Zambia**	-0.70557	10.07632	-0.40325	10.43302	130.21746	9.70704	1.49950	10.36525
…									
49	**Libya**	7.98336	-6.51140	0.83739	-4.24597	49.71961	-2.84628	**12.47740**	2.28240
50	**Malaysia**	1.93033	-3.86112	-0.13636	-2.74022	243.12949	-3.05278	1.68312	-2.28130

```
/* Check other assumptions */
/* Define shortcut macro, using line copied from
Course Canvas Page */
%macro resid_num_diag(dataset, ...

/* Call shortcut macro */
%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 Value

Obs	t_BF	BF_pvalue
1	2.40263	0.020193

Output for correlation test of normality of Residual (Check text Table B.6 for threshold)

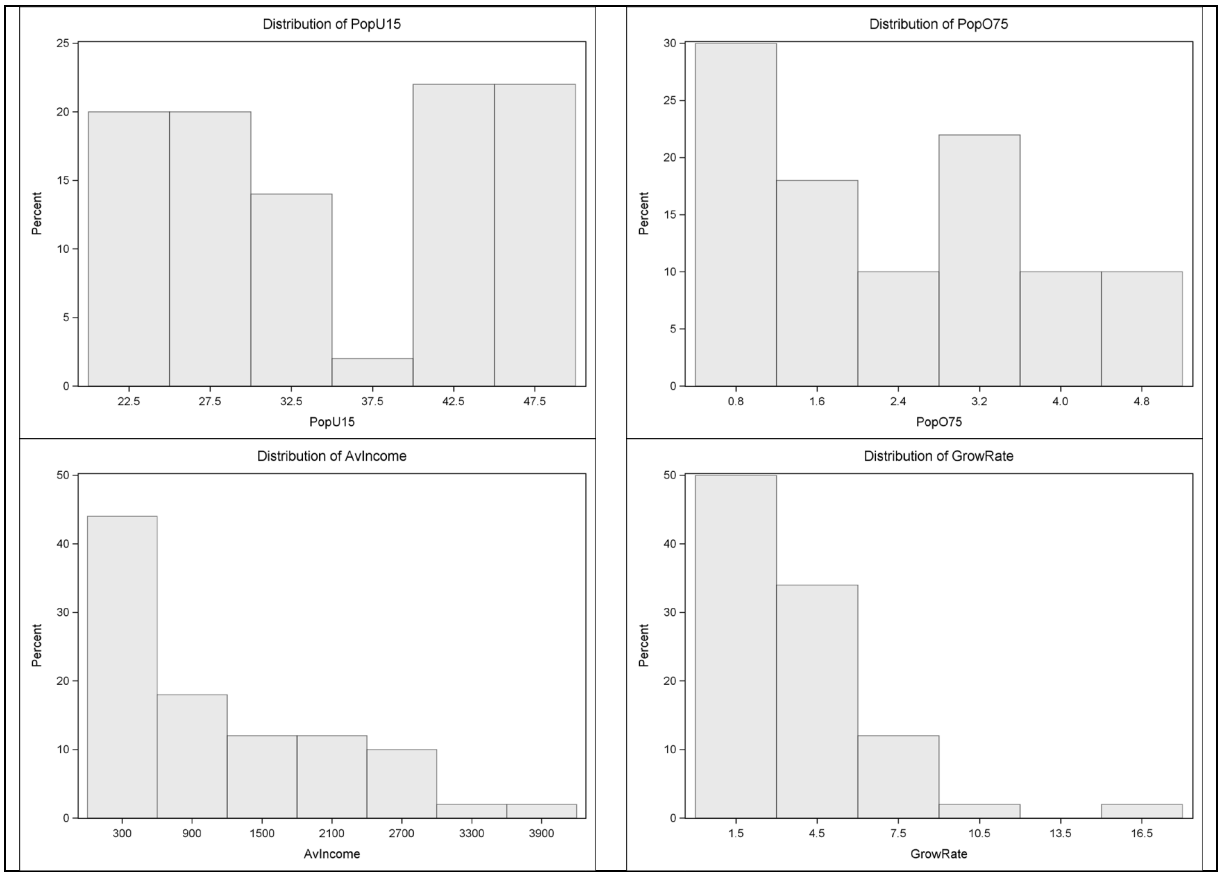
Pearson Correlation Coefficients, N = 50 Prob > |r| under H0: Rho=0		
	resid	expectNorm
resid Residual	1.00000	0.99252 <.0001
expectNorm	0.99252 <.0001	1.00000

```
/* Alternative thresholds for influential obs.
   and outlier diagnostics */
data temp;
  p=5;          /* p = # beta's (incl. intercept */
  n = 50;       /* n = sample size */
  CooksDsimple = 4/n;
  CooksD10 = finv(.10,p,n-p);
  CooksD20 = finv(.20,p,n-p);
  CooksD50 = finv(.50,p,n-p);
  RStudent95 = tinv((1-.05/2),(n-p));
  RStudent95Bonf = tinv((1-.05/2/n),(n-p));
  Leverage2 = 2*p/n;
  Leverage3 = 3*p/n;
  DFBETAS = 2/n**0.5; if (n <= 30) then DFBETAS = 1;
  DFFITS = 2*(p/n)**0.5; if (n <= 30) then DFFITS = 1;
  ;
proc print data=temp;
  var CooksDsimple CooksD10 CooksD20 CooksD50 RStudent95
      RStudent95Bonf Leverage2 Leverage3 DFBETAS DFFITS;
  title1 'Alternative thresholds';
run;
```

Alternative thresholds

Obs	CooksDsimple	CooksD10	CooksD20	CooksD50	RStudent95	RStudent95Bonf	Leverage2	Leverage3	DFBETAS	DFFITS
1	0.08	0.31729	0.46527	0.88349	2.01410	3.52025	0.2	0.3	0.28284	0.63246

```
/* Now look more closely at distribution of predictors,
   and suspect observations */
proc univariate data=savings noprint;
 var PopU15 PopO75 AvIncome GrowRate;
 histogram  PopU15 PopO75 AvIncome GrowRate;
 title1;
run;
```




```
proc print data=savings;
  where country = 'Ireland' | country = 'Japan'
     | country = 'United States' | country = 'Libya'
     | country = 'Zambia';
  var country SavRatio PopU15 PopO75 AvIncome GrowRate;
  title1 'Suspect observations';
run;
```

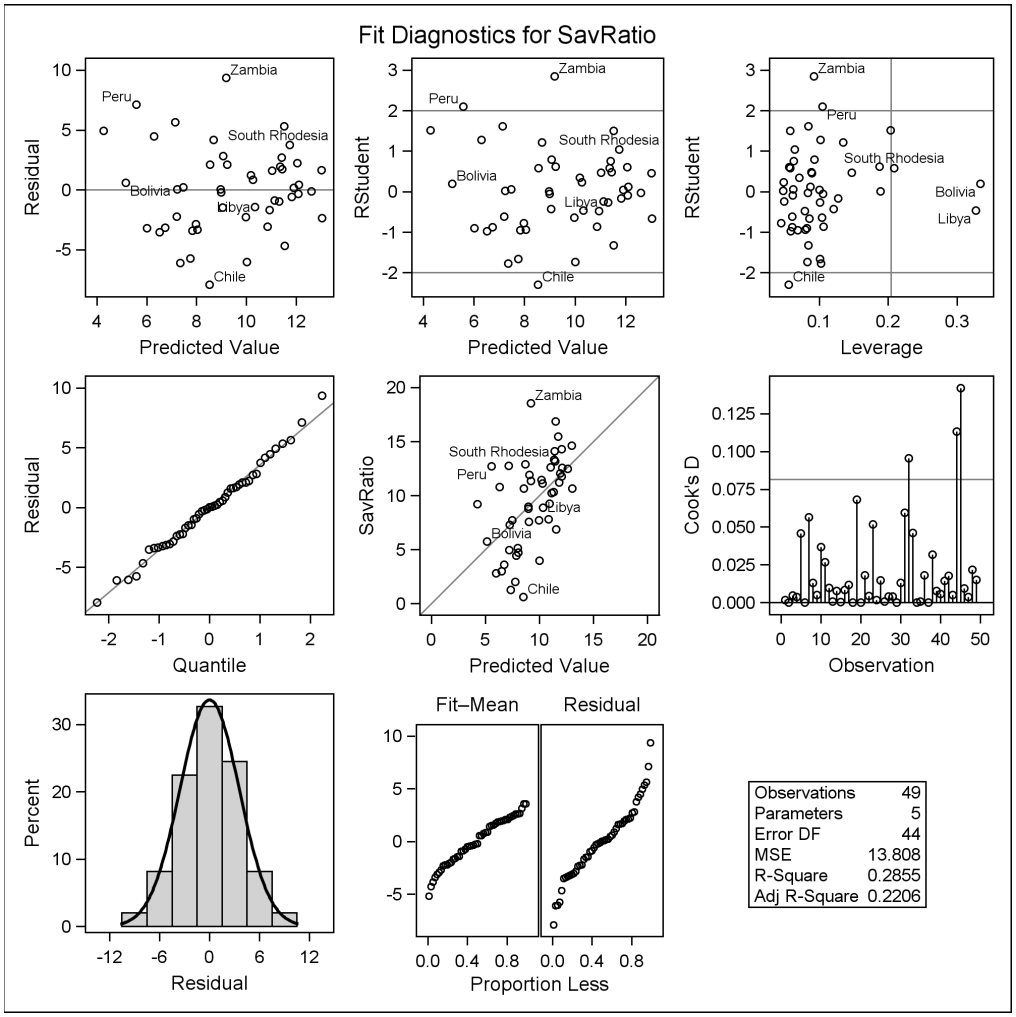
Suspect observations

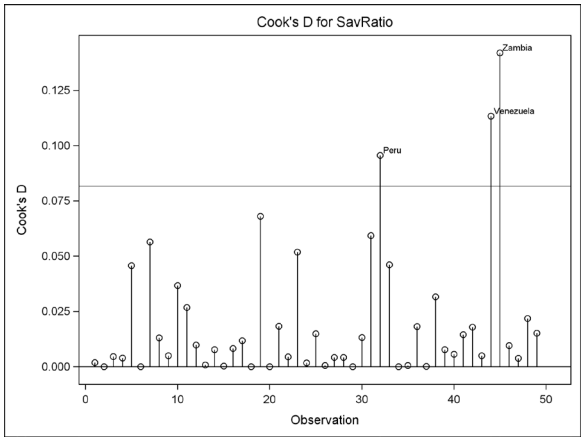
Obs	Country	SavRatio	PopU15	PopO75	AvIncome	GrowRate
21	Ireland	11.34	31.16	4.19	1139.95	2.99
23	Japan	21.1	27.01	1.91	1257.28	8.21
44	United States	7.56	29.81	3.43	4001.89	2.45
46	Zambia	18.56	45.25	0.56	138.33	5.14
49	Libya	8.89	43.69	2.07	123.58	16.71

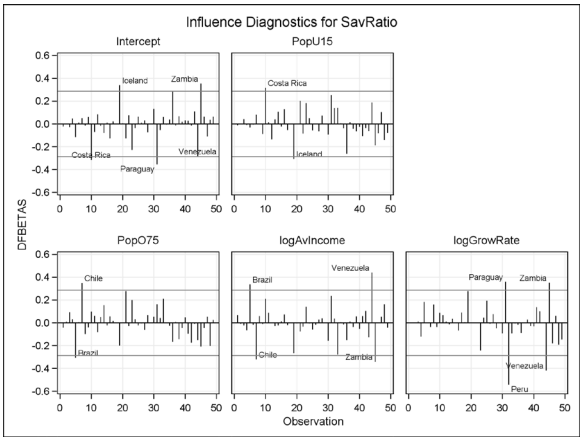
```
/*************************************************
  Possible Remedial Measures for these data:

    Drop Japan
       -- PopU15 and PopO75 don't match profile
          (influential but not outliers)

    Take log of AvIncome and log of GrowRate
       -- their distributions are skew right
       -- the extreme observation in each is a suspect obs.
            (United States for AvIncome,
             Libya for GrowRate)

 *************************************************/

/* Create new data set and fit regression model;
   check assumptions */
data newsavings; set savings;
   if country ne 'Japan';
   logAvIncome = log(AvIncome);
   logGrowRate = log(GrowRate);
run;

proc reg data = newsavings
 plots(label)=(CooksD RStudentByLeverage DFFITS DFBETAS);
 id Country;
 model SavRatio = PopU15 PopO75 logAvIncome logGrowRate
    / partial;
 output out=out2 r=resid p=pred;
 title1 'Predict SavRatio';
 title2 '(after remedial measures)';
run;
```

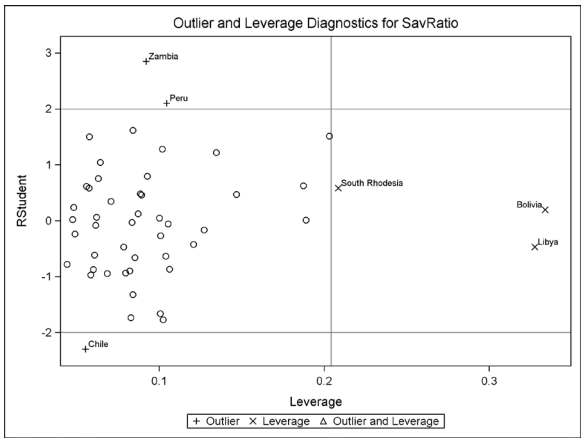
Predict SavRatio
(after remedial measures)

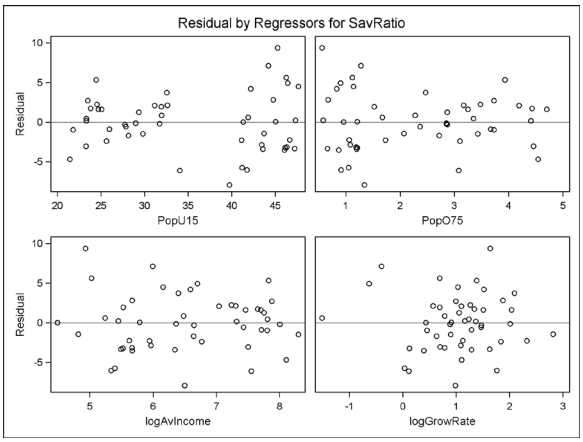
Parameter Estimates					
Variable	**DF**	**Parameter Estimate**	**Standard Error**	**t Value**	**Pr > |t|**
Intercept	**1**	26.25118	10.52632	2.49	0.0165
PopU15	**1**	-0.33837	0.15791	-2.14	0.0377
PopO75	**1**	-0.68558	1.13571	-0.60	0.5492
logAvIncome	**1**	-0.71860	0.97492	-0.74	0.4650
logGrowRate	**1**	1.33042	0.72528	1.83	0.0734

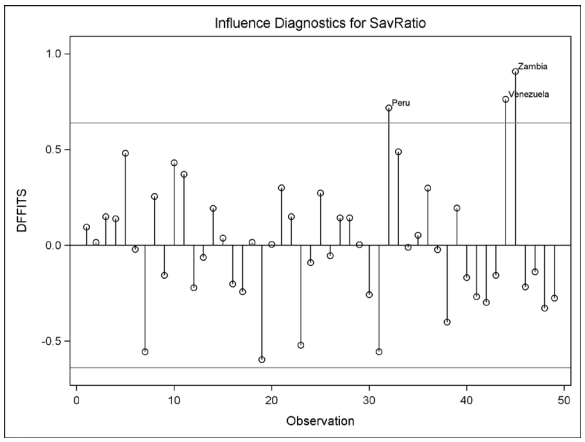


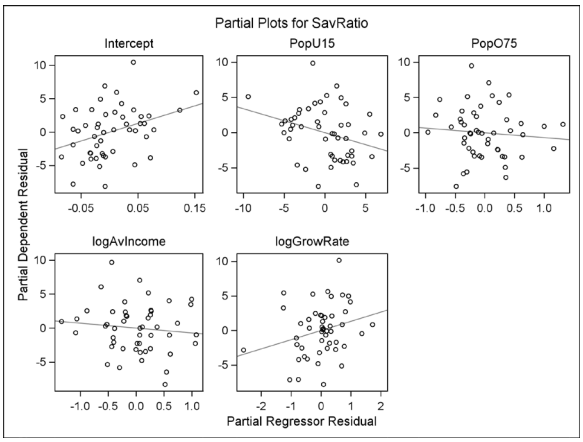













```
/* Check model assumptions */
%resid_num_diag(dataset=out2, datavar=resid,
  label='New Residual', predvar=pred,
  predlabel='New Predicted Value');
run;
```

P-value for Brown-Forsythe test of constant variance in New Residual vs. New Predicted Value

Obs	t_BF	BF_pvalue
1	2.43339	0.018815

Output for correlation test of normality of New Residual (Check text Table B.6 for threshold)

Pearson Correlation Coefficients, N = 49 Prob > |r| under H0: Rho=0		
	resid	expectNorm
resid New Residual	1.00000	0.99516 <.0001
expectNorm	0.99516 <.0001	1.00000

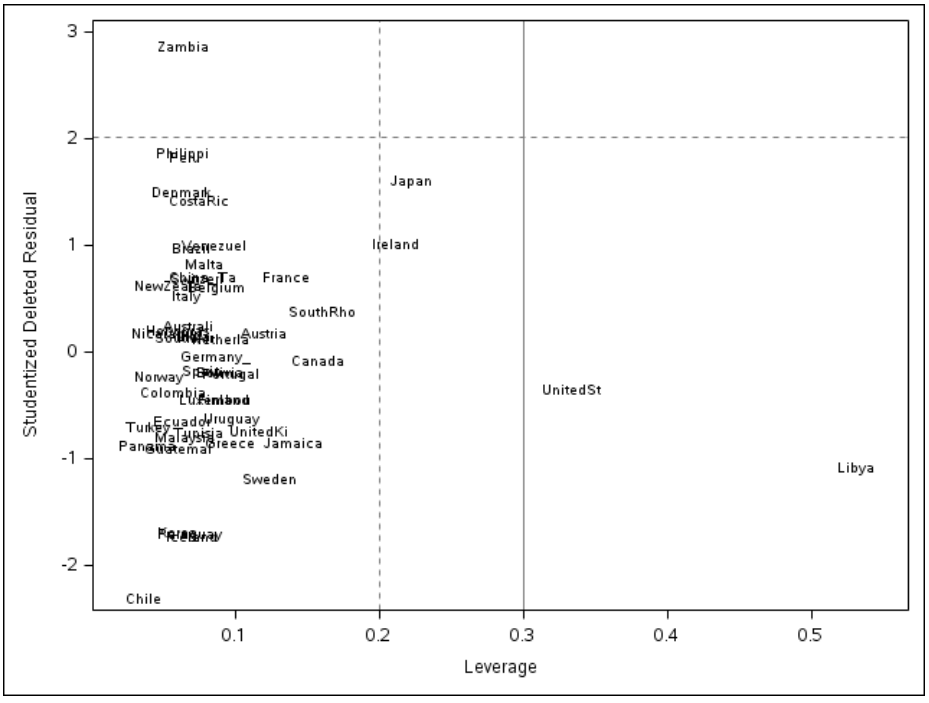
```
/* Look at final model */
proc reg data = newsavings;
     model SavRatio = PopU15 logGrowRate;
      title1 'Final Model';
run;
```

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	14.27955	2.40166	5.95	<.0001
PopU15	1	-0.18046	0.05915	-3.05	0.0038
logGrowRate	1	1.45209	0.71058	2.04	0.0468

```
/************************************************
  What if want to add your own reference lines?
 ************************************************/

proc reg data = savings ;
     id country;
     model SavRatio = PopU15 PopO75 AvIncome GrowRate /
             influence;
     ods output outputstatistics=out3;
run;

proc print data=out3;
run;
```

Observation	Country	Residual	RStudent	HatDiagonal	CovRatio	DFFITS	...	DFB_GrowRate
1	Australi	0.8636	0.2327	0.0677	1.1928	0.0627		-0.0002
2	Austria	0.6164	0.1710	0.1204	1.2678	0.0632		-0.0082
3	Belgium	2.2190	0.6066	0.0875	1.1762	0.1878		-0.0073
...								

```
proc sgplot data=out3;
 scatter x=HatDiagonal y=RStudent / markerchar=country;
 xaxis label='Leverage';
 yaxis label='Studentized Deleted Residual';
 refline 2.01 / axis=Y lineattrs=(pattern=2);
 refline 3.52 / axis=Y;
 refline .2 / axis=X lineattrs=(pattern=2);
 refline .3 / axis=X;
run;
```



3.4: Model Validation

Dr. Bean - Stat 5100

1 Why Model Validation?

Recall that there are two, distinct, goals of linear modeling and we don't always care about both at the same time:

- Inference: Is there a significant, linear relationship between X_k and Y, after accounting for the effect of a set of other X variables?
  - Example: Do students who use the tutor center see a significant positive affect to their GPA after accounting for study time and demographics?
- Prediction: Given a set of variables that are *easy* to measure, can I predict a variable that is hard to measure?
  - Use car weight (easy to measure) to predict car safety (hard to measure).

For **prediction**, there are a lot of alternatives to linear regression for which measures such as AIC, SBC, $C(p)$, and even R^2 are not relevant.

We need an *objective* way to compare the effectiveness of models with incomparable forms.

Why is the data we are using to fit our models not a fair measure of model effectiveness?

We ultimately want a model that can predict well on new data. Complex models have incentive to overfit the current data at the sacrifice of good predictions on new data.

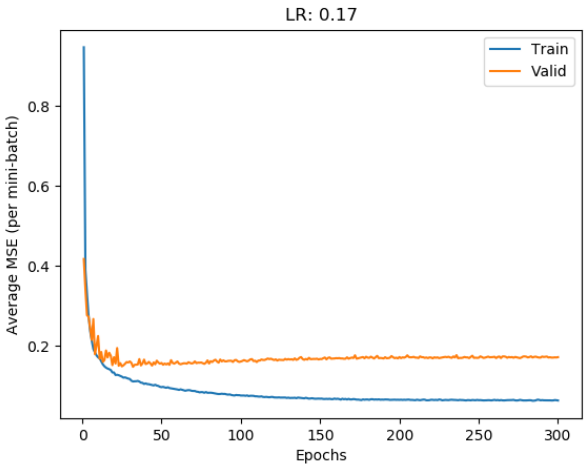


Figure 1: Comparison of accuracy on training and test sets for a neural network.

2 Validation Details

Terminology

- **Training set:** the data that is used to fit each model.
- **Test set:** data not used in model fitting that is used to compare model accuracy.
- **Validation set (optional):** If you perform too many comparisons with the test set, you run the risk of overfitting the test data. A validation set is a third set of data that is also withheld and only used to validate the best one or two models based on the test set.

Example in SAS: proc surveyselect can randomly assign observations to training and test sets.

3 Cross Validation

Whenever you have enough data, withholding a subset of the data prior to model building is ideal.

However, collecting new data can be very expensive such that creating a "test set" is not feasible.
Cross Validation: is a method that tries to estimate test set error using training data.

The process:

- Randomly separate our data into k-groups (usually five or ten).
- Treat all but one of the groups as a training set, the remaining group as a test set.
- Fit a model using the training data, predict for the test data.
- Repeat the process, each time treating a different group as the test data until all observations have a prediction.

SAS does not have an easy method for performing custom cross validation. For this purpose, we will stick to validation accuracy from a test set in our projects.

However, certain procedures use cross validation as a means of performing variable selection such as proc glmselect.

Cautions and Considerations:

- *Any* variable selection techniques or other forms of training must be included as part of the cross validation process. In other words, you can't use all of the data to select variables, then act "blind" to that same data in the model validation step.
  - The consequence of such a move is that you will likely overestimate your model's predictive capability.
  - Trying to embed variable selection into cross validation is extraordinarily difficult and not necessarily stable.
  - Check out this optional video for a more detailed explanation: `https://www.youtube.com/watch?v=r64tRyHFAJ8&list=PLOg0ngHtcqbPTlZzRHA2ocQZqB1D_qZ5V&index=23`
- The more groups you create, the more models you must fit, which can get computationally expensive.

- Too many groups makes it hard to estimate the true "test set" error.
  - Less groups, more bias, less variance in the test set error estimation. Try to select a number of cross validation groups that balance the bias and variance (usually five or ten groups).
- Check out chapter 5 in this book for more details on cross validation and other forms of model validation: `http://faculty.marshall.usc.edu/gareth-james/ISL/ISLR%20Seventh%20Printing.pdf`

3.4.1: Model Validation

Dr. Bean - Stat 5100

```
/* Project 2 is focused on using information regarding Tinder profiles
to predict the genuineness of the user. Information regarding the total
set of variables are included in the project 2 description. For purposes
of illustration, only a subset of variables are considered here. */

/* This first line of code will need to be changed */
FILENAME REFFILE '/home/u41171697/data/project2/tinder.csv';

PROC IMPORT DATAFILE=REFFILE replace
        DBMS=CSV
        OUT=WORK.tinder;
        GETNAMES=YES;
RUN;


/* Separate Into Training and Test Sets.
Only Fit Models to the Training Set. The variable
"Selected" separates training (0) from test (1)
seed - sets a random seed that allows your code to be reproduced
out - the name of the output dataset that includes the selected variable
rate - the percentage of points (between 0 and 1) that will be "selected"
       for validation */
proc surveyselect data=tinder seed=12345 out=tinder2
     rate=0.2 outall; /* Withold 20% for validation */
run;

proc print data=tinder2;
run;
```

Obs	Selected	ID	Genuine	SocPrivConc	InstPrivConc	Narcissism	SelfEsteem	Loneliness	Hookup	Friends
1	0	Subj57	-0.5	1	1	1.75	3.4	2.71	1	4
2	0	Subj310	-0.25	1.5	2.75	2.5	4	3.52	4	1.5
3	0	Subj303	1.5	3.75	4	1	3.4	3.27	3.25	4.25
4	0	Subj309	4	5	5	1.5	4.2	2.94	1	4.5
5	0	Subj426	2	2.25	3	2	2.2	4.19	5	3.5
6	0	Subj316	1.75	3.5	2.75	2.75	2.2	1.98	4.5	2
7	1	Subj5	2.5	2	2	3	3.2	2.98	3.75	1.75
8	0	Subj115	2	2.25	3	3	4	1	3.5	4
9	0	Subj327	0.5	1	1	2.25	3.8	1.1	2.75	2.5
10	0	Subj252	-2	1	3.75	3	2	2.79	4	3
11	0	Subj339	-1.5	4.25	4.5	2	3.8	2.57	2	4.25

```
data train; set tinder2;
if Selected = 0;
run;

data test; set tinder2;
```

```
if Selected = 1;
run;

proc print data = train;
run;

/* Fit one model with 4 variables.  */
proc reg data=train noprint;
 model genuine = socprivconc instprivconc narcissism selfesteem;
store regModel;
run;

/* Fit another model with more variables.  */
proc reg data=train noprint;
 model genuine = socprivconc instprivconc narcissism selfesteem loneliness
                 hookup friends partner travel selfvalidation entertainment;
store regModel2;
run;

/* Fit a third model with NO variables  */
proc reg data=train noprint;
 model genuine = ;
store regModel3;
run;

/* Calculate MSPR for each model by first making predictions
(via proc plm), then estimating errors (via a data step) and
calculating the means (via proc means). */
proc plm restore=regModel;
 score data=test out=newTest predicted;
 run;
 proc plm restore=regModel2;
 score data=test out=newTest2 predicted;
 run;

proc plm restore=regModel3;
 score data=test out=newTest3 predicted;
 run;


data newTest; set newTest;
ASE = (Genuine - Predicted)**2;
run;
data newTest2; set newTest2;
ASE = (Genuine - Predicted)**2;
run;
data newTest3; set newTest3;
ASE = (Genuine - Predicted)**2;
run;
```

```
proc means data = newTest;
var ASE;
run;
proc means data = newTest2;
var ASE;
run;
proc means data = newTest3;
var ASE;
run;
```

The SURVEYSELECT Procedure

Selection Method	Simple Random Sampling

Input Data Set	TINDER
Random Number Seed	12345
Sampling Rate	0.2
Sample Size	100
Selection Probability	0.200803
Sampling Weight	4.98
Output Data Set	TINDER2

The MEANS Procedure

Analysis Variable : ASE				
N	Mean	Std Dev	Minimum	Maximum
99	3.4138416	4.4208025	8.8289731E-6	18.9501242

The MEANS Procedure

Analysis Variable : ASE				
N	Mean	Std Dev	Minimum	Maximum
99	2.7276352	3.5324712	0.000862991	17.8117101

The MEANS Procedure

Analysis Variable : ASE				
N	Mean	Std Dev	Minimum	Maximum
99	3.5206134	3.8275996	0.0065659	21.7992795