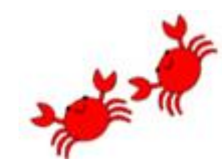
Using SAS® Regression Splines in the Banking Industry

Jonas V. Bilenas, AB
Nish Herat, Verizon



Outline

- History of Credit Scoring.
- Quick Introduction to Linear Regression.
- Scatterplot Smoothing Techniques to Investigate Nonlinearity.
- Simple Linear Splines to Handle Nonlinearity.
- Restricted Cubic Splines, AKA Natural Cubic Splines.
- MARS.

History of Credit Scoring

FICO

Interesting Chat on Binning in Linkedin started by Wensui Liu about 9 days ago; around March 10, 2019.

https://www.linkedin.com/feed/update/urn:li:activity:6510606874491052032/

Quick Introduction to Linear Regression

- Linear Regression has the form of:

$$Y_i = b_0 + b_1 * X_{1i} + b_2 * X_{2i} \ldots + b_p * X_{pi} + e_i$$

- **i** ranges from 1 to n where n is the number of observations in your modelling sample.
- **Y** is the variable you are trying to predict, also known as the dependent variable (DV).
- $\boldsymbol{X_1}, \boldsymbol{X_2}, .., \boldsymbol{X_{pi}}$ are the **p** predictors or independent variables (IVs) that are being used to predict **Y** in the linear equation. X variables can include nonlinear transformations of original X terms and/or include interactions of other independent variables.
- $\boldsymbol{e_i}$ is the error term (or residual) for each observation.

Quick Introduction to Linear Regression

- OLS Regression procedures fit the values of the b_j's for j=0 to p. Typically solving for b's that would minimize the sum of errors squared, $\sum_{i=1}^{n} e_i^2$
- Linear regression solutions require that the error terms follow a normal distribution with constant variance.
- Logistic regression requires a LOGIT link and a BINOMIAL distribution.
- Poisson or negative Binomial distribution for count models.

Issues with Binning Independent Variables

- Binning continuous variables will reduce the predictive power of the variable in a predictive model.
- Results are expressed in terms of a step function relationship between the predictors and the dependent variable.
- Results often don’t validate well in out of time samples.
- See Irwin and McClelland (2003).

Alternatives to Binning

- Use the continuous variable (ordinal, interval, and ratio) as a continuous independent variable.
- What if the relationships between an IV and a DV are not linear?
  - If the relationship is piecewise linear then linear splines can be used to fit the data points. However linear splines cannot fit curvilinear data. Decision on where to place Knots should be validated as being logical.
  - Power and/or log transformations of the independent or dependent variable can prove useful in linearizing the relationship.
  - Polynomial functions and/or piecewise polynomial splines such as cubic splines can fit curved relationships.
  - The issue with cubic splines is that the tails of the fit often don't behave well. As an alternative to cubic splines, restricted cubic splines force the tails to be linear and have other advantages we will review in this paper. Also Knot placement is not that important.

Scatterplot Smoothing Techniques to Investigate Nonlinearity.

- How you select variables is part art and part science. Some guidelines to consider:
  - Don’t use stepwise regression (Flom and Cassell, 2007 and Frank Harrell 2015).
  - Run collinearity diagnostics on all IVs to eliminate harmful collinearity. Many methods are available but would suggest the COLLIN option in PROC REG without the COLLINOINT option or PROC VARCLUS.
  - Run scatterplot smoothing for each IV on the X axis and the DV on the Y axes. This may generate lot of plots to look at you maybe able to weed out variables showing no bivariate relationship to the DV. There still maybe a multivariate effect but with large number of variables often common in banking and finance models, the effects maybe different than what is observed in the bivariate scatterplot.
  - These plots may show unexpected behavior that one may want to investigate the quality of the data.
  - Missing values will not show up in the plots but that is another discussion on imputation.
- Understanding your data is as important as running the regression model.

Scatterplot Smoothing Techniques to Investigate Nonlinearity.

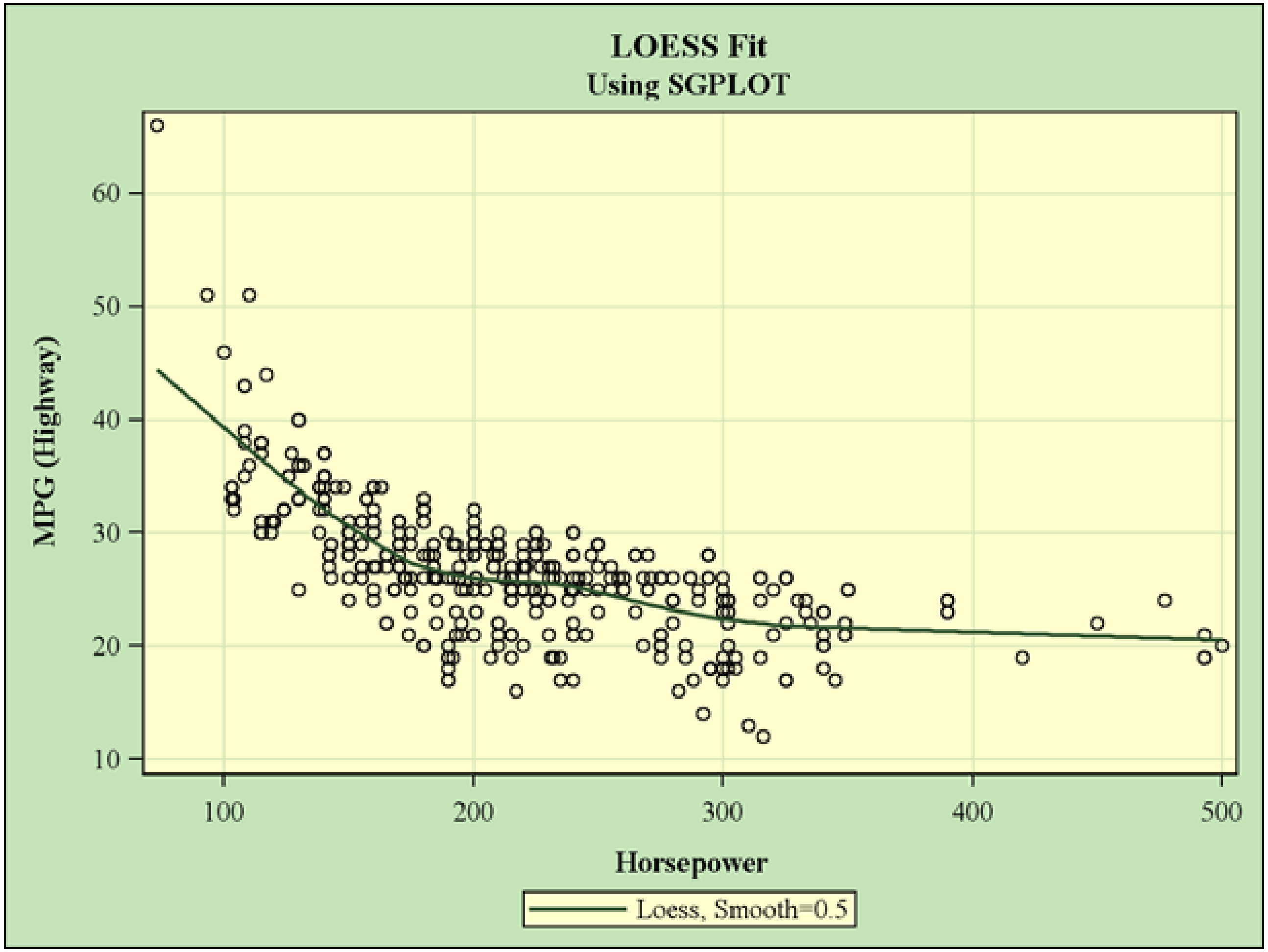
- There are 2 scatterplot smoothing options in the new SG procedures which we will review.
  - LOESS: Not a spline but a nonparametric local weighted regression function fit to the data within a chosen neighborhood of points. There is a LOESS procedure if you want more control of the output which we will illustrate a few examples.

  - PBSPLINE: Plots a spline that automatically picks the smoothing parameter that minimizes AICC (Eilers and Marx 1996).

Scatterplot Smoothing: LOESS Procedure Example using SGPLOT

```
%let DS=sashelp.cars;
%let Y=MPG_Highway;
%let X=Horsepower;


ods rtf  file ="LOESS_TESTING.doc" style=banker;

ods graphics on / ANTIALIASMAX=21500;

proc sgplot data=&DS.;
  LOESS Y=&Y. X=&X. / smooth=0.5;
  XAXIS  grid;
  YAXIS  grid;
  title LOESS Fit;
  title2 Using SGPLOT;
run;

ods graphics off;
ods rtf close;
```

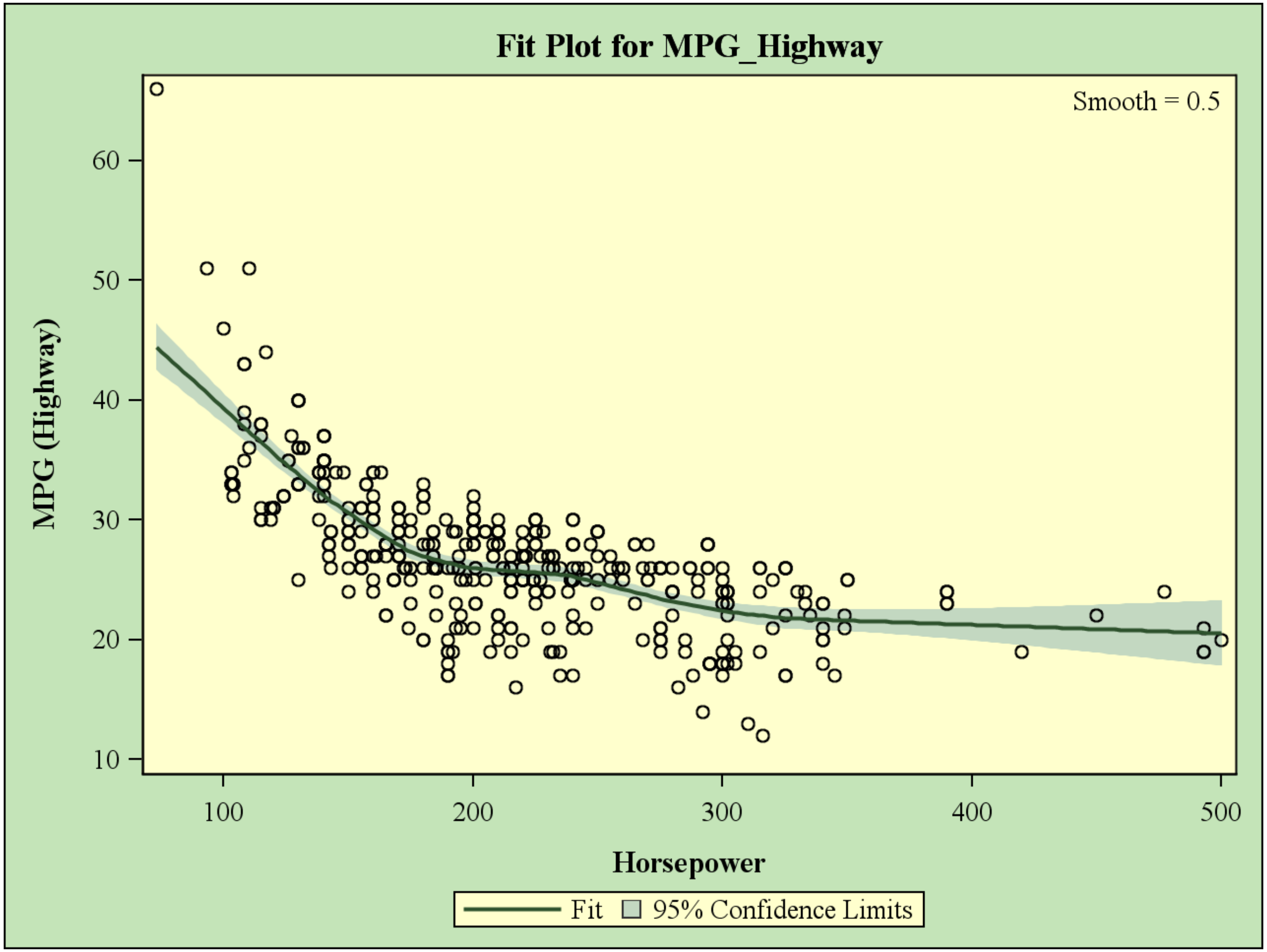
Scatterplot Smoothing: <u>LOESS Procedure Example using SGPLOT</u>



LOESS Procedure Example using PROC LOESS

```
%let DS=sashelp.cars;
%let Y=MPG_Highway;
%let X=Horsepower;

options orientation=landscape;

ods rtf  file ="LOESS_TESTING.doc" style=banker;

ods graphics on / ANTIALIASMAX=21500;

title PROC LOESS;
title2;
proc loess data=&ds. plots(only)=(FitPlot);
  model &Y.=&x.
       /smooth=0.5 alpha=.05 all;
run;

ods graphics off;
ods rtf close;
```

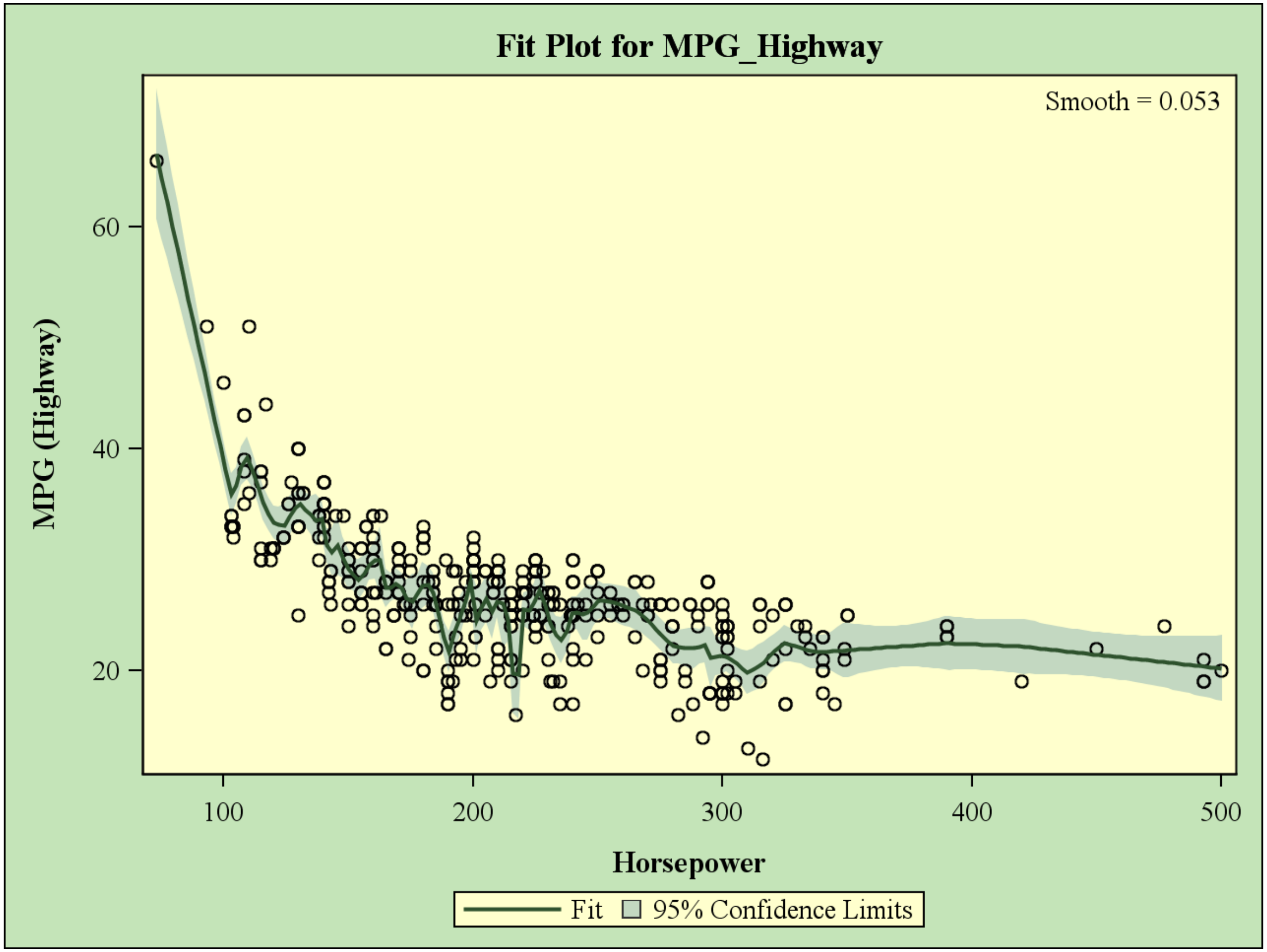
LOESS Procedure Example using PROC LOESS



PROC LOESS Variations: Let LOESS pick the Smooth= parm

```
%let DS=sashelp.cars;
%let Y=MPG_Highway;
%let X=Horsepower;

options orientation=landscape;
ods rtf  file ="LOESS_TESTING.doc" style=banker;

ods graphics on / ANTIALIASMAX=21500;

proc loess data=&ds. plots(only)=(fitplot);
  model &Y.=&x.
       /select=AICC alpha=.05 all;
run


ods graphics off;
ods rtf close;
```

PROC LOESS Variations: Let LOESS pick the Smooth= parm

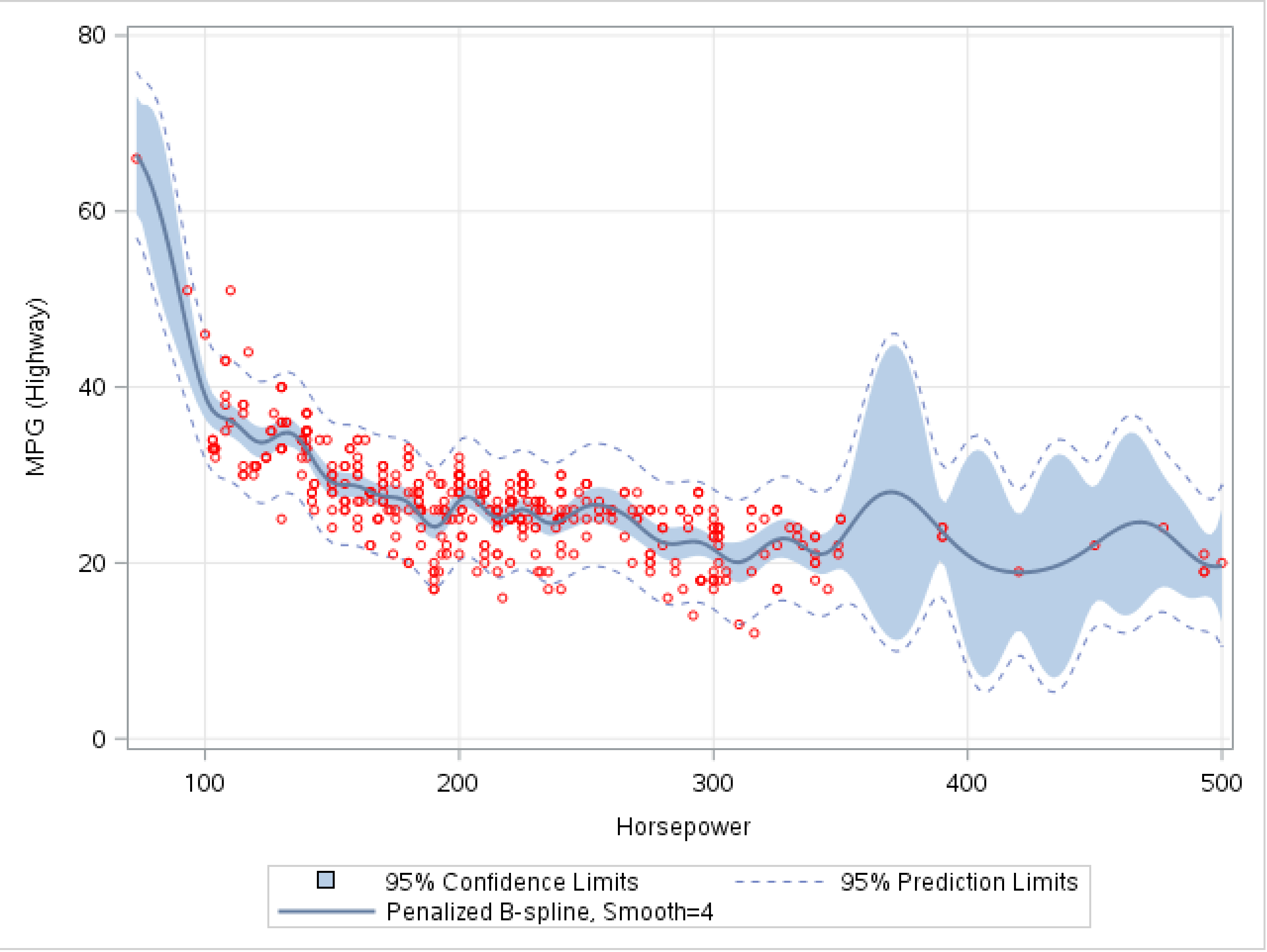


Scatterplot Smoothing: PBSPLINE Example using SGPLOT

- Let's simulate some data that looks similar to the ratio of actual balances on a loan product over the contractual balance each month on book (mob) for a number of vintages.

```
proc sgplot data=sashelp.cars;
  pbspline y=MPG_HIGHWAY x=HORSEPOWER /
           CLM
           CLI
           alpha=0.05
           smooth=4
           markerattrs=(symbol=dot color=red size=5)
           ;
  xaxis grid; yaxis grid;
run;
```

Scatterplot Smoothing: PBSPLINE Example using SGPLOT. Output



Scatterplot Smoothing For Binary Dependent Variables

- For credit scores based on logistic regression we need to see the LOG of ODDS transformation as the DV variable.
- Options:
  - One can run a binning for each IV using PROC RANK and calculate the log of odds for each bin.
  - Calculate the log of odds for each level of the IV.
- For most multivariate models, the model will be developed not using binning of the IV. Binning is done to visualize the Log of Odds as opposed to the binary results.

$$DV = LN\left(\frac{mean(event)}{(1 - mean(event))}\right)$$

Scatterplot Smoothing For Binary Dependent Variables

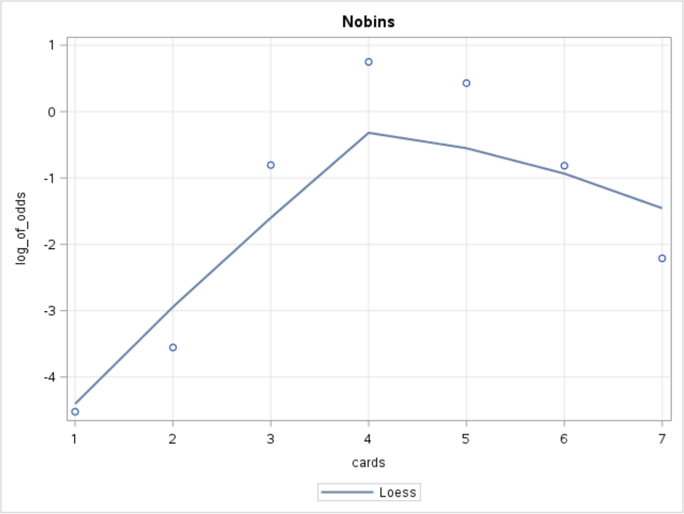
- **Simulated data. Code is available if interested.**
- **Getting DV, Log of Odds without binning. Large data (n=23,938) and few classes of the IV; Number of Revolving Open Credit Cards.**
- **NO BINNING**

```
proc means data=simulate nway noprint;
    class cards;
    var good;
    output out=nobins mean=;
 run;
 proc print data=nobins; run;

 data model;
   set nobins;
   log_of_odds = log(good/(1-good));
run;

  proc sgplot data=model;
    loess y=log_of_odds x=cards;
    xaxis grid;
    yaxis grid;
    title Nobins
  run;
title;
```

Scatterplot Smoothing For Binary Dependent Variables

- **Loess Plot**

Obs	cards	_TYPE_	_FREQ_	good
1	1	1	5123	0.01074
2	2	1	5506	0.02779
3	3	1	2290	0.30873
4	4	1	109	0.67890
5	5	1	4447	0.60558
6	6	1	5055	0.30663
7	7	1	1408	0.09872



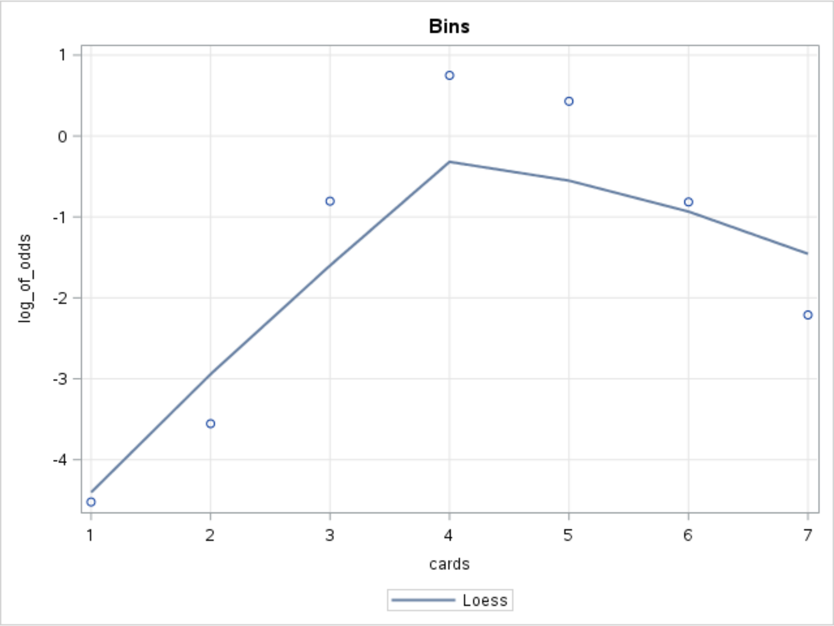
Scatterplot Smoothing For Binary Dependent Variables with 100 bins.

```
proc rank data=simulate
          out=ranky
          ties = low
          groups=100;
  var   cards;
  ranks r_cards;
 run;

 proc means data=ranky nway noprint;
   class r_cards;
   var cards good;
   output  out=bins mean=;
 run;
 proc print data=bins; run;

 data model2;
   set bins;
    log_of_odds = log(good/(1-good));
  run;

  proc sgplot data=model2;
    loess y=log_of_odds x=cards;
    xaxis grid;
    yaxis grid;
    title Bins;
  run;
```

Scatterplot Smoothing For Binary Dependent Variables

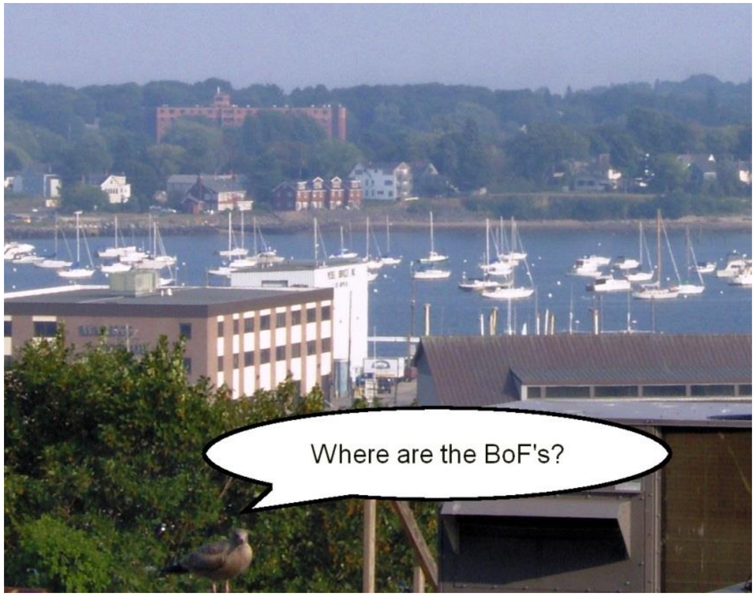
- **Loess Plot**

Obs	r_cards	_TYPE_	_FREQ_	cards	good
1	0	1	5123	1	0.01074
2	21	1	5506	2	0.02779
3	44	1	2290	3	0.30873
4	53	1	109	4	0.67890
5	54	1	4447	5	0.60558
6	73	1	5055	6	0.30663
7	94	1	1408	7	0.09872



Bilenas, J. (2010), "Using PROC RANK and PROC UNIVARIATE to Rank or Decile Variables"


OPEN


Where are the BoF's?

Some parametric and nonparametric splines in model development.

- There are a growing number of model procedures in SAS that incorporate splines as independent variables. Some of these are parametric splines and some are nonparametric. These can be used for scatterplot smoothing and also included in multivariate regression models.
- In this tutorial we will focus in on these spline methods.
  - Linear Splines
  - ~~Monotonic Splines using PROC TRANSREG~~
  - Restricted (or Natural) Cubic Splines.
  - MARS via ADAPTIVEREG.

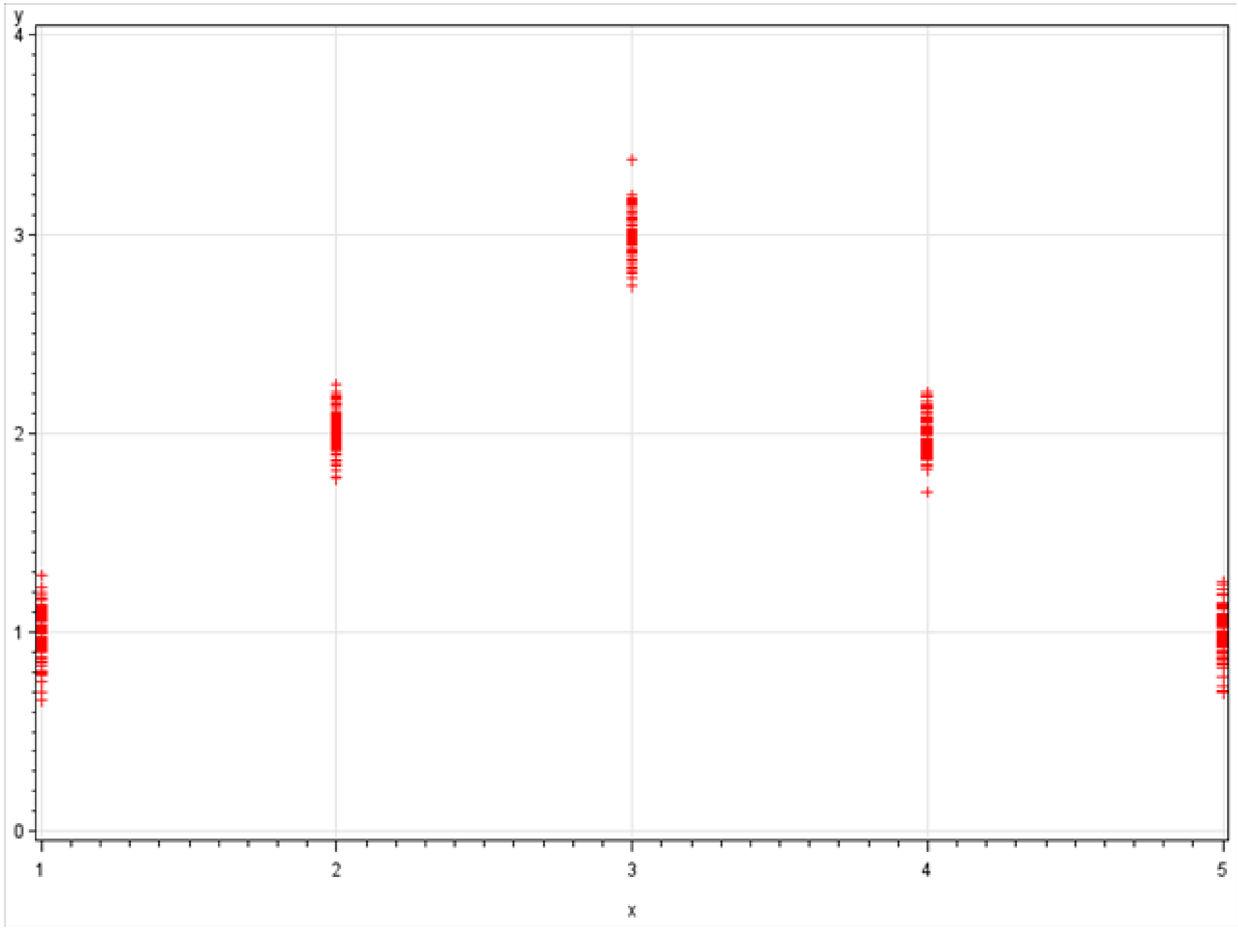
- Some SAS procedures use splines but may not provide spline transformations. However, with these procedures you can save a MODEL STORE to score other data using PROC PLM (Tobias and Cai, 2010).

Simple Linear Spline

Why Looking at Correlations May not Identify Strong Predictors.

- What does a correlation of 0 mean?
- If a potential IV has a correlation with the DV of, say -0.006 should that variable be dropped?

Why Looking at Correlations May not Identify Strong Predictors.



Correlation between x and y is low at -0.00617.

Simple Linear Splines to Handle Nonlinearity.

$$Y = b_0 + b_1 * X + b_2 * (X - a)_+ + b_3 * (X - b)_+ + b_4 * (X - c)_+ \ \ldots$$

Where $(u)_+ = u,\ u > 0$

$= 0, u \le 0$

a, b, c … are locations (knots) where the curve changes.

Slopes are additive:

- For X $\le$ a: slope is b_1
- For a < X $\le b$: *slope is* $b_1 + b_2$
- For b < X $\le c$: *slope is* $b_1 + b_2 + b_3$
- For X $> c$: *slope is* $b_1 + b_2 + b_3 + b_4$

Simple Linear Splines to Handle Nonlinearity. Code:

```
data mod;
  set sample;
  xt = max(0,x-3);
  *xt = 0 <> x-3;  /* this will work too */
run;

proc reg data=mod;
  model y = x xt;
run;
```

Simple Linear Splines to Handle Nonlinearity. Output:

Analysis of Variance

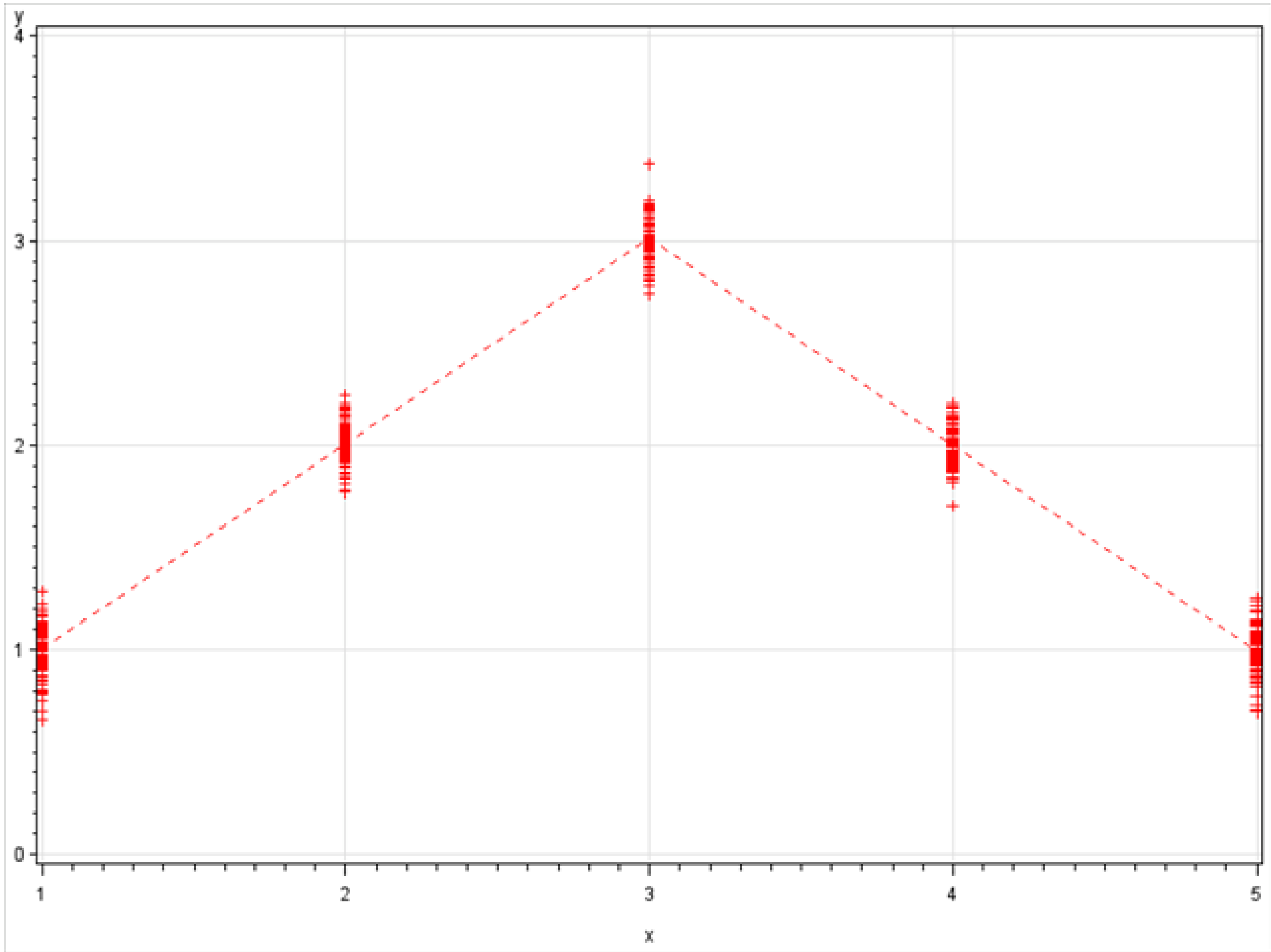
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	279.99950	139.99975	3610.46	<.0001
Error	497	19.27177	0.03878		
Corrected Total	499	299.27127			

Root MSE	0.19692	R-Square	**0.9356**
Dependent Mean	1.79263	Adj R-Sq	0.9353
Coeff Var	10.98478		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	-0.00009683	0.02958	-0.00	0.9974
x	1	0.99757	0.01331	74.93	<.0001
xt	1	-1.99998	0.02354	-84.98	<.0001

Simple Linear Splines to Handle Nonlinearity. Model Fit:



Restricted Cubic Splines: Parametric Splines

- Many Polynomial transformations and/or Cubic Splines do not fit well at the tails. An alternative to consider is Restricted Cubic Splines (Stone and Koo 1985). Also known as Natural Cubic Splines.
- Splines are required to be linear at end points. As a result fewer terms are required in the model
- Placement of Knots are not important. Usually predetermined percentiles based on sample size:

k	**Quantiles**
3	.10 .5 .90
4	.05 .35 .65 .95
5	.05 .275 .5 .725 .95
6	.05 .23 .41 .59 .77 .95
7	.025 .1833 .3417 .5 .6583 .8167 .975

Restricted Cubic Splines

- Percentile values can be derived using PROC UNIVARIATE.

- Can Optimize number of Knots selecting number based on minimizing AICC or SBC.

- Provides a parametric regression function.

- Sometimes knot transformations make for difficult interpretation. Graphical review of the model will be required.

- May be difficult to incorporate interaction terms.

- Much more efficient than categorizing continuous variables into dummy terms.

- Macro available from Frank Harrell.
  - http://biostat.mc.vanderbilt.edu/wiki/pub/Main/SasMacros/survrisk.txt

Restricted Cubic Splines

```
proc univariate data=sashelp.cars noprint;
  var horsepower;
  output out=knots pctlpre=P_ pctlpts=5 27.5 50 72.5 95;
run;

proc print data=knots; run;
```

Obs	P_5	P_27_5	P_50	P_72_5	P_95
1	115	170	210	245	340

Restricted Cubic Splines

```
options nocenter mprint;
data test;
  set sashelp.cars;
  %rcspline (horsepower,115, 170, 210, 245, 340);
run;
```

%rcspline

```
/*MACRO RCSPLINE

  For a given variable named X and from 3-10 knot locations,
  generates SAS assignment statements to compute k-2 components
  of cubic spline function restricted to be linear before the
  first knot and after the last knot, where k is the number of
  knots given.  These component variables are named c1, c2, ...
  ck-2, where c is the first 7 letters of X.

  Usage:

  DATA; ....
  %RCSPLINE(x,knot1,knot2,...,norm=)   e.g. %RCSPLINE(x,-1.4,0,2,8)

       norm=0 : no normalization of constructed variables
       norm=1 : divide by cube of difference in last 2 knots
                makes all variables unitless
       norm=2 : (default) divide by square of difference in outer knots
                makes all variables in original units of x

  Reference:

  Devlin TF, Weeks BJ (1986): Spline functions for logistic regression
  modeling. Proc Eleventh Annual SAS Users Group International.
  Cary NC: SAS Institute, Inc., pp. 646-51.

  Author  : Frank E. Harrell Jr.
            Clinical Biostatistics, Duke University Medical Center
  Date    : 10 Apr 88
  Mod     : 22 Feb 91 - normalized as in S function rcspline.eval
            06 May 91 - added norm, with default= 22 Feb 91
            10 May 91 - fixed bug re precedence of <>

                                                                 */
```

%rcspline

```
%MACRO RCSPLINE(x,knot1,knot2,knot3,knot4,knot5,knot6,knot7,
                  knot8,knot9,knot10, norm=2);
%LOCAL j v7 k tk tk1 t k1 k2;
%LET v7=&x; %IF %LENGTH(&v7)=8 %THEN %LET v7=%SUBSTR(&v7,1,7);
  %*Get no. knots, last knot, next to last knot;
    %DO k=1 %TO 10;
    %IF %QUOTE(&&knot&k)=  %THEN %GOTO nomorek;
    %END;
%LET k=11;
%nomorek: %LET k=%EVAL(&k-1); %LET k1=%EVAL(&k-1); %LET k2=%EVAL(&k-2);
%IF &k<3 %THEN %PUT ERROR: <3 KNOTS GIVEN.  NO SPLINE VARIABLES CREATED.;
%ELSE %DO;
 %LET tk=&&knot&k;
 %LET tk1=&&knot&k1;
 DROP _kd_; _kd_=
 %IF &norm=0 %THEN 1;
 %ELSE %IF &norm=1 %THEN &tk - &tk1;
 %ELSE (&tk - &knot1)**.666666666666; ;
    %DO j=1 %TO &k2;
    %LET t=&&knot&j;
    &v7&j=max((&x-&t)/_kd_,0)**3+((&tk1-&t)*max((&x-&tk)/_kd_,0)**3
        -(&tk-&t)*max((&x-&tk1)/_kd_,0)**3)/(&tk-&tk1)%STR(;);
    %END;
 %END;
%MEND;
```

Restricted Cubic Splines: Variable Transformations

```
LOG:
MPRINT(RCSPLINE):   DROP _kd_;
MPRINT(RCSPLINE):   _kd_= (340 - 115)**.666666666666 ;
MPRINT(RCSPLINE):
horsepower1=max((horsepower-115)/_kd_,0)**3+((245-
115)*max((horsepower-340)/_kd_,0)**3
-(340-115)*max((horsepower-245)/_kd_,0)**3)/(340-245);
MPRINT(RCSPLINE):   ;
MPRINT(RCSPLINE):
horsepower2=max((horsepower-170)/_kd_,0)**3+((245-
170)*max((horsepower-340)/_kd_,0)**3
-(340-170)*max((horsepower-245)/_kd_,0)**3)/(340-245);
MPRINT(RCSPLINE):   ;
MPRINT(RCSPLINE):
horsepower3=max((horsepower-210)/_kd_,0)**3+((245-
210)*max((horsepower-340)/_kd_,0)**3
-(340-210)*max((horsepower-245)/_kd_,0)**3)/(340-245);
MPRINT(RCSPLINE):   ;
43    run;
```

Restricted Cubic Splines

```
proc reg data=sashelp.cars;
  model MPG_Highway = horsepower horsepower1
                      horsepower2 horsepower3;
  LINEAR: TEST horsepower1, horsepower2, horsepower3;
run; quit;

proc genmod data=test;
 model MPG_Highway = horsepower horsepower1
                     horsepower2 horsepower3 / dist=normal link=identity;
 output out=spline pred=fit;
 run;

 proc sort data=spline;
   by horsepower;
 run;

 proc sgplot data=spline;
   scatter x=horsepower y=MPG_Highway;
   series x=horsepower y=Fit / lineattrs=(thickness=3 color=red);
   xaxis grid;
   yaxis grid;
run;
```

Restricted Cubic Splines

Number of Observations Read	428
Number of Observations Used	428

Analysis of Variance					
Source	**DF**	**Sum of Squares**	**Mean Square**	**F Value**	**Pr > F**
Model	4	8147.64458	2036.91115	145.37	<.0001
Error	423	5926.86710	14.01151		
Corrected Total	427	14075			

Root MSE	3.74319	**R-Square**	0.5789
Dependent Mean	26.84346	**Adj R-Sq**	0.5749
Coeff Var	13.94453		

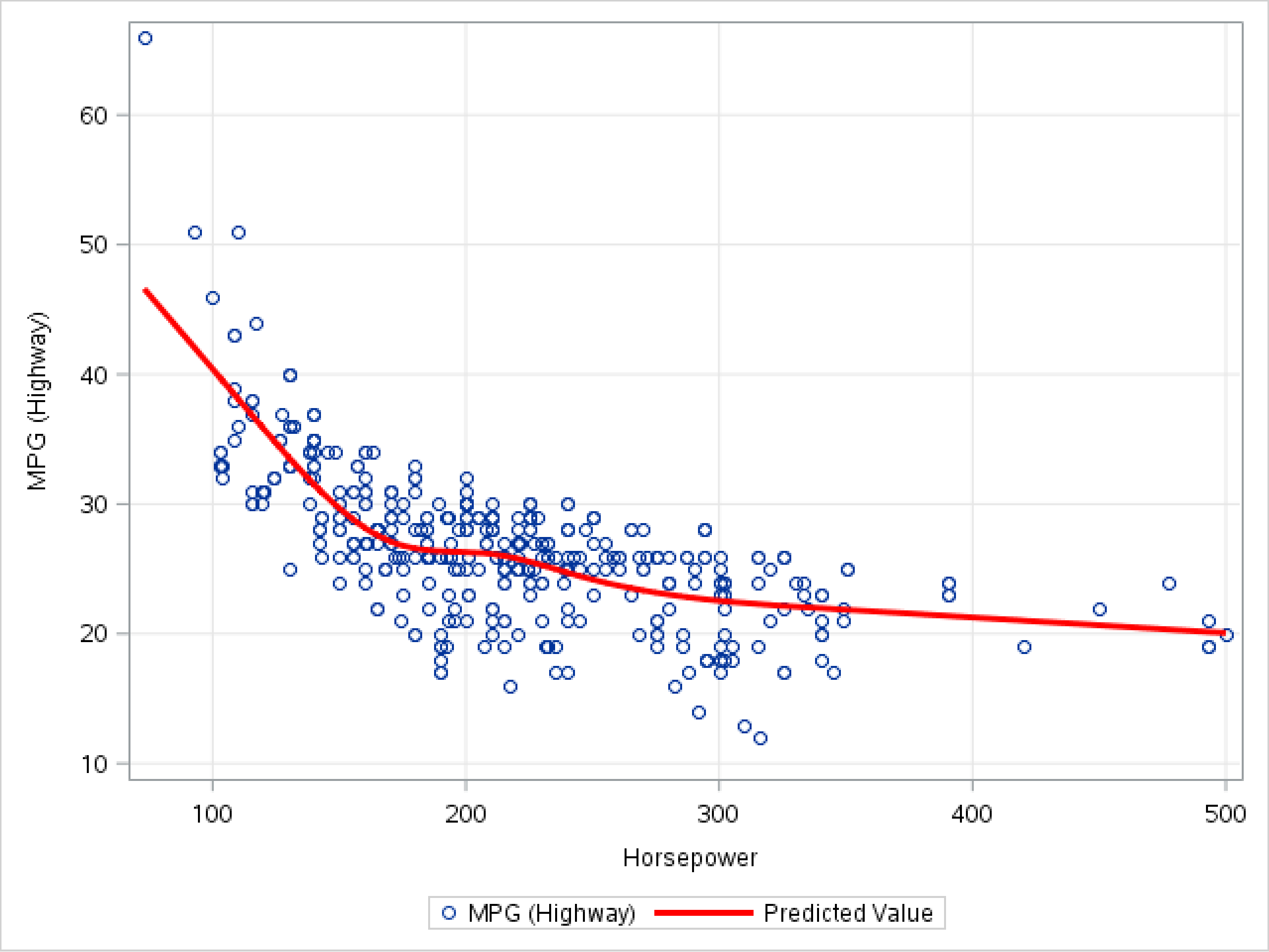
Restricted Cubic Splines

Parameter Estimates						
Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	Intercept	1	63.32145	2.50445	25.28	<.0001
Horsepower		1	-0.22900	0.01837	-12.46	<.0001
horsepower1		1	0.83439	0.12653	6.59	<.0001
horsepower2		1	-2.53834	0.49019	-5.18	<.0001
horsepower3		1	2.55417	0.66356	3.85	0.0001

Test LINEAR Results for Dependent Variable MPG_Highway				
Source	DF	Mean Square	F Value	Pr > F
Numerator	3	750.78949	53.58	<.0001
Denominator	423	14.01151		

NOTE: GENMOD output not displayed in this presentation.

Restricted Cubic Splines (5 Knots)



Using PROC GLMSELECT

Wicklin, R. (2017) The DO Loop Blog: Regression with restricted cubic splines in SAS. https://blogs.sas.com/content/iml/2017/04/19/restricted-cubic-splines-sas.html.

```
title Restricted Cubic Spline;
title2 Harell Knot Placement;

proc glmselect data=sashelp.cars;
  effect spl = spline(horsepower / details naturalcubic  basis=tpf(noint)
                              KNOTMETHOD=LIST(115, 170, 210, 245, 340) );
   model MPG_Highway = spl / selection=none; /* fit model by using
                                                spline effects */
   output out=SplineOut predicted=Fit;       /* output predicted values for
                                                graphing */
quit;

proc sort data=splineout;
  by horsepower;
run;

proc sgplot data=SplineOut noautolegend;
   scatter x=horsepower y=MPG_Highway;
   series x=horsepower y=Fit / lineattrs=(thickness=3 color=red);
   xaxis grid;
   yaxis grid;
run;
```

Using PROC GLMSELECT

Restricted Cubic Spline
Harell Knot Placement
The GLMSELECT Procedure

Least Squares Summary				
Step	Effect Entered	Number Effects In	Number Parms In	SBC
* Optimal Value of Criterion				
0	Intercept	1	1	1501.0621
1	spl	2	5	1155.1343*

Least Squares Summary			
Step	Effect Entered	Number Effects In	SBC
* Optimal Value of Criterion			
0	Intercept	1	1501.0621
1	Horsepower	2	1274.8170*

```
/*SBC Excluding SPLINES */
proc glmselect data=sashelp.cars;
  model MPG_HIGHWAY =  horsepower / selection=none;
run;
```

Using PROC GLMSELECT

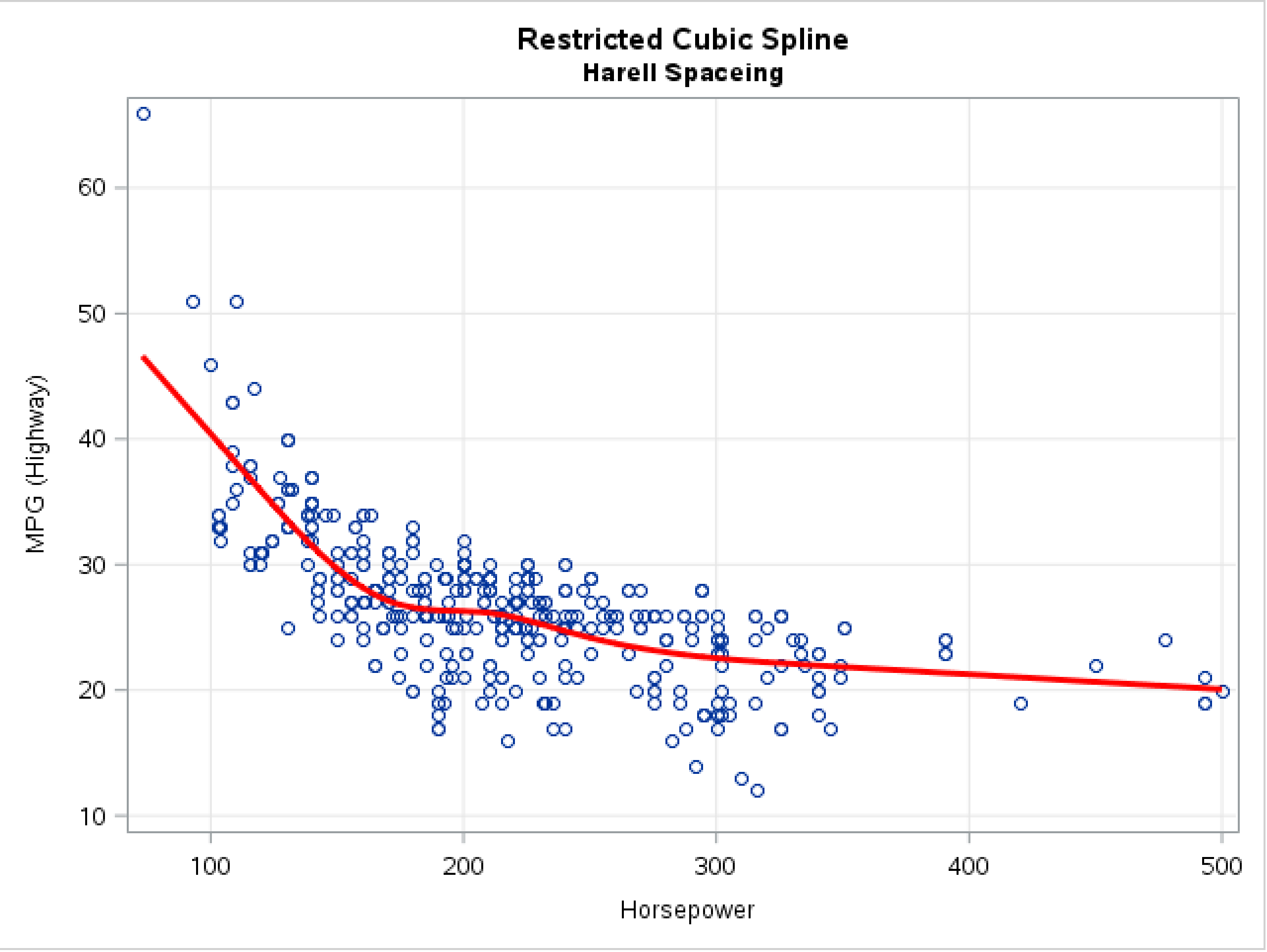
Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	4	8147.64458	2036.91115	145.37	<.0001
Error	423	5926.86710	14.01151		
Corrected Total	427	14075			

Root MSE	3.74319
Dependent Mean	26.84346
R-Square	0.5789
Adj R-Sq	0.5749
AIC	1564.83872
AICC	1565.03824
SBC	1155.13433

Parameter Estimates					
Parameter	DF	Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	63.321452	2.504447	25.28	<.0001
spl 1	1	-0.229002	0.018374	-12.46	<.0001
spl 2	1	0.003708	0.000562	6.59	<.0001
spl 3	1	-0.008524	0.001646	-5.18	<.0001
spl 4	1	0.006559	0.001704	3.85	0.0001

- **Analysis of Variance, R-Squares Match the results we got using the %RCSPLINE**
- **First 2 estimates match. Spl 1 is the Horsepower. Spl2 – spl4 are the 3 knot transformed terms; Horsepower1 – Horsepower3. GLMSELCT may be using different normalizations (norm=0 or norm=1). Nope, no match.**
- **Similar splines can be added in PROC LOGISTIC and other procedures using the EFFECT statement.**
- **I should of named the EFFECT HORSEPOWER as opposed to SPL. You can run multivariate splines in the model so you want to name the spline the original IV.**

Using GLMSELECT



And Now, for something completely different:

MARS

MARS: Multivariate Adaptive Regression Splines

- MARS (Friedman 1991) modeling methodology is available through PROC ADAPTIVEREG *(experimental in SAS/STAT 12.1; production in 13.1).*
- MARS is a nonparametric regression technique that derives knots directly from the data.
- Use recursive partitioning concepts like in a binomial tree model, but instead of bins you get continuous, differentiable piecewise truncated power spline functions, also known as "basis" functions of the form:

  MAX(0, x-k)

  or

  MAX(0, k-x)

  where *k* is the knot value and *x* is the value of the independent variable.

MARS: Multivariate Adaptive Regression Splines

- MARS was designed for high dimensional problems that can involve high order interactions that can result in equations that can be extremely difficult to interpret.

- Simulation and visual methods as suggested by Flom (2015) can be used to examine monotonicity and the effects of changes to the independent variables.

PROC ADAPTIVEREG: Features

- SAS implementation of MARS in PROC ADAPTIVEREG offers a myriad features that demonstrate the power of this flexible methodology.

- It can handle non-normal distributions such as
  - Binomial for logistic regression.
    - Single trial as well as events/trials syntax support for the dependent variable
  - Poisson/negative binomial for count models

MARS: Process

- MARS differs from the other techniques mentioned in that it determines the final selection of splines and knots through a form of forward (growing) and backward (pruning) stepwise selection.
  1. In the forward process pairs of basis functions are added until a lack of fit (LOF) criteria are met.
  2. Backward selection then removes bases from the over fit forward process and selects a model that minimizes the generalized cross validation criterion (GCV).

- Both LOF and GCV are dependent on the residual sum of squares.

MARS: Default model code and forward bases

```
proc adaptivereg data=sashelp.cars plots=all details=bases;
  model MPG_Highway = horsepower;
run;
```

The ADAPTIVEREG Procedure

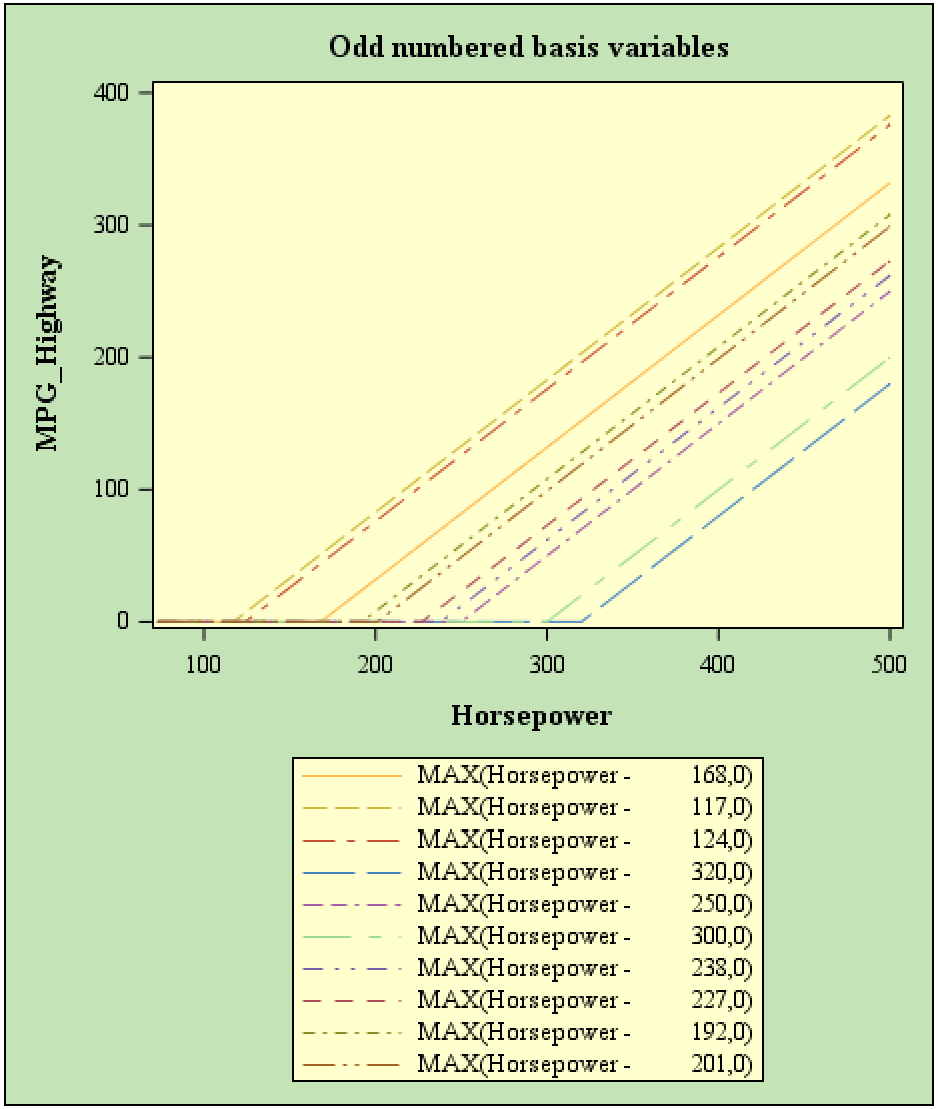
Fit Statistics

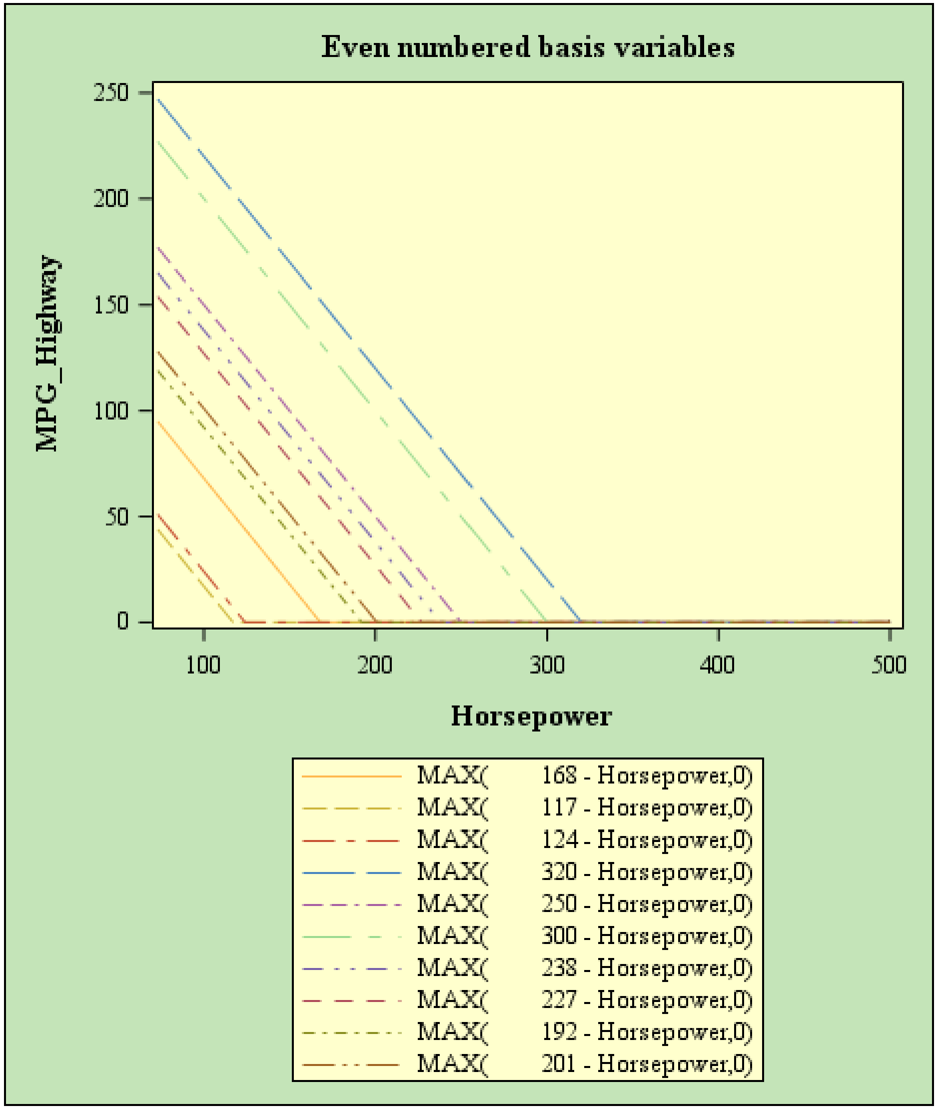
GCV	13.44050
GCV R-Square	0.59319
Effective Degrees of Freedom	15
R-Square	0.61943
Adjusted R-Square	0.61308
Mean Square Error	12.75330
Average Square Error	12.51492

Basis Information

Name	Transformation
Basis0	1
Basis1	Basis0*MAX(Horsepower - 168,0)
Basis2	Basis0*MAX(168 - Horsepower,0)
Basis3	Basis0*MAX(Horsepower - 117,0)
Basis4	Basis0*MAX(117 - Horsepower,0)
Basis5	Basis0*MAX(Horsepower - 124,0)
Basis6	Basis0*MAX(124 - Horsepower,0)
Basis7	Basis0*MAX(Horsepower - 320,0)
Basis8	Basis0*MAX(320 - Horsepower,0)
Basis9	Basis0*MAX(Horsepower - 250,0)
Basis10	Basis0*MAX(250 - Horsepower,0)
Basis11	Basis0*MAX(Horsepower - 300,0)
Basis12	Basis0*MAX(300 - Horsepower,0)
Basis13	Basis0*MAX(Horsepower - 238,0)
Basis14	Basis0*MAX(238 - Horsepower,0)
Basis15	Basis0*MAX(Horsepower - 227,0)
Basis16	Basis0*MAX(227 - Horsepower,0)
Basis17	Basis0*MAX(Horsepower - 192,0)
Basis18	Basis0*MAX(192 - Horsepower,0)
Basis19	Basis0*MAX(Horsepower - 201,0)
Basis20	Basis0*MAX(201 - Horsepower,0)

MARS: Default model individual bases



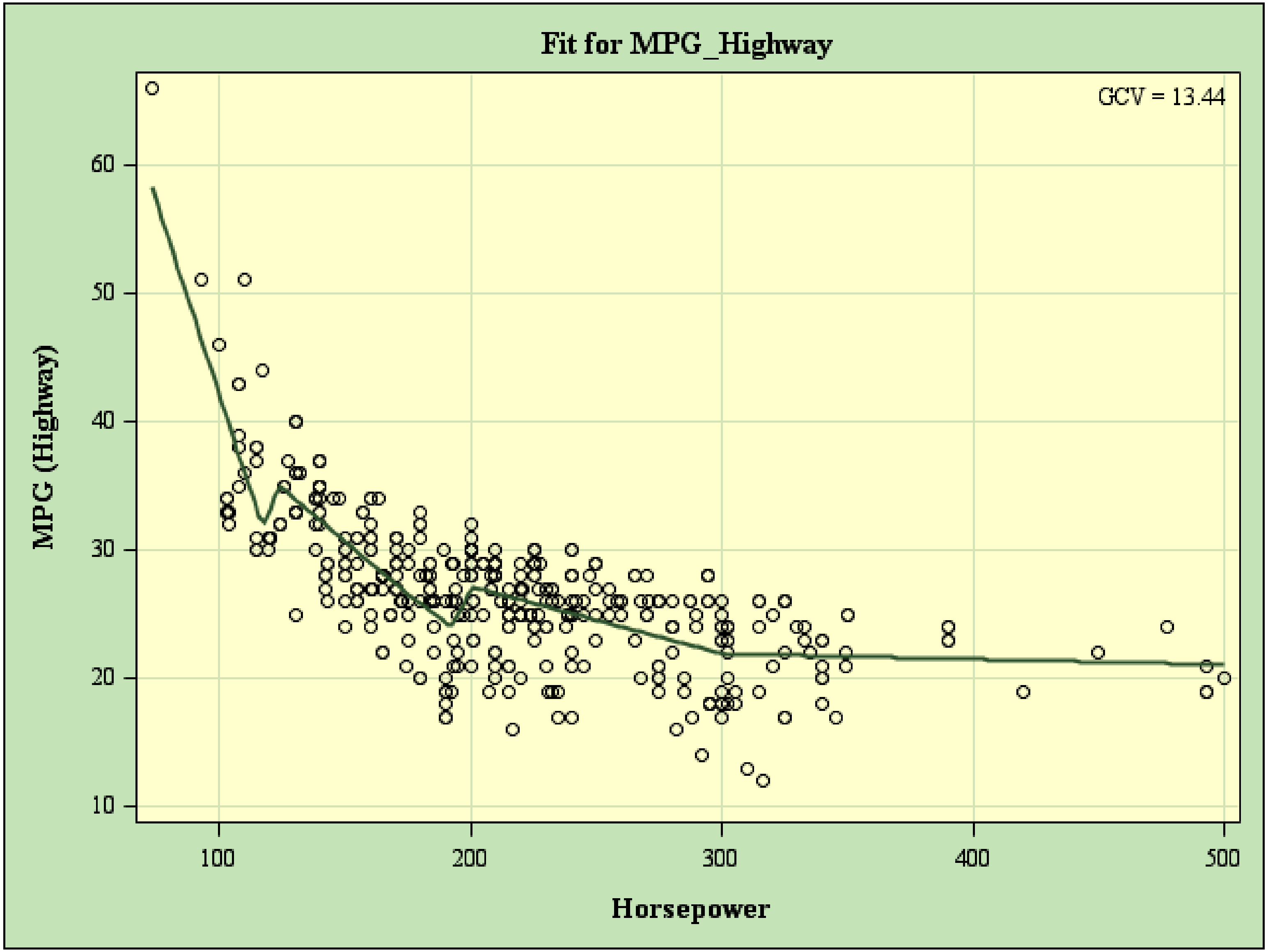


MARS: Default final model listing

Regression Spline Model after Backward Selection

Name	Coefficient	Parent	Variable	Knot
Basis0	1.1099		Intercept	
Basis1	-0.5889	Basis0	Horsepower	168.00
Basis2	0.6016	Basis0	Horsepower	168.00
Basis3	1.0551	Basis0	Horsepower	117.00
Basis5	-0.6193	Basis0	Horsepower	124.00
Basis11	0.04818	Basis0	Horsepower	300.00
Basis17	0.4986	Basis0	Horsepower	192.00
Basis19	-0.3979	Basis0	Horsepower	201.00

MARS: Default model output



MARS: Reducing the number of basis functions

```
proc adaptivereg data=sashelp.cars plots=all details=bases;
  model MPG_Highway = horsepower/MAXBASIS=5;
run;
```

The ADAPTIVEREG Procedure

Fit Statistics

GCV	14.11053
GCV R-Square	0.57291
Effective Degrees of Freedom	7
R-Square	0.58483
Adjusted R-Square	0.58189
Mean Square Error	13.78154
Average Square Error	13.65275

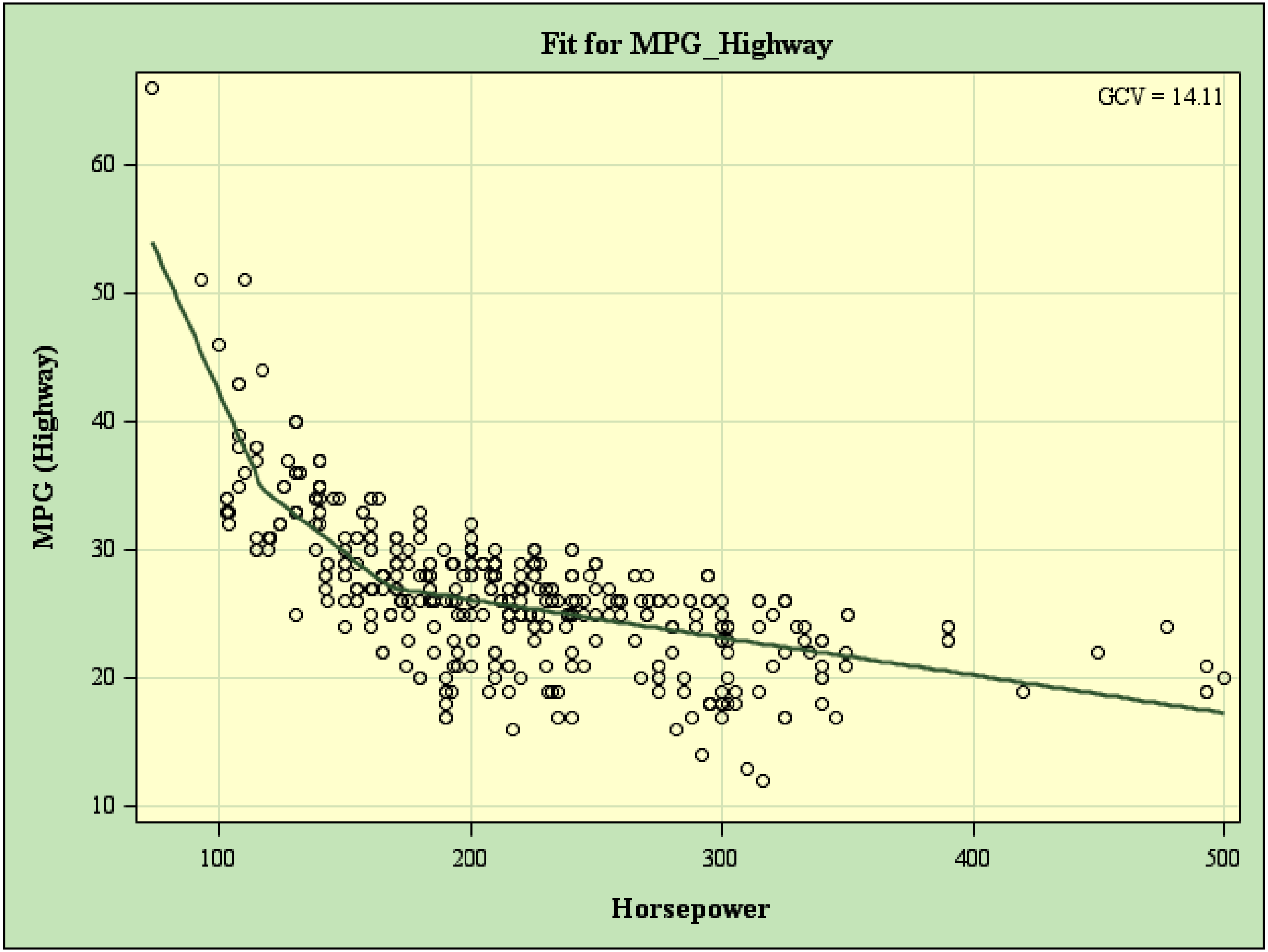
Basis Information

Name	Transformation
Basis0	1
Basis1	Basis0*MAX(Horsepower - 168,0)
Basis2	Basis0*MAX(168 - Horsepower,0)
Basis3	Basis0*MAX(Horsepower - 117,0)
Basis4	Basis0*MAX(117 - Horsepower,0)

Regression Spline Model after Backward Selection

Name	Coefficient	Parent	Variable	Knot
Basis0	12.3262		Intercept	
Basis1	-0.3182	Basis0	Horsepower	168.00
Basis2	0.4394	Basis0	Horsepower	168.00
Basis3	0.2888	Basis0	Horsepower	117.00

MARS: Reduced basis function output



MARS: Cross validation and scoring code

```
proc adaptivereg data=sashelp.cars plots=all details=bases SEED=789;
  ODS OUTPUT BASES=b BWDPARAMS=p;
  model MPG_Highway = horsepower/maxbasis=5;
  PARTITION FRACTION(TEST=0.25 VALIDATE=0.25 );
run;
data b;
     set b;
     transformation=transtrn(transformation,"Basis0*",trimn(''));
run;
data _null_;
     set b end=eof;
     file "bases.sas";
     put name '= ' transformation '; '   ' label '  name ' = "' transformation
'";';
run;
proc sort data=b; by name; run;
proc sort data=p; by name; run;
data _null_;
     merge b p(in=p); by name; if p;
     file "score.sas";
     if _n_ = 1 then put "predicted = 0;";
     put "predicted + " coefficient best16. ' * ' transformation +(-1) ';';
run;
```

MARS: Bases.sas and Scores.sas

```
data bases_score;
  set sashelp.cars;
  %include "bases.sas";
  %include "score.sas";
run;
```

```
data bases_score;
  set sashelp.cars;
  Basis0 = 1 ;  label Basis0  = "1 ";
  Basis1 = MAX(Horsepower - 160,0) ;  label Basis1  = "MAX(Horsepower - 160,0) ";
  Basis2 = MAX(160 - Horsepower,0) ;  label Basis2  = "MAX(160 - Horsepower,0) ";
  Basis3 = MAX(Horsepower - 120,0) ;  label Basis3  = "MAX(Horsepower - 120,0) ";
  Basis4 = MAX(120 - Horsepower,0) ;  label Basis4  = "MAX(120 - Horsepower,0) ";
  predicted = 0;
  predicted + 20.3511093361348 * 1;
  predicted + -0.2246883942159 * MAX(Horsepower - 160,0);
  predicted + 0.36250353038409 * MAX(160 - Horsepower,0);
  predicted + 0.18824377782436 * MAX(Horsepower - 120,0);
run;
```

References

- Akaike, H. (1973), “Information Theory and an Extension of the Maximum Likelihood Principle,” in Petrov and Csaki, eds., Proceedings of the Second International Symposium on Information Theory, 267–281.
- Bilenas, J. (2010), “Using PROC RANK and PROC UNIVARIATE to Rank or Decile Variables”, NESUG 2010. Search in https://lexjansen.com/ or https://jonasbilenascom.wordpress.com/.
- Cleveland, W. S., Devlin, S. J., and Grosse, E. (1988), “Regression by Local Fitting,” Journal of Econometrics, 37, 87–114.
- Cleveland, W. S. and Grosse, E. (1991), “Computational Methods for Local Regression,” Statistics and Computing, 1, 47–62.
- Cohen, R.A. (SUGI 24). “An Introduction to PROC LOESS for Local Regression,” Paper 273-24.
- Eilers, P.H.C. and Marx, B.D. (1996), “Flexible Smoothing with B-Splines and Penalties,” Statistical Science, 11, 89-121 and discussion.
- Flom, P.L. (NESUG 2007) “Stopping stepwise: Why stepwise and similar selection methods are bad, and what you should use.”
- Flom, P. (2015), “Alternative methods of regression when OLS is not right,” SAS Global Forum 2015 (paper 3412-2015).
- Friedman, J. H. (1991), “Multivariate Adaptive Regression Splines,” Annals of Statistics, 19, 1–67.
- Harrell, F. (2015). “Regression Modeling Strategies: With Applications to Linear Models, Logistic Regression, and Survival Analysis (Springer Series in Statistics),” Springer.
- Harrell RCSPLINE MACRO:
  - http://biostat.mc.vanderbilt.edu/wiki/pub/Main/SasMacros/survrisk.txt
- Irwin, J.R. and McClelland, G.H. (2003), “Negative Consequences of Dichotomizing Continuous Predictor Variables”, Journal of Marketing Research, Vol. 40, No. 3 (Aug., 2003), pp. 366-371.
- Kuhfeld, W. F. and Cai, W. (2013), “Introducing the New ADAPTIVEREG Procedure for Adaptive Regression,” SAS Global Forum 2013 (paper 457-2013).
- Liu, W. and Xin, J. (2014), “Modeling Fractional Outcomes with SAS” Paper 1304, http://support.sas.com/resources/papers/proceedings14/1304-2014.pdf
- Stone, C. J. and Koo, C. Y. (1985), “Additive splines in statistics,” In Proceedings of the Statistical Computing Section ASA, pages 45-48, Washington, DC, 1985. [34, 39]
- Wicklin, R. (2017) The DO Loop Blog: Regression with restricted cubic splines in SAS. https://blogs.sas.com/content/iml/2017/04/19/restricted-cubic-splines-sas.html
- Tobias, R. and Cai, W. (2010), “Introducing PROC PLM and Postfitting Analysis for Very General Linear Models in SAS/STAT® 9.22,” SAS GLOBAL FORUM 2010.

Disclaimers

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names used in this presentation are trademarks of their respective companies.

The contents of this paper are the work of the authors and do not necessarily represent the opinions, recommendations, or practices of any company that we have worked for or are currently working for.

Splines work in all industries, not just Banking.

The crabs in this presentation are left over form the original presentation we made at SESUG 2016 October 16-18, 2016, Bethesda, Maryland