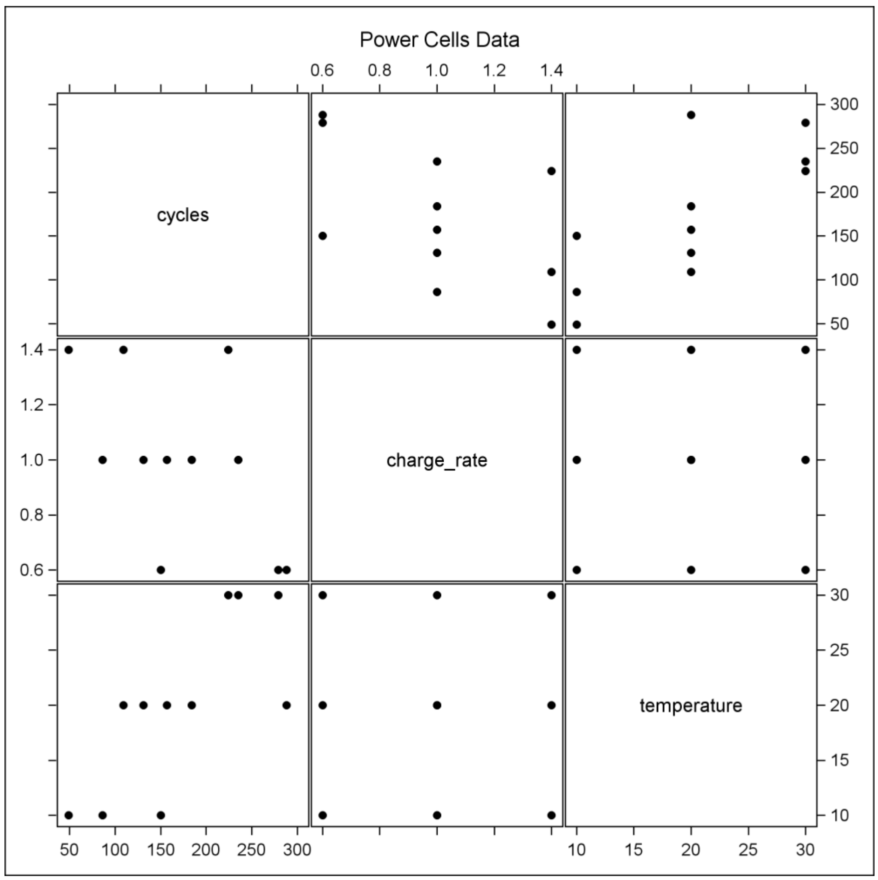
Stat 5100 Handout 3.1.1 – SAS: Alternative Predictor Variable Types

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)).

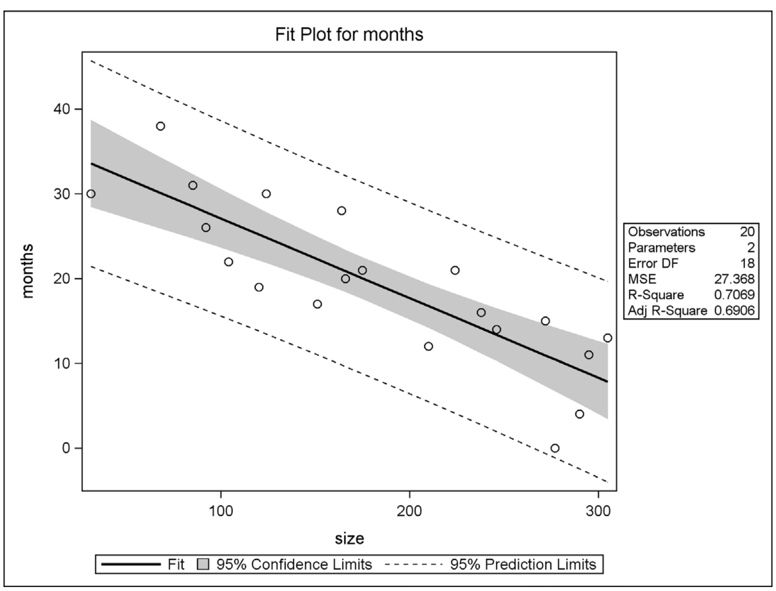
```
/* Input data -- see Table 8.2 of text */
data insurance; input months size type @@; cards;
  17  151  0      26   92  0      21  175  0      30   31  0
  22  104  0       0  277  0      12  210  0      19  120  0
   4  290  0      16  238  0      28  164  1      15  272  1
  11  295  1      38 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

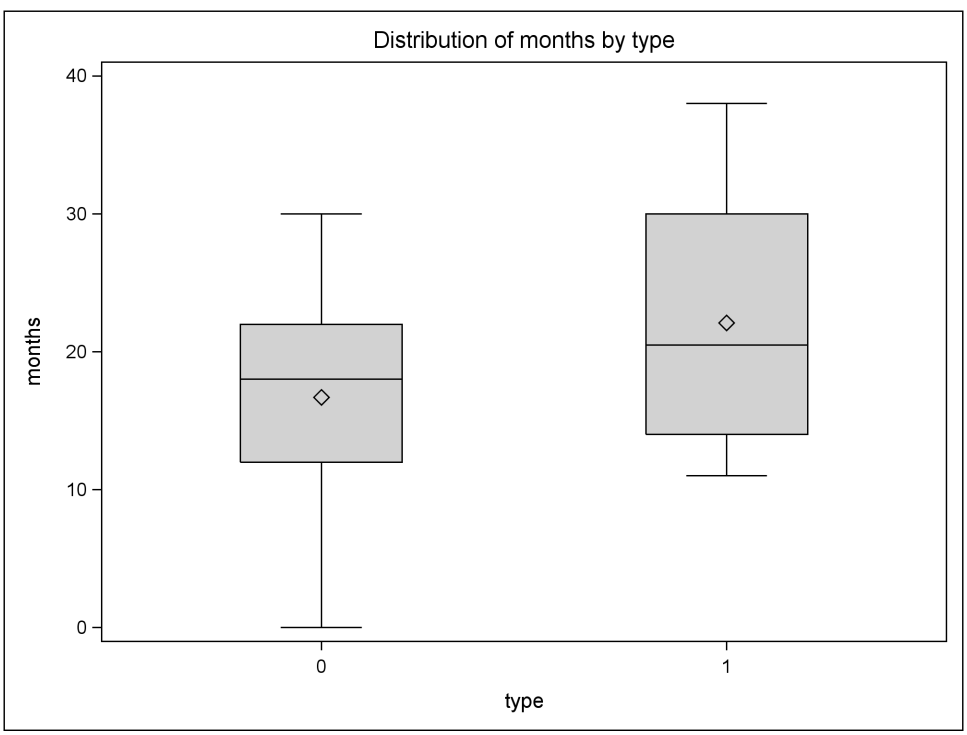


```
/* Model with only qualitative predictor */
proc reg data=insurance;
  model months = type;
  title1 'Single qualitative predictor';
run;
```

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;
```

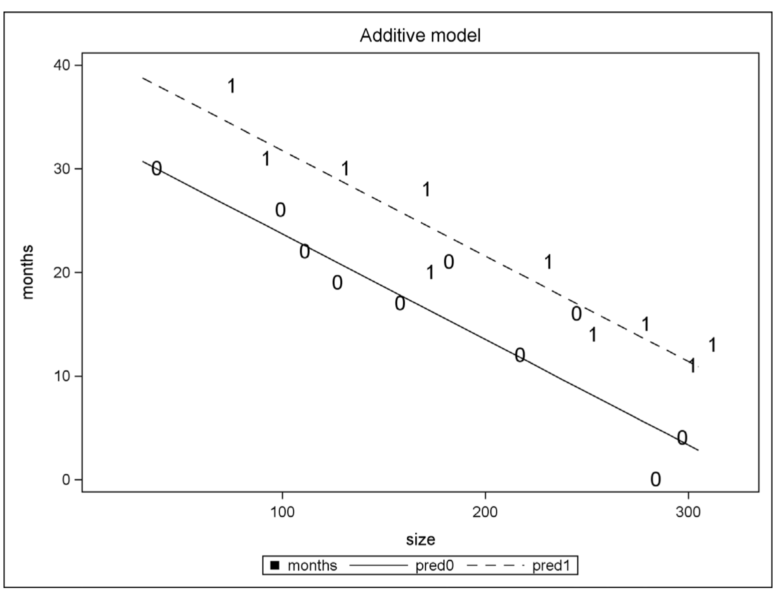


```
/* Additive model */
proc reg data=insurance;
  model months = size type;
  title1 'Additive model';
run;
```

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;
```