SAS Data Analysis
Poisson Regression

Poisson regression is for modeling count variables.

Please note: The purpose of this page is to show how to use various data analysis commands. It does not cover all aspects of the research process which researchers are expected to do. In particular, it does not cover data cleaning and checking, verification of assumptions, model diagnostics or potential follow-up analyses.

This analysis was done using SAS version 9.22.

Examples of Poisson regression

Example 1. The number of persons killed by mule or horse kicks in the Prussian army per year. von Bortkiewicz collected data from 20 volumes of *Preussischen Statistik*. These data were collected on 10 corps of the Prussian army in the late 1800s over the course of 20 years.

Example 2. A health-related researcher is studying the number of hospital visits in past 12 months by senior citizens in a community based on the characteristics of the individuals and the types of health plans under which each one is covered.

Example 3. A researcher in education is interested in the association between the number of awards earned by students at one high school and the students' performance in math and the type of program (e.g., vocational, general or academic) in which students were enrolled.

Description of the data

For the purpose of illustration, we have simulated a data set for Example 3 above: poisson_sim.sas7bdat. In this example, **num_awards** is the outcome variable and indicates the number of awards earned by students at a high school in a year, **math** is a continuous predictor variable and represents students' scores on their math final exam, and **prog** is a categorical predictor variable with three levels indicating the type of program in which the students were enrolled. It is coded as 1 = "General", 2 = "Academic" and 3 = "Vocational".

```
proc means data = poisson_sim n mean var min max;
  var num_awards math;
run;
```

```
The MEANS Procedure

Variable      Label            N            Mean         Variance
Minimum         Maximum
------------------------------------------------------------------------------
------------------
num_awards                   200       0.6300000       1.1086432
0       6.0000000
math          math score     200      52.6450000      87.7678141
33.0000000      75.0000000
```

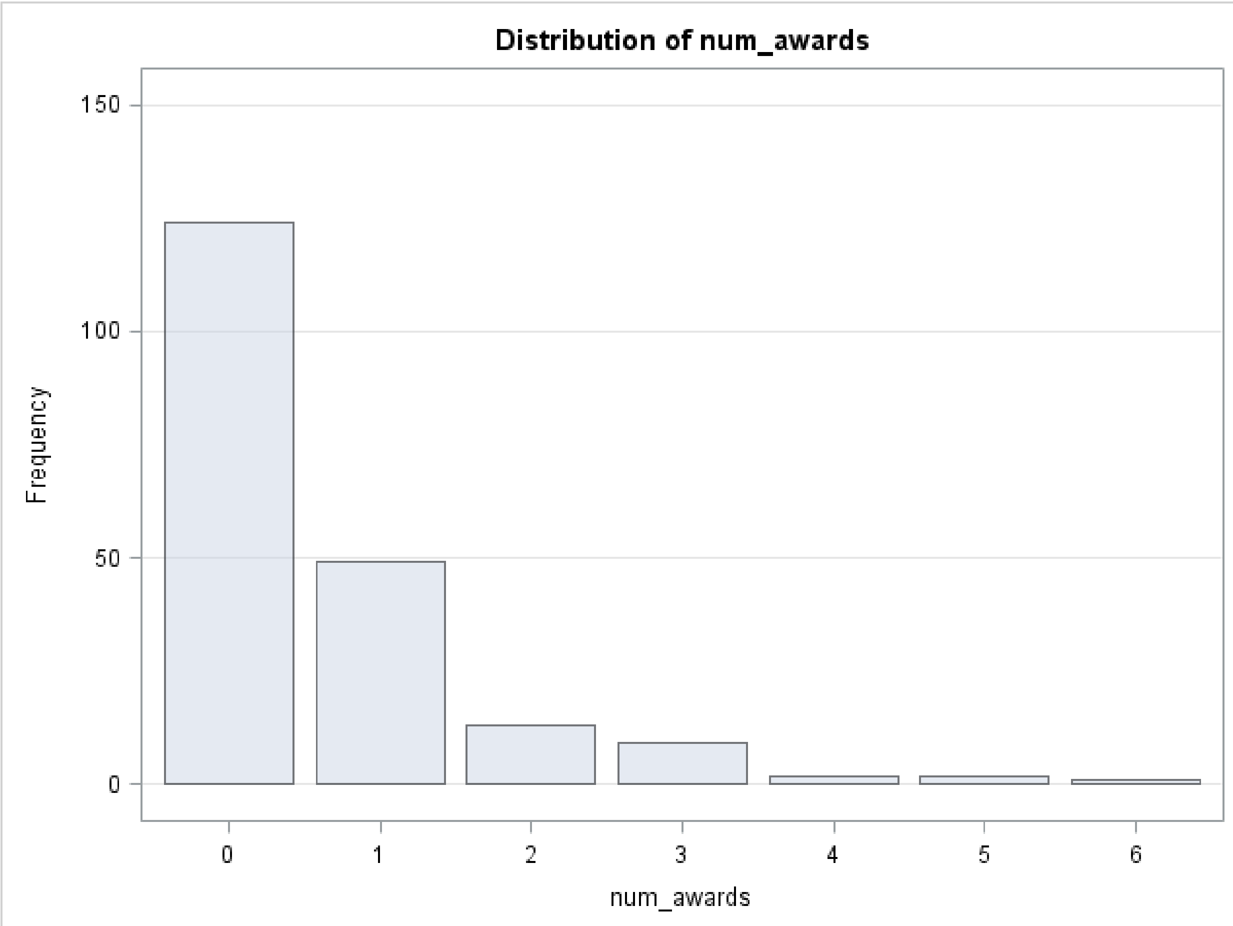

Each variable has 200 valid observations and their distributions seem quite reasonable. The *unconditional* mean and variance of our outcome variable are not extremely different. Our model assumes that these values, conditioned on the predictor variables, will be equal (or at least roughly so).

We can look at summary statistics by program type. The table below shows the mean and variance of numbers of awards by program type and seems to suggest that program type is a good candidate for predicting the number of awards, our outcome variable, because the mean value of the outcome appears to vary by **prog**. Additionally, the means and variances within each level of **prog**--the *conditional* means and variances--are similar. A frequency plot is also produced to display the distribution of the outcome variable.

```
proc means data = poisson_sim mean var;
  class prog;
  var num_awards;
run;
```

```
The MEANS Procedure

          Analysis Variable : num_awards

     type of      N
     program    Obs            Mean        Variance
---------------------------------------------------
           1     45       0.2000000       0.1636364

           2    105       1.0000000       1.6346154

           3     50       0.2400000       0.2677551
---------------------------------------------------
```

```
proc freq data=poisson_sim;
tables num_awards / plots=freqplot;
run;
```




```
proc freq data = poisson_sim;
  tables prog;
run;
The FREQ Procedure

                     type of program

                                 Cumulative    Cumulative
prog    Frequency     Percent     Frequency      Percent
---------------------------------------------------------
   1          45       22.50            45        22.50
   2         105       52.50           150        75.00
   3          50       25.00           200       100.00
```

Analysis methods you might consider

Below is a list of some analysis methods you may have encountered. Some of the methods listed are quite reasonable, while others have either fallen out of favor or have limitations.

- Poisson regression - Poisson regression is often used for modeling count data. It has a number of extensions useful for count models.

- Negative binomial regression - Negative binomial regression can be used for over-dispersed count data, that is when the conditional variance exceeds the conditional mean. It can be considered as a generalization of Poisson regression since it has the same mean structure as Poisson regression and it has an extra parameter to model the over-dispersion. If the conditional distribution of the outcome variable is over-dispersed, the confidence intervals for Negative binomial regression are likely to be narrower as compared to those from a Poisson regession.
- Zero-inflated regression model - Zero-inflated models attempt to account for excess zeros. In other words, two kinds of zeros are thought to exist in the data, "true zeros" and "excess zeros". Zero-inflated models estimate two equations simultaneously, one for the count model and one for the excess zeros.
- OLS regression - Count outcome variables are sometimes log-transformed and analyzed using OLS regression. Many issues arise with this approach, including loss of data due to undefined values generated by taking the log of zero (which is undefined) and biased estimates.

Poisson regression analysis

At this point, we are ready to perform our Poisson model analysis. **Proc genmod** is usually used for Poisson regression analysis in SAS.

On the **class** statement we list the variable **prog**, since **prog** is a categorical variable. We use the global option **param = glm** so we can save the model using the **store** statement for future post estimations. The **type3** option in the model statement is used to get the multi-degree-of-freedom test of the categorical variables listed on the **class** statement, and the **dist = poisson** option is used to indicate that a Poisson distribution should be used. Statement "store" allows us to store the parameter estimates to a data set, which we call p1, so we can perform post estimation without rerunning the model.

```
proc genmod data = poisson_sim;
  class prog  /param=glm;
  model num_awards = prog math / type3 dist=poisson;
  store p1;
run;
```

```
The GENMOD Procedure

          Model Information

Data Set                 WORK.POISSON_SIM
Distribution                      Poisson
Link Function                         Log
Dependent Variable             num_awards


Number of Observations Read         200
Number of Observations Used         200
```

Class Level Information

Class	Levels	Values
prog	3	1 2 3

Criteria For Assessing Goodness Of Fit

Criterion	DF	Value	Value/DF
Deviance	196	189.4496	0.9666
Scaled Deviance	196	189.4496	0.9666
Pearson Chi-Square	196	212.1437	1.0824
Scaled Pearson X2	196	212.1437	1.0824
Log Likelihood		-135.1052	
Full Log Likelihood		-182.7523	
AIC (smaller is better)		373.5045	
AICC (smaller is better)		373.7096	
BIC (smaller is better)		386.6978	

Algorithm converged.

Analysis Of Maximum Likelihood Parameter Estimates

Parameter		DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept		1	-4.8773	0.6282	-6.1085	-3.6461	60.28	<.0001
prog	1	1	-0.3698	0.4411	-1.2343	0.4947	0.70	0.4018
prog	2	1	0.7140	0.3200	0.0868	1.3413	4.98	0.0257
prog	3	0	0.0000	0.0000	0.0000	0.0000	.	.
math		1	0.0702	0.0106	0.0494	0.0909	43.81	<.0001
Scale		0	1.0000	0.0000	1.0000	1.0000		

NOTE: The scale parameter was held fixed.

LR Statistics For Type 3 Analysis

Source	DF	Chi-Square	Pr > ChiSq
prog	2	14.57	0.0007
math	1	45.01	<.0001

- The output begins with the basic model information and then provides a list of goodness-of-fit statistics including the log likelihood, AIC, and BIC.
- Next you will find the Poisson regression coefficients for each of the variables along with standard errors, Wald Chi-Square statistics and intervals, and p-values for the coefficients. The coefficient for **math** is .07. This means that the expected increase in log count for a one-unit increase in **math** is .07. For our three-level categorical predictor **prog**, the model presents coefficients relating levels 1 and 2 to level 3. The indicator variable **prog(2)** is the expected difference in log count between group 2 (**prog**=2) and the reference group (**prog**=3). So the expected log count for level 2 of **prog** is 0.714 higher than the expected log count for level 3 of **prog**. Similarly the expected log count for level 1 of **prog** is 0.3698 lower than the expected log count for level 3.
- To determine if **prog** itself, overall, is statistically significant, we can look at the Type 3 table in the outcome that includes the two degrees-of-freedom test of this variable. This is testing the null hypothesis that both **prog** estimates (level 1 vs. level 3 and level 2 vs. level 3) are equal to zero. We see there that **prog** is a statistically significant predictor.

To help assess the fit of the model, we can use the goodness-of-fit chi-squared test. This assumes the deviance follows a chi-square distribution with degrees of freedom equal to the model residual. From the first line of our Goodness of Fit output, we can see these values are 189.4495 and 196.

```
data pvalue;
  df = 196; chisq = 189.4495;
  pvalue = 1 - probchi(chisq, df);
run;
proc print data = pvalue noobs;
run;
```

```
 df      chisq      pvalue

196    189.450    0.61823
```

This is not a test of the model coefficients (which we saw in the header information), but a test of the model form: Does the poisson model form fit our data? We conclude that the model fits reasonably well because the goodness-of-fit chi-squared test is not statistically significant. If the test had been statistically significant, it would indicate that the data do not fit the model well. In that situation, we may try to determine if there are omitted predictor variables, if our linearity assumption holds and/or if there is an issue of over-dispersion.

Cameron and Trivedi (2009) recommend using robust standard errors for the parameter estimates to control for mild violation of the distribution assumption that the variance equals the mean. In SAS, we can do this by running **proc genmod** with the **repeated** statement in order to obtain robust standard errors for the Poisson regression coefficients.

```
proc genmod data = poisson_sim;
  class prog id /param=glm;
  model num_awards = prog math /dist=poisson;
  repeated subject=id;
```

```
run;

             GEE Model Information

Correlation Structure               Independent
Subject Effect                  id (200 levels)
Number of Clusters                          200
Correlation Matrix Dimension                  1
Maximum Cluster Size                          1
Minimum Cluster Size                          1


Algorithm converged.

  GEE Fit Criteria

QIC          256.8581
QICu         257.6478


              Analysis Of GEE Parameter Estimates
               Empirical Standard Error Estimates

                     Standard   95% Confidence
Parameter   Estimate    Error       Limits            Z Pr > |Z|

Intercept    -4.8773   0.6297  -6.1116  -3.6430   -7.74   <.0001
prog      1  -0.3698   0.4004  -1.1546   0.4150   -0.92   0.3557
prog      2   0.7140   0.2986   0.1287   1.2994    2.39   0.0168
prog      3   0.0000   0.0000   0.0000   0.0000     .       .
math          0.0702   0.0104   0.0497   0.0906    6.72   <.0001
```

We can see that our estimates are unchanged, but our standard errors are slightly different.

We have the model stored in a data set called **p1**. Using **proc plm**, we can request many different post estimation tasks. For example, we might want to displayed the results as incident rate ratios (IRR). We can do so with a **data** step after using **proc plm** to create a dataset of our model estimates.

```
ods output ParameterEstimates = est;
proc plm source = p1;
  show parameters;
run;

data est_exp;
  set est;
  irr = exp(estimate);
  if parameter ^="Intercept";
run;
proc print data = est_exp;
run;
Obs    Parameter            prog    Estimate      StdErr      irr

 1     type of program 1     1       -0.3698      0.4411    0.69087
 2     type of program 2     2        0.7140      0.3200    2.04225
```

```
3     type of program 3     3             0            .     1.00000
4     math score            _       0.07015      0.01060     1.07267
```

The output above indicates that the incident rate for **prog**=2 is 2.04 times the incident rate for the reference group (**prog**=3). Likewise, the incident rate for **prog**=1 is 0.69 times the incident rate for the reference group holding the other variables constant. The percent change in the incident rate of **num_awards** is by 7% for every unit increase in **math**.

Recall the form of our model equation:

$$\log(\text{num\_awards}) = \text{Intercept} + b_1(\text{prog}=1) + b_2(\text{prog}=2) + b_3\text{math}.$$

This implies:

$$\text{num\_awards} = \exp(\text{Intercept} + b_1(\text{prog}=1) + b_2(\text{prog}=2)+ b_3\text{math}) = \exp(\text{Intercept}) * \exp(b_1(\text{prog}=1)) * \exp(b_2(\text{prog}=2)) * \exp(b_3\text{math})$$

The coefficients have an *additive* effect in the log(y) scale and the IRR have a *multiplicative* effect in the y scale.

For additional information on the various metrics in which the results can be presented, and the interpretation of such, please see *Regression Models for Categorical Dependent Variables Using Stata, Second Edition* by J. Scott Long and Jeremy Freese (2006).

Below we use **lsmeans** statements in **proc plm** to calculate the predicted number of events at each level of **prog**, holding all other variables (in this example, **math**) in the model at their means. We use the "**ilink**" option (for inverse link) to get the predicted means (predicted count) in addition to the linear predictions.

```
proc plm source = p1;
  lsmeans prog /ilink cl;
run;
```

prog Least Squares Means

type of program	Estimate	Standard Error	z Value	Pr > |z|	Alpha	Lower	Upper
1	-1.5540	0.3335	-4.66	<.0001	0.05	-2.2076	-0.9003
2	-0.4701	0.1381	-3.40	0.0007	0.05	-0.7407	-0.1995
3	-1.1841	0.2887	-4.10	<.0001	0.05	-1.7499	-0.6183

prog Least Squares Means

type of program	Mean	Standard Error of Mean	Lower Mean	Upper Mean

```
1              0.2114     0.07050      0.1100      0.4064
2              0.6249     0.08628      0.4768      0.8191
3              0.3060     0.08834      0.1738      0.5388
```

The first block of output above shows the predicted log count. The second block shows predicted number of events in the "mean" column.

In the output above, we see that the predicted number of events for level 1 of **prog** is about .21, holding **math** at its mean. The predicted number of events for level 2 of **prog** is higher at .62, and the predicted number of events for level 3 of **prog** is about .31. Note that the predicted count of level 1 of **prog** is (.2114/.3060) = 0.6908 times the predicted count for level 3 of **prog**. This matches what we saw in the IRR output table.

Below we will obtain the averaged predicted counts for values of **math** that range from 35 to 75 in increments of 10, using a data step and the **score** statement of **proc plm**.

```
data toscore;
  set poisson_sim;
  do math_cat = 35 to 75 by 10;
     math = math_cat;
     output;
  end;
run;
proc plm source=p1;
  score data = toscore out=math /ilink;
run;
proc means data = math mean;
  class math_cat;
  var predicted;
run;
```

```
                   N
    math_cat     Obs           Mean
------------------------------------
          35     200      0.1311326
          45     200      0.2644714
          55     200      0.5333923
          65     200      1.0757584
          75     200      2.1696153
```

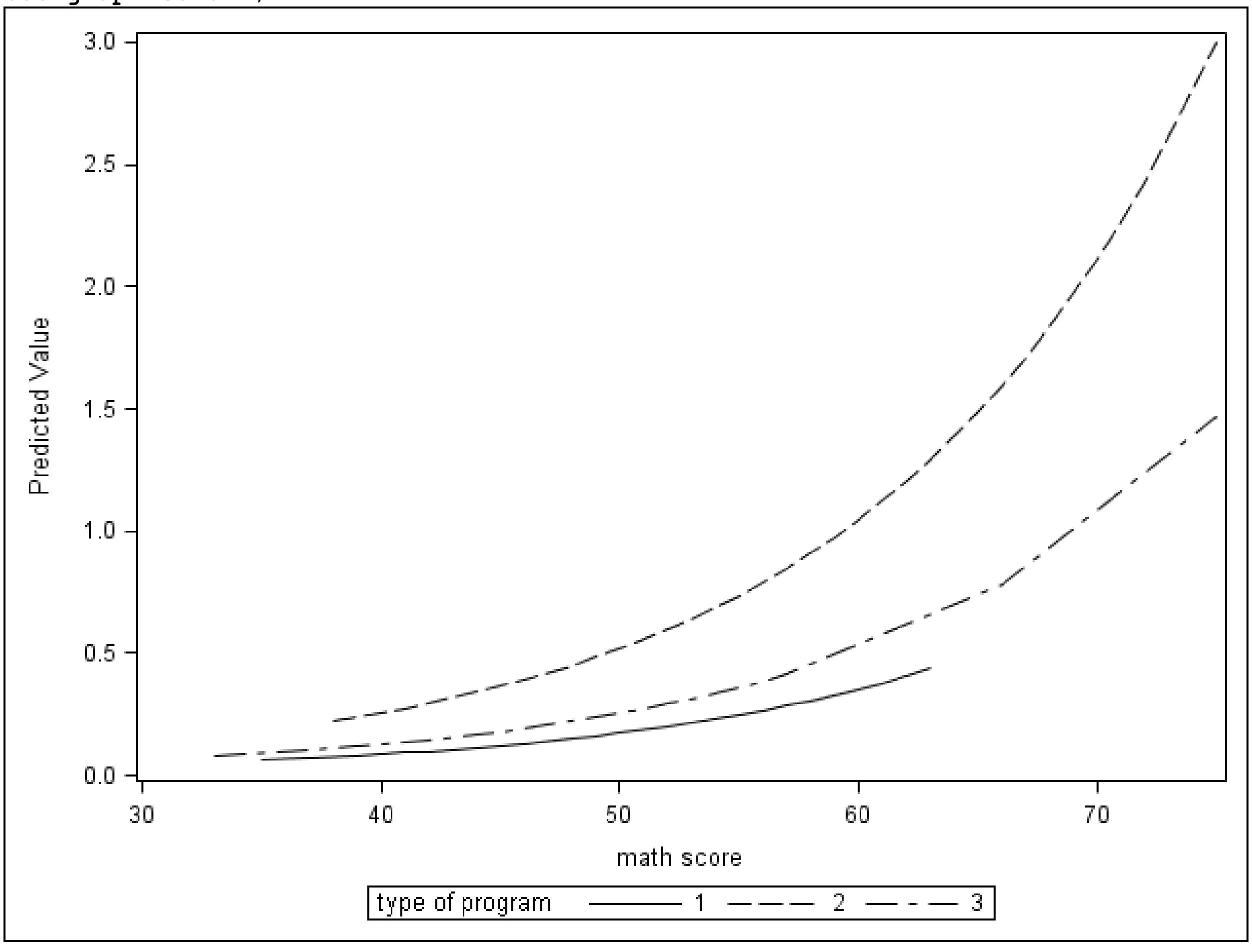
The table above shows that with **prog** at its observed values and **math** held at 35 for all observations, the average predicted count (or average number of awards) is about .13; when **math** = 75, the average predicted count is about 2.17.

If we compare the predicted counts at math = 35 and math = 45, we can see that the ratio is (.2644714/.1311326) = 2.017. This matches the IRR of 1.0727 for a 10 unit change: 1.0727^10 = 2.017.

You can graph the predicted number of events using **proc plm** and **proc sgplot** below.

```
ods graphics on;
ods html style=journal;
```

```
proc plm source=p1;
  score data = poisson_sim out=pred /ilink;
run;
proc sort data = pred;
  by prog math;
run;
proc sgplot data = pred;
  series x = math y = predicted /group=prog;
run;
ods graphics off;
```



Things to consider

- When there seems to be an issue of dispersion, we should first check if our model is appropriately specified, such as omitted variables and functional forms. For example, if we omitted the predictor variable **prog** in the example above, our model would seem to have a problem with over-dispersion. In other words, a mis-specified model could present a symptom like an over-dispersion problem.
- Assuming that the model is correctly specified, you may want to check for overdispersion. There are several tests including the likelihood ratio test of over-dispersion parameter alpha by running the same regression model using negative binomial distribution.

- One common cause of over-dispersion is excess zeros, which in turn are generated by an additional data generating process. In this situation, a zero-inflated model should be considered.
- If the data-generating process does not allow for any 0s (such as the number of days spent in the hospital), then a zero-truncated model may be more appropriate.
- The outcome variable in a Poisson regression cannot have negative numbers.
- Poisson regression is estimated via maximum likelihood estimation. It usually requires a large sample size.

SAS Data Analysis
Multivariate Regression Analysis

As the name implies, multivariate regression is a technique that estimates a single regression model with multiple outcome variables and one or more predictor variables.

Please Note: The purpose of this page is to show how to use various data analysis commands. It does not cover all aspects of the research process which researchers are expected to do. In particular, it does not cover data cleaning and checking, verification of assumptions, model diagnostics and potential follow-up analyses.

Examples of multivariate regression analysis

Example 1. A researcher has collected data on three psychological variables, four academic variables (standardized test scores), and the type of educational program the student is in for 600 high school students. She is interested in how the set of psychological variables relate to the academic variables and gender. In particular, the researcher is interested in how many dimensions are necessary to understand the association between the two sets of variables.

Example 2. A doctor has collected data on cholesterol, blood pressure and weight. She also collected data on the eating habits of the subjects (e.g., how many ounces of red meat, fish, dairy products, and chocolate consumed per week). She wants to investigate the relationship between the three measures of health and eating habits.

Example 3. A researcher is interested in determining what factors influence the health African Violet plants. She collects data on the average leaf diameter, the mass of the root ball, and the average diameter of the blooms, as well as how long the plant has been in the current container. For predictor variables, she measures several elements in the soil, in addition to the amount of light and water each plant receives.

Description of the data

Let's pursue Example 1 from above. We have a hypothetical dataset, mvreg.sas7bdat, with 600 observations on seven variables. The psychological variables are **locus of control**, **self-concept** and **motivation**. The academic variables are standardized tests scores in **reading**, **writing**, and **science**, as well as a categorical variable giving the type of program the student is in (general, academic, or vocational). In our example the dataset **mvreg.sas7bdat** is saved in a library called **data**.

Let's look at the data (note that there are no missing values in this data set).

```
proc means data = data.mvreg;
  vars locus_of_control self_concept motivation read write science;
run;
```

```
                                       The MEANS Procedure

 Variable            Label      N            Mean         Std Dev
Minimum         Maximum
-----------------------------------------------------------------------------
--------------------
 LOCUS_OF_CONTROL             600       0.0965333       0.6702799      -
1.9959567       2.2055113
 SELF_CONCEPT                 600       0.0049167       0.7055125      -
2.5327499       2.0935633
 MOTIVATION                   600       0.0038979       0.8224000      -
2.7466691       2.5837522
 READ                         600      51.9018333      10.1029831
24.6200066      80.5864944
 WRITE                        600      52.3848332       9.7264550
20.0688801      83.9348221
 SCIENCE                      600      51.7633331       9.7061791
21.9895325      80.3694153
-----------------------------------------------------------------------------
--------------------
```

```
proc freq data = data.mvreg;
  table prog;
run;
```

```
                    The FREQ Procedure

                       program type

                                 Cumulative    Cumulative
PROG    Frequency     Percent     Frequency      Percent
---------------------------------------------------------
   1         138       23.00           138        23.00
   2         271       45.17           409        68.17
   3         191       31.83           600       100.00
```

```
proc corr data = data.mvreg nosimple;
  var locus_of_control self_concept motivation;
run;
```

```
                        The CORR Procedure

3  Variables:    LOCUS_OF_CONTROL SELF_CONCEPT     MOTIVATION


            Pearson Correlation Coefficients, N = 600
                    Prob > |r| under H0: Rho=0

                         LOCUS_OF_        SELF_
                           CONTROL       CONCEPT      MOTIVATION

   LOCUS_OF_CONTROL         1.00000       0.17119         0.24513
                                           <.0001          <.0001
```

```
   SELF_CONCEPT              0.17119       1.00000         0.28857
                              <.0001                        <.0001

   MOTIVATION                0.24513       0.28857         1.00000
                              <.0001        <.0001
```

```
proc corr data = data.mvreg nosimple;
  var read write science;
run;
```

```
                The CORR Procedure

    3  Variables:    READ     WRITE    SCIENCE


   Pearson Correlation Coefficients, N = 600
           Prob > |r| under H0: Rho=0

                   READ          WRITE       SCIENCE

READ           1.00000       0.62859       0.69069
                              <.0001        <.0001

WRITE          0.62859       1.00000       0.56915
                <.0001                      <.0001

SCIENCE        0.69069       0.56915       1.00000
                <.0001        <.0001
```

Analysis methods you might consider

Below is a list of some analysis methods you may have encountered. Some of the methods listed are quite reasonable while others have either fallen out of favor or have limitations.

- Multivariate multiple regression, the focus of this page.
- Separate OLS Regressions - You could analyze these data using separate OLS regression analyses for each outcome variable. The individual coefficients, as well as their standard errors, will be the same as those produced by the multivariate regression. However, the OLS regressions will not produce multivariate results, nor will they allow for testing of coefficients across equations.
- Canonical correlation analysis might be feasible if don't want to consider one set of variables as outcome variables and the other set as predictor variables.

Multivariate regression analysis

Technically speaking, we will be conducting a multivariate multiple regression. This regression is "multivariate" because there is more than one outcome variable. It is a "multiple" regression because there is more than one predictor variable. Of course, you can conduct a multivariate regression with only one predictor variable, although that is rare in practice.

To conduct a multivariate regression in SAS, you can use **proc glm**, which is the same procedure that is often used to perform ANOVA or OLS regression. The syntax for estimating a multivariate regression is similar to running a model with a single outcome, the primary difference is the use of the **manova** statement so that the output includes the multivariate statistics. The f- and p-values for four multivariate criterion are given, including Wilks' lambda, Lawley-Hotelling trace, Pillai's trace, and Roy's largest root. By specifying **h=_ALL_** on the **manova** statement, we indicate that we would like multivariate statistics for all of the predictor variables in the model, if we were only interested in the multivariate statistics for some variables, we could replace **_ALL_** with the name of a variable (e.g. **h=read**).

```
proc glm data = data.mvreg;
  class prog;
  model locus_of_control self_concept motivation
      = read write science prog / solution ss3;
  manova h=_ALL_;
run;
quit;
```

The SAS output for multivariate regression can be very long, especially if the model has many outcome variables. The output from our example has four parts: one for each of the three outcome variables, and the fourth from the **manova** statement. Below we will discuss the output in sections.

```
                                            The GLM Procedure

                                         Class Level Information

                                      Class         Levels    Values

                                      PROG               3    1 2 3


                                 Number of Observations Read         600
                                 Number of Observations Used         600
```

Above we see that the class variable **prog** has three levels. Just below the class level information, we see the number of observations read form the data and the number of observations used in the analysis. If the variables used in the analysis contained missing values the number of observations used would be smaller than the number of observations read.

```
Dependent Variable: LOCUS_OF_CONTROL

                                                    Sum of
       Source                      DF             Squares     Mean Square    F
Value    Pr > F

       Model                        5      50.2595509      10.0519102
27.28    <.0001

       Error                      594     218.8562365       0.3684448

       Corrected Total            599     269.1157874
```

```
                  R-Square     Coeff Var      Root MSE    LOCUS_OF_CONTROL
Mean

                  0.186758      628.7948      0.606997
0.096533


       Source                     DF     Type III SS     Mean Square    F
Value    Pr > F

       READ                        1      4.16815963      4.16815963
11.31    0.0008
       WRITE                       1      4.72524304      4.72524304
12.82    0.0004
       SCIENCE                     1      0.92248638      0.92248638
2.50    0.1141
       PROG                        2      5.02961991      2.51480995
6.83    0.0012
```

- The dependent variable, **locus_of_control**, is listed at the top of the output above.
- The ANOVA table for **locus_of_control** gives the sum of squares and mean square for both the model and error term. The model for **locus_of_control** is statistically significant with a p-value of less than 0.0001.
- Below the ANOVA table we see the R-square value of 0.187, indicating that 18.7% of variance in **locus_of_control** is explained by the model.
- The final table shown above gives the predictor variables in the model, along with the type III sum of squares for each variable. We can see that **read**, **write**, and **prog** are statistically significant. Note that because **prog** is a class variable with three levels, it uses 2 degrees of freedom (shown in the column labeled DF).

```
                                                          Standard
              Parameter           Estimate                   Error    t Value    Pr
> |t|

              Intercept       -1.373094234 B      0.16259260      -8.44
<.0001
              READ             0.012504619        0.00371779       3.36
0.0008
              WRITE            0.012145048        0.00339136       3.58
0.0004
              SCIENCE          0.005761477        0.00364116       1.58
0.1141
              PROG      1     -0.251670509 B      0.06846988      -3.68
0.0003
              PROG      2     -0.123875431 B      0.05760714      -2.15
0.0319
              PROG      3      0.000000000 B       .                 .
.

NOTE: The X'X matrix has been found to be singular, and a generalized inverse
was used to solve
```

```
      the normal equations.  Terms whose estimates are followed by the letter
'B' are not
      uniquely estimable.
```

- The table above gives the parameter estimates, their standard errors, t-value, and associated p-value. The coefficients are interpreted in the same manner as OLS regression coefficients. For example, a one unit increase in **read** is associated with a 0.013 increase in the predicted value of **locus_of_control**.
- The note shown above is SAS's way of telling us that it could not include the terms for all three levels of **prog** and the intercept in the model. Instead it has included the intercept and terms for **prog**=1 and **prog**=2, leaving **prog**=3 as the reference group.

The output for the first outcome variable (**locus_of_control**) is followed by similar output for each additional outcome (**self_concept** and **motivation**). This output is shown below, but we will not discuss it further, instead we will move on to the multivariate output.

```
                                                The GLM Procedure

Dependent Variable: SELF_CONCEPT

                                                       Sum of
       Source                      DF         Squares     Mean Square    F
Value    Pr > F

       Model                        5      16.1107053       3.2221411
6.79   <.0001

       Error                      594     282.0402900       0.4748153

       Corrected Total            599     298.1509953


                    R-Square     Coeff Var      Root MSE    SELF_CONCEPT Mean

                    0.054035      14014.91      0.689068             0.004917


       Source                      DF     Type III SS     Mean Square    F
Value    Pr > F

       READ                         1      0.04557875      0.04557875
0.10   0.7568
       WRITE                        1      0.59051932      0.59051932
1.24   0.2652
       SCIENCE                      1      0.78237876      0.78237876
1.65   0.1998
       PROG                         2     14.21838537      7.10919268
14.97   <.0001


                                                            Standard
              Parameter           Estimate               Error    t Value    Pr
> |t|
```

Intercept		0.0510179965 B	0.18457670	0.28	0.7823
READ		0.0013076138	0.00422047	0.31	0.7568
WRITE		-.0042934282	0.00384990	-1.12	0.2652
SCIENCE		0.0053059405	0.00413348	1.28	0.1998
PROG	1	-.4233591913 B	0.07772768	-5.45	<.0001
PROG	2	-.1468757972 B	0.06539618	-2.25	0.0251
PROG	3	0.0000000000 B	.	.	.

NOTE: The X'X matrix has been found to be singular, and a generalized inverse was used to solve the normal equations. Terms whose estimates are followed by the letter 'B' are not uniquely estimable.

The GLM Procedure

Dependent Variable: MOTIVATION

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	60.7672827	12.1534565	20.96	<.0001
Error	594	344.3614302	0.5797330		
Corrected Total	599	405.1287128			

R-Square	Coeff Var	Root MSE	MOTIVATION Mean
0.149995	19533.65	0.761402	0.003898

Source	DF	Type III SS	Mean Square	F Value	Pr > F
READ	1	2.49445035	2.49445035	4.30	0.0385
WRITE	1	9.85052717	9.85052717	16.99	<.0001
SCIENCE	1	2.25173630	2.25173630	3.88	0.0492
PROG	2	30.18084209	15.09042104	26.03	<.0001

Standard

```
               Parameter            Estimate              Error    t Value    Pr
> |t|

               Intercept        -.6911458885 B      0.20395228      -3.39
0.0007
               READ             0.0096735465        0.00466350       2.07
0.0385
               WRITE            0.0175354486        0.00425404       4.12
<.0001
               SCIENCE          -.0090014528        0.00456739      -1.97
0.0492
               PROG      1      -.6196960376 B      0.08588699      -7.22
<.0001
               PROG      2      -.2593666472 B      0.07226102      -3.59
0.0004
               PROG      3      0.0000000000 B       .                  .
.

NOTE: The X'X matrix has been found to be singular, and a generalized inverse
was used to solve
      the normal equations.  Terms whose estimates are followed by the letter
'B' are not
      uniquely estimable.
```

The final section of output for our model is output for the multivariate tests of the model.

```
                                             The GLM Procedure
                                     Multivariate Analysis of Variance

                    Characteristic Roots and Vectors of: E Inverse * H, where
                                     H = Type III SSCP Matrix for READ
                                           E = Error SSCP Matrix

          Characteristic               Characteristic Vector  V'EV=1
                    Root    Percent    LOCUS_OF_CONTROL    SELF_CONCEPT
MOTIVATION

              0.02414400     100.00          0.05725523     -0.00912678
0.02560444
              0.00000000       0.00         -0.00704393      0.05979895
0.00102214
              0.00000000       0.00         -0.03710958     -0.01295454
0.04972124


     MANOVA Test Criteria and Exact F Statistics for the Hypothesis of No
Overall READ Effect
                                     H = Type III SSCP Matrix for READ
                                           E = Error SSCP Matrix

                                           S=1    M=0.5    N=295

         Statistic                        Value    F Value    Num DF    Den
DF    Pr > F
```

```
         Wilks' Lambda               0.97642519       4.76          3
592    0.0027
         Pillai's Trace              0.02357481       4.76          3
592    0.0027
         Hotelling-Lawley Trace      0.02414400       4.76          3
592    0.0027
         Roy's Greatest Root         0.02414400       4.76          3
592    0.0027
```

- The second table shown above gives the tests for the overall effect of **read**. These results indicate that **read** is statistically significant regardless of what type of multivariate criteria is used (i.e., all of the p-values are less than 0.01).

SAS prints similar output for each of the predictor variables in the model (in this case **write**, **science**, and **prog**), this output is shown below, but we will not discuss it further. Instead we will move on to additional tests.

```
                    Characteristic Roots and Vectors of: E Inverse * H, where
                                H = Type III SSCP Matrix for WRITE
                                      E = Error SSCP Matrix

          Characteristic               Characteristic Vector  V'EV=1
                    Root    Percent    LOCUS_OF_CONTROL    SELF_CONCEPT
MOTIVATION

              0.05552705     100.00          0.03976623     -0.02762931
0.04077279
              0.00000000       0.00          0.00235865      0.05460081
0.01173502
              0.00000000       0.00          0.05583890      0.00907776     -
0.03645138


    MANOVA Test Criteria and Exact F Statistics for the Hypothesis of No
Overall WRITE Effect
                               H = Type III SSCP Matrix for WRITE
                                      E = Error SSCP Matrix

                                      S=1    M=0.5    N=295

         Statistic                        Value    F Value    Num DF    Den
DF    Pr > F

         Wilks' Lambda               0.94739400      10.96          3
592    <.0001
         Pillai's Trace              0.05260600      10.96          3
592    <.0001
         Hotelling-Lawley Trace      0.05552705      10.96          3
592    <.0001
         Roy's Greatest Root         0.05552705      10.96          3
592    <.0001
```

Multivariate Analysis of Variance

Characteristic Roots and Vectors of: E Inverse * H, where
H = Type III SSCP Matrix for SCIENCE
E = Error SSCP Matrix

Characteristic Root	Percent	Characteristic Vector V'EV=1 LOCUS_OF_CONTROL	SELF_CONCEPT	MOTIVATION
0.01687455	100.00	0.03609681	0.03206920	-0.04456052
0.00000000	0.00	-0.02316137	0.05234944	0.01603289
0.00000000	0.00	0.05353009	-0.00762467	0.02976812

MANOVA Test Criteria and Exact F Statistics for the Hypothesis of No Overall SCIENCE Effect
H = Type III SSCP Matrix for SCIENCE
E = Error SSCP Matrix

S=1 M=0.5 N=295

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.98340548	3.33	3	592	0.0193
Pillai's Trace	0.01659452	3.33	3	592	0.0193
Hotelling-Lawley Trace	0.01687455	3.33	3	592	0.0193
Roy's Greatest Root	0.01687455	3.33	3	592	0.0193

Characteristic Roots and Vectors of: E Inverse * H, where
H = Type III SSCP Matrix for PROG
E = Error SSCP Matrix

Characteristic Root	Percent	Characteristic Vector V'EV=1 LOCUS_OF_CONTROL	SELF_CONCEPT	MOTIVATION
0.12087752	99.34	0.01903925	0.02549291	0.03813193
0.00080748	0.66	0.04668032	-0.04866125	0.01435613
0.00000000	0.00	0.04651187	0.02844692	-0.03832351

Multivariate Analysis of Variance

```
      MANOVA Test Criteria and F Approximations for the Hypothesis of No
Overall PROG Effect
                                H = Type III SSCP Matrix for PROG
                                      E = Error SSCP Matrix

                                       S=2    M=0    N=295

         Statistic                        Value    F Value    Num DF    Den
DF    Pr > F

         Wilks' Lambda               0.89143832      11.67         6
1184    <.0001
         Pillai's Trace              0.10864869      11.35         6
1186    <.0001
         Hotelling-Lawley Trace      0.12168500      12.00         6
787.56    <.0001
         Roy's Greatest Root         0.12087752      23.89         3
593    <.0001

                  NOTE: F Statistic for Roy's Greatest Root is an upper
bound.
                          NOTE: F Statistic for Wilks' Lambda is exact.
```

As mentioned above, if you ran a separate regression for each outcome variable, you would get exactly the same coefficients, standard errors, t- and p-values, and confidence intervals as shown above. So why conduct a multivariate regression? One of the advantages is that you can conduct tests of the coefficients across the different models. Below we show a few of the hypothesis tests you can perform.

For the first test, the null hypothesis is that the coefficient for **prog**=1 is equal to the coefficient for **prog**=2 for each dependent variable separately. An alternative way to state this hypothesis is that the difference between the two coefficients (i.e., **prog**=1 - **prog**=2) is equal to 0. The **estimate** statement can be used to perform this test. The text between the apostrophes (i.e., ') is a label for the output. Next we list the variable name (**prog**) followed by a series of numbers, one for each level of **prog** in order, these are the values by which the coefficients will be multiplied to perform the test. To estimate the difference between the coefficient for **prog**=1 and **prog**=2 we multiply the coefficient for **prog**=1 by 1, and the coefficient for **prog**=2 by -1, **prog**=3 is not involved in this test, so we multiply it by 0.

```
proc glm data = data.mvreg;
  class prog;
  model locus_of_control self_concept motivation
    = read write science prog / solution ss3;
  manova h= _ALL_ ;
  estimate 'prog 1 vs. prog 2' prog 1 -1 0;
run;
quit;
```

The output produced by this model is similar to the output for the previous model, except that it contains additional output associated with the use of the **estimate** statement. To save space, we will only show the additional output.

```
Dependent Variable: LOCUS_OF_CONTROL

                                                              Standard
           Parameter                     Estimate                Error    t Value
Pr > |t|

           prog 1 vs. prog 2       -0.12779508      0.06395501      -2.00
0.0462


Dependent Variable: SELF_CONCEPT
                                                              Standard
           Parameter                     Estimate                Error    t Value
Pr > |t|

           prog 1 vs. prog 2       -0.27648339      0.07260235      -3.81
0.0002


Dependent Variable: MOTIVATION
                                                              Standard
           Parameter                     Estimate                Error    t Value
Pr > |t|

           prog 1 vs. prog 2       -0.36032939      0.08022363      -4.49
<.0001
```

There is separate output for each of the outcome variables. Each of the tables in the output gives the estimate (in this case the difference between the coefficients), the standard error of this estimate, the t-value and associated p-value. The output indicates that the coefficient for **prog**=1 is significantly different from the coefficient for **prog**=2 for each of the outcomes.

The next example tests the null hypothesis that the coefficient for the variable **write** in the equation with **locus_of_control** as the outcome is equal to the coefficient for **write** in the equation with **self_concept** as the outcome. We request this test by adding a second **manova** statement, where **h** gives the predictor variable or variables to be tested (i.e., **h=write**) and **m** gives the combination of outcome variables to test (i.e., **m=locus_of_control - self_concept**).

```
proc glm data = data.mvreg;
  class prog  ;
  model locus_of_control self_concept motivation
    = read write science prog / solution ss3;
  manova h= _ALL_ ;
  manova h=write m=locus_of_control - self_concept;
run;
quit;
```

Again, we will only show the portion of the output associated with the new **manova** statement. The first table (shown below) gives the matrix for the outcome variables. In this case, we want to subtract the coefficients for **self_concept** (multiplied by -1) from the values of the coefficients for **locus_of_control** (multiplied by 1). Because motivation isn't involved in the test, it is multiplied by zero.

```
M Matrix Describing Transformed Variables

              LOCUS_OF_
                CONTROL      SELF_CONCEPT        MOTIVATION

MVAR1                 1                -1                 0


                 The GLM Procedure
         Multivariate Analysis of Variance

Characteristic Roots and Vectors of: E Inverse * H, where
            H = Type III SSCP Matrix for WRITE
                  E = Error SSCP Matrix

     Variables have been transformed by the M Matrix

         Characteristic              Characteristic Vector  V'EV=1
                   Root    Percent           MVAR1

             0.02001074     100.00      0.04807919


MANOVA Test Criteria and Exact F Statistics for the Hypothesis of No Overall
WRITE Effect
                 on the Variables Defined by the M Matrix Transformation
                           H = Type III SSCP Matrix for WRITE
                                  E = Error SSCP Matrix

                                 S=1    M=-0.5    N=296

Statistic                        Value    F Value    Num DF    Den DF    Pr >
F

Wilks' Lambda               0.98038183      11.89         1       594
0.0006
Pillai's Trace              0.01961817      11.89         1       594
0.0006
Hotelling-Lawley Trace      0.02001074      11.89         1       594
0.0006
Roy's Greatest Root         0.02001074      11.89         1       594
0.0006
```

The last table in the output shows that regardless of which multivariate statistic is used, the coefficient for **write** with **locus_of_control** as the outcome and the coefficient for **write** with **self_concept** as the outcome are significantly different.

For the final example, we test the null hypothesis that the coefficient for **science** in the equation for **locus_of_control** is equal to the coefficient for **science** in the equation for **self_concept**, and that the coefficient for the variable **write** in the equation for **locus_of_control** is equal to the coefficient for **write** in the equation for **self_concept**. To perform this test we need to use both the **contrast** statement and the **manova** statement. In the **contrast** statement, we specify the predictor variables we wish to test, in this case, we want to multiply the coefficients for **write** and **science** by 1. In the **manova** statement, we specify the portions of the test specific to the outcome variables, in this case, we want to compare the coefficients for **locus_of_control** and **self_concept**, by subtracting one set of coefficients from the other.

```
proc glm data = data.mvreg;
  class prog;
  model locus_of_control self_concept motivation
           = read write science prog / solution ss3;
  contrast 'write & science' write 1,
                             science 1 /e;
  manova m=locus_of_control - self_concept;
run;
quit;
```

As before, we will only show the portions of output associated with the test we are performing. Towards the beginning of the output (just after the class level information section) we see the table of contrasts for the coefficients. The matrix has two columns, one for each of the effects we wish to test.

```
Coefficients for Contrast write & science

                      Row 1           Row 2

Intercept                 0               0

READ                      0               0

WRITE                     1               0

SCIENCE                   0               1

PROG      1               0               0
PROG      2               0               0
PROG      3               0               0
```

The output shown below is generated by the **manova** statement, and as before it appears towards the end of the output.

```
M Matrix Describing Transformed Variables

              LOCUS_OF_
                CONTROL      SELF_CONCEPT        MOTIVATION

MVAR1                 1                -1                 0
Multivariate Analysis of Variance

Characteristic Roots and Vectors of: E Inverse * H, where
       H = Contrast SSCP Matrix for write & science
                  E = Error SSCP Matrix

Variables have been transformed by the M Matrix

Characteristic               Characteristic Vector  V'EV=1
          Root    Percent            MVAR1
```

```
    0.02150343     100.00      0.04807919


            MANOVA Test Criteria and Exact F Statistics for the
              Hypothesis of No Overall write & science Effect
          on the Variables Defined by the M Matrix Transformation
                 H = Contrast SSCP Matrix for write & science
                            E = Error SSCP Matrix

                             S=1    M=0    N=296

Statistic                        Value    F Value    Num DF    Den DF    Pr >
F

Wilks' Lambda               0.97894924       6.39         2       594
0.0018
Pillai's Trace              0.02105076       6.39         2       594
0.0018
Hotelling-Lawley Trace      0.02150343       6.39         2       594
0.0018
Roy's Greatest Root         0.02150343       6.39         2       594
0.0018
```

The last table in the above output shows that regardless of which multivariate statistic is used, taken together, the two sets of coefficients are significantly different.

Things to consider

- The residuals from multivariate regression models are assumed to be multivariate normal. This is analogous to the assumption of normally distributed errors in univariate linear regression (i.e., OLS regression).
- Multivariate regression analysis is not recommended for small samples.
- The outcome variables should be at least moderately correlated for the multivariate regression analysis to make sense.