

Using SAS Output Delivery System to Calculate the Probability Value for the F-test Used to Compare Variance Components

By Sandra E. Weinert, MS and Timothy L. Beck, MS, MT(ASCP)

Abstract

In the realm of medical diagnostics, there is often a desire to modify a device by offering new features. But it is important to show that the essential performance of the device has not been changed by the modifications. It is always necessary to show that the precision of the new device is not significantly worse than the precision of the old device. That is, test the hypothesis:

$$H_o: \sigma_{old}^2 = \sigma_{new}^2 \text{ versus } H_a: \sigma_{old}^2 \neq \sigma_{new}^2$$

There may also be interest in knowing if the precision has been significantly improved, thereby testing a different hypothesis. Before the release of SAS version 7, it was necessary to first calculate the variance component, and then manually enter the data to perform the F-test. With the launch of the Output Delivery System (ODS), this testing can be automated without manual entry, thereby simplifying the process and reducing the risk of errors. This paper demonstrates the previous manual method and automation of this method by using the Output Delivery System with version 8.2 of SAS.

Introduction

The Food and Drug Administration (FDA) regulates the medical device industry and all manufacturers must submit a 510(K) report to the FDA before they can legally market a new device in the United States. In a 510(K) report, a device is typically compared to an existing device, so a modified device is generally compared back to the original device. The goal is to show that the modified device is substantially equivalent to or better than the existing device.

Discussion

A typical diabetes-monitoring system consists of a test strip and a meter (device). During the life of the system, there may be opportunities to modify the device or the strip while keeping the other unchanged. If the device is modified, then both device types must work with the same strip and the results must be equivalent or improved. Studies are designed to collect data from both the modified and existing device and then the data are analyzed to determine if the two are equivalent.

For the simple case of comparing one glucose level (e.g. a fixed concentration solution) and one strip type between systems, the problem is solved simply by calculating the F-test. This can be accomplished using output from the VARCOMP procedure and manually constructing the PROBF function with the appropriate variance components.

However, equivalency testing is rarely this simple. After several modifications of the device and strip, the number of comparisons can become quite large. Suppose two different modified devices need to be compared to an existing device and that there are three different strip types that could be used. Another complicating factor is that these comparisons are typically done at three different glucose levels just to make sure that performance is the same at various points in the measurement range. In this set up, there are nine different experiments for each device type and 27 different comparisons possible across the three different device types. The way this analysis was performed in the previous versions of SAS was as follows:

Generate an ANOVA table to determine if the **device** factor is significant for each **level**, **strip type** and **device type**. The following is an example of that SAS code. Note, variable names are in bold text for readability.

```
proc anova;  
  by level strip_type device_type;  
  class device;  
  model glucose=device;  
  random device / test;  
run;  
quit;
```

In this case the **device** factor is random as the individual **devices** are randomly selected from a pool for each **device type**. Note that ANOVA procedure requires the explicit statement of the random variable.

Once it is determined that the **device** factor is significant, then variance component analysis is performed. This is done using PROC VARCOMP. The following is an example of the SAS code needed to perform this analysis.

```
proc varcomp method=type1;  
  by level strip_type device_type;  
  class device;  
  model glucose=device;  
run;  
quit;
```

Note that the VARCOMP procedure assumes the factor is random unless otherwise directed. The above code generates 27 different variance components. Constructing these comparisons manually requires each of these variance components to be entered twice along with the corresponding degrees of freedom. An F-test is used to compare the variances of each of the 3 **device_types** to each other for each **strip_type** and **level**. The F statistic is calculated as (Larger Variance Component/ Smaller Variance Component). The function that is used is ProbF to return the p-value associated with F-test. The following is the syntax used in SAS.

```
Pvalue=1-ProbF((Lg.Varcomp/Sm. Varcomp),lg varcomp df, sm. varcomp df);
```

The following is an example of a program that was written for this type of comparison (manual method).

```
/******  
data one;  
  pvalue_1=1-probf((2.96475/2.32301),9,9);  
  pvalue_2=1-probf((2.96475/1.16965),9,9);  
  pvalue_3=1-probf((2.32301/1.16965),9,9);  
  .  
  .  
  .  
  pvalue_25=1-probf((1.53829/0.37463),9,9);  
  pvalue_26=1-probf((2.37561/0.37463),9,9);  
  pvalue_27=1-probf((2.37561/1.53829),9,9);  
run;  
/******  
proc print data=one;  
run;
```

Obviously, one of the problems of re-entering data is the risk of keying in the wrong data. Using the method we discuss below, this manual re-entry can be prevented by using output directly from ODS as the basis for calculation.

The following sections are divided into 2 parts – the motivation and sample code.

Section 1

This section generates a data set to demonstrate this concept. In this case, 20 observations for each of 10 **devices** are generated for each of the 3 **strip_types** at each of 3 glucose **levels**. The 1st and 2nd **device_types** were generated together, with the 3rd **device_type** generated separately in order to change the variance for that group. The random normal function was used to keep the error structures the same and satisfy the necessary assumptions for the ANOVA. A seed of 1 was selected so that each time that this code was run, the results would be the same for demonstration purposes. (If a number of < 0 is used as the seed, SAS will use the system clock as the seed and therefore the results would change with each run of the program.) Also, for the sake of this demonstration, each **device** was varied in a step-wise way so as to force the **device** within **device_type** to be significant.

```

data simul8;
do strip_type= 1 to 3;
  do device_type= 1 to 2;
    do device = 1 to 10;
      do replic8=1 to 20;
        level=40;
        glucose= (level-3+device/2) + sqrt(20)*rannor(1);
        output;
        level=150;
        glucose= (level-5+device/2) + sqrt(20)*rannor(1);
        output;
        level=500;
        glucose= (level-10+device/2) + sqrt(20)*rannor(1);
        output;
      end;
    end;
  end;
do device_type= 3 to 3;
  do device = 1 to 10;
    do replic8=1 to 20;
      level=40;
      glucose= (level-6+device/2) + sqrt(30)*rannor(1);
      output;
      level=150;
      glucose= (level-10+device/2) + sqrt(30)*rannor(1);
      output;
      level=500;
      glucose= (level-20+device/2) + sqrt(30)*rannor(1);
      output;
    end;
  end;
end;
run;
quit;

proc sort data=simul8;
  by level strip_type device_type;
run;

```

Section 2.

An analysis of variance was performed, as in the earlier example, to determine if the factor **device** is significant in order to justify estimation of the variance components. Of all the cases generated with the simulated data set, only the glucose level of 500, **strip_type 1 device_type 2** and **strip_type 3 device_type 1** were not significant with p-values of 0.2320 and 0.2087 respectively. In these cases the variance component cannot be estimated, or are not significantly different than 0.

```
proc anova data=simul8;  
  by level strip_type device_type;  
  class device;  
  model glucose= device;  
run;  
quit;
```

Section 3.

This is the stage where ODS streamlines the analysis. Only 3 lines of SAS code are necessary to make the variance components and the degrees of freedom available in data sets. The statement “*ODS trace on;*” turns on the output delivery system option. The statement “*ODS output estimates=work.est anova=work.anova;*” puts the variance component estimates in to the SAS working data set of *est*, and the degrees of freedom in to the data set of *anova*. The variance components and the degrees of freedom are put into separate data sets by ODS and need to be merged into one data set in order to generate the p-values for the F-tests. (ODS writes the information to the Log window, which can provide information to the user as to the name of the dataset. The variable names can be found by looking at the dataset in the explorer tab within SAS.) The final statement “*ODS output close;*” stops output from ODS.

```
ODS trace on;  
  
ODS output estimates=work.est anova=work.anova;  
  
proc varcomp method=type1 data=simul8;  
  by level strip_type device_type;  
  class device;  
  model glucose=device;  
run;  
  
ODS output close;  
  
data newanova (drop=ss ms ems);  
  set anova;  
  if source in ("Error" "Corrected Total") then delete;  
run;
```

```

data newest;
  set est;
  if varcomp="Var(Error)" then delete;
run;

proc sort data=newanova;
  by level strip_type device_type;
run;

proc sort data=newest;
  by level strip_type device_type;
run;

data calc (drop=source);
  merge newest newanova;
  by level strip_type device_type;
run;

```

Section 4.

Once the *est* and *anova* data sets were merged together into a data set named *calc* the next step was to generate arrays for all the factors required for each of the 3 **device_types**. The required factors are **device_type**, the **variance component estimate** for each device type, and the **degrees of freedom** for each device type.

```

data getit (drop=device_type df varcomp);
  set calc;
  by level strip_type;

  retain mtyp1 est1 dfd1 mtyp2 est2 dfd2 mtyp3 est3 dfd3;
  array rest(*) est1-est3;
  array rdfd(*) dfd1-dfd3;
  array rmtyp(*) $ mtyp1-mtyp3;

  if first.strip_type then c=0;
  c+1;
  rest(c) = estimate;
  rdfd(c) = df;
  rmtyp(c) = device_type;
  if last.strip_type then output;
run;

```

Section 5.

The arrays generated in section 4 hold the data for comparison in this step. This code generates the F-test and returns the p-value for each **level** and **strip_type**. This data step determines which estimate is larger, putting the larger over the smaller and performs the three way comparisons between **device_types**. Finally, the returned p-values are printed. This section could be generalized with some more coding, and made to be more robust in the general data structure.

```
data end;
set getit;
by level strip_type;
  if est1 > est2 then pv_1vs2=1-probf((est1/est2),dfd1,dfd2);
  else pv_1vs2=1-probf((est2/est1),dfd2,dfd1);

  if est1 > est3 then pv_1vs3=1-probf((est1/est3),dfd1,dfd3);
  else pv_1vs3=1-probf((est3/est1),dfd3,dfd1);

  if est3 > est2 then pv_2vs3=1-probf((est3/est2),dfd3,dfd2);
  else pv_2vs3=1-probf((est2/est3),dfd2,dfd3);
run;

proc print data=end;
  var strip_type level pv_1vs2 pv_1vs3 pv_2vs3;
run;
```

When reviewing the p-values below, four of the comparisons in this example must be disregarded due to the insignificance of the **device** factors. Unfortunately, this decision of significance is still a manual operation. In this example, these were at glucose **level** 500, **strip_type** 1 for the comparisons of **device_types** of 1 vs 2 and 2 vs 3 and **strip_type** 3 for the comparisons of **device_types** of 1 vs 2 and 1 vs 3. At the glucose **level** of 500, **strip_type** 1 **device_type** 2 and **strip_type** 3 **device_type** 1 were not significant and therefore these variance components cannot be used, because they are not distinguishable from 0 or are inestimable. Using a proc print statement, the following table was generated. The disregarded p-values are shaded.

Strip Type	Level	pv_1vs2	pv_1vs3	pv_2vs3
1	40	0.36112	0.09105	0.16065
2	40	0.37123	0.14555	0.23140
3	40	0.23467	0.42884	0.18417
1	150	0.31665	0.47868	0.33581
2	150	0.04384	0.46763	0.05128
3	150	0.17482	0.27183	0.06438
1	500	0.01420	0.09298	0.00050
2	500	0.49197	0.39109	0.39882
3	500	0.02353	0.00558	0.26381

The results in the above table reveal 3 comparisons that were statistically significant, indicating that there were **device** to **device** differences within **strip_type** and **level**.

Conclusions

With the release of SAS version 7, the Output Delivery System became available, and has allowed automation of previously manual operations. Output from Proc Varcomp can be captured in a useful way by ODS into a data set, thereby removing a manual entry step. There is still manual work to be done, such as reviewing the ANOVA output to determine if the **device** factor is significant in order to interpret the results.

About the authors

Sandy Weinert, MS has worked for Roche Diagnostics for 14 years. She has a BS in Biomedical Science and a MS in Biostatistics from Western Michigan University. Sandy has over 14 years of experience using and programming in SAS. Her current position is in the Evaluations area of Research and Development, analyzing data for 510(K) submission to the FDA.

Tim Beck, MS, MT(ASCP) is a Director of R&D at Roche Diagnostics, heading the Applied Math & IT Group. Tim holds degrees in Biochemistry, Biology, Medical Technology, and Applied Statistics. Tim has been with Roche for 16 years, and uses SAS from time to time for data modeling, simulation, and production applications.

Contact Information

Sandy Weinert
Roche Diagnostics, Corp.
9115 Hague Rd.
Indianapolis, IN 46250
(317)-521-3429
sandy.weinert@roche.com

Tim Beck
Roche Diagnostics, Corp.
9115 Hague Rd.
Indianapolis, IN 46250
(317)-521-3947
tim.beck@roche.com