

Strategies for Developing and Delivering an Effective Online SAS® Programming Course

Justina M. Flavin, Independent Consultant, San Diego, CA

ABSTRACT

Teaching online courses is very different from teaching in the classroom setting. Developing and delivering an effective online class entails more than just transferring traditional course materials into written documents and posting them in a course shell. This paper will discuss the author's experience in converting a traditional hands-on introductory SAS® programming class into an online course and present some ideas to promote successful learning and knowledge transfer when teaching online.

INTRODUCTION

I have been an instructor for both traditional classroom and online courses at the University of California, San Diego, Extension Division since 2003. My teaching experience at UCSD Extension started with the development and teaching of a traditional hands-on introductory SAS programming class. This is a three unit course taught one night per week for three hours over nine weeks. Grades are based upon five short programming assignments, a comprehensive programming assignment due on the last day of the course, and class participation.

Several years later, I became involved in online learning when I assumed the teaching responsibilities of statistics courses that were developed by another instructor. While much effort had been expended on the development of these course materials, the content and structure was consistent with that of a traditional classroom course and in particular, did not utilize or take advantage of the technology inherent in a web based course. The course material consisted of textbook reading and weekly lectures in PowerPoint slide format without audio. However, it must be noted that this was the lecture format used at that time by UCSD Extension. There was also a weekly assignment of a single statistics problem; students were required to type up the solution and email it to the instructor. Students completed evaluation forms at the end of the course and comments regarding these courses were somewhat negative. Most students felt that these were essentially self-study courses and commented that there was little interaction with the instructor and no interaction with classmates. After doing some research on developing effective online courses, I began to make changes to these courses and now have completely redeveloped the course content. The weekly assignment has been replaced with a weekly multiple choice quiz which is graded immediately upon submission. To facilitate interaction between students, each student is assigned to a small group of 4-6 students and each group must analyze some data and then prepare a report on the analysis and findings. Evaluation comments from students are now approximately half positive and half negative, but the overall recommendation of the courses is positive.

For a number of years, administrators at UCSD Extension had been encouraging me to convert my SAS programming course into an online course. The need for an online course became apparent when the Director of Life Sciences, Hugo Villar, and I discussed the idea of developing a specialized certificate program in Biostatistics which would be offered completely online. The emphasis of the certificate would be the application of statistical techniques to the analysis of clinical data. Since SAS is the industry standard for the analysis of clinical data, my SAS programming course was to be one of the required courses for the certificate program.

COURSE DEVELOPMENT

OVERVIEW OF THE VIRTUAL LEARNING ENVIRONMENT

Online courses at UCSD Extension are taught through a web based system called Blackboard Academic Suite. Course material is organized using the various pre-defined pages available within the course shell. These pages are named Announcements, Introduction, Syllabus, Lessons, Discussion Board, Assignments, and Resources. There are a few required elements for some of the pages, but an instructor also has the flexibility to add additional items and materials to each page. Upon logging into the course, the user is taken to the Announcements page. The instructor usually provides some introductory course information on this page and updates it with course announcements and important information during the running of the course. Course lectures are found on the Lessons page.

The graphic below shows a sample page from a Blackboard course. This is the Lessons page for my programming course. The weekly lectures and other materials for that week are found within the folders. The toolbar on the left hand side is used to navigate to the other course pages.

The Discussion Board is an area where all course participants may post text for everyone to read, and it can be used to simulate a class discussion. I use the Resources page to provide links to SAS related websites of interest. From within the course shell, the instructor and students can send email which is delivered to external email accounts. Grades are recorded in an online grade book and students can view their own grades during the course. Other features available to instructors include the ability to see when students last logged in to the system and what parts of the course have been accessed by each student.

In designing my online course, I maintained the same syllabus, assignments and grading criteria as I had used in my traditional course. However, I replaced the class participation portion of the grade with Discussion Board participation. I also decided to make most of my course material available for the duration of the course to provide students with the opportunity to work ahead if they needed or wished to do so.

MAJOR CHALLENGES

While I had been interested in developing an online version of my SAS programming course for several years, I felt that there were two major difficulties.

The first difficulty entailed getting student access to SAS software at a reasonable cost. My traditional course was taught in a computer lab classroom and each computer had a copy of PC SAS loaded on it. For completing homework assignments, most students had access to SAS at their workplace. Students without workplace access to SAS either purchased SAS Learning Edition or came back to the computer lab classroom at other times to complete their assignments. For the online course, I considered the option of allowing students to use whatever version of SAS they had available at their workplace and requiring students without workplace access to purchase SAS Learning Edition. However, I decided that this option was probably not a good idea since I knew that I would have students running different versions of SAS on different platforms and that I might end up spending a good deal of time trying to diagnose and solve technical problems on installations about which I knew nothing. I decided that the only way I could offer an effective introductory course without having to play "remote IT help desk" was by having all students running the same version of SAS on the same platform. I considered requiring all students to purchase SAS Learning Edition, but I was concerned about the impact that requirement would have on my enrollment since the cost of the software was around \$200. In recent years, my course has attracted a large number of unemployed

professionals who pay the course fee out of pocket, so the additional software expense might make it impossible for them to take the course. For students who had access to SAS in the workplace, this requirement would also impose an unnecessary and sometimes non-reimbursable expense since some employers will reimburse tuition costs for professional education, but not for additional materials such as textbooks or software. In late 2008, I became aware of a web based educational version of SAS called SAS® OnDemand for Academics which, at \$60 per student, appeared to be a cost effective solution that would enable me to have all students using the same version of SAS.

The second difficulty in creating an online version of my traditional course was that I had never developed any formal PowerPoint presentations for my traditional course. Since it was presented as a hands-on course in a computer lab, it was a highly interactive class. My weekly lectures mostly consisted of a brief discussion of topics and concepts, running SAS code which illustrated the concepts, and then discussing the code and results. I was uncertain as to how I could convert code examples and the subsequent discussion into a meaningful presentation for an online course. I also wished to capture some of spontaneity of my traditional course. Eventually, I resolved this problem by teaching my own class to myself (and my parrots) over a period of a few weeks. For each lecture, I talked through each concept and code example and then typed up my discussion as text around the code. I also included many screen shots of the various windows in SAS as I discussed them to ensure that students could follow along easily if they were also running the code in their own SAS session.

The graphic below shows a slide from one of my lectures. The audio portion of the slide consists of me reading and talking through the text. Students have the ability to view and review the slides as often as they wish. Copies of the slides are also available in .pdf format for students to download.

Performing Conditional Processing

Justina Flavin

One way to fix this problem is to rewrite the first if statement as:

```
if . < Age < 20 then AgeGroup = 1;
```

"." is the notation used to denote a missing value. Since a missing value is smaller than the smallest value in your data, this will exclude all missing values.

Since we would not anticipate negative values for Age, we could also rewrite the statement as:

```
if 0 <= Age < 20 then AgeGroup = 1;
```

Program 7-2 provides another option, illustrating the use of the missing function:

```
if Age lt 20 and not missing(age) then AgeGroup = 1;
```

The missing function checks if a variable value is missing (or not). SAS processes this statement by evaluating each value of age. If Age is less than 20 and age is not missing, then AgeGroup=1

Yet another equivalent way to write the line is:

```
if Age lt 20 and age ne . then AgeGroup = 1;
```

Copyright 2008, Justina Flavin. All rights reserved. No part of this material may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of the copyright holder. 5

Outline	Thumb	Notes	Search
Slide Title			Duration
Slide 2			01:09
Slide 3			00:42
Slide 4			00:38
Slide 5			00:47
Slide 6			00:08
Slide 7			00:41
Slide 8			00:37
Slide 9			00:27
Slide 10			01:13
Slide 11			00:28
Slide 12			00:13
Slide 13			00:27

6 Minutes 34 Seconds Remaining

Slide 5 / 21 | Playing 00:07 / 00:47

COMPARISON OF TRADITIONAL AND ONLINE COURSES

COMMUNICATION

Verbal communication is an essential element of traditional courses and this is probably the most distinct difference between traditional and online courses. When teaching in the classroom setting, if I observe a sea of bewildered faces, I can react immediately to the situation by providing a different explanation for a difficult concept, modifying code to create a new example, or reviewing and clarifying material previously covered. Written material is supplemented by verbal conversation and discussion, so students have the opportunity to write down notes to clarify anything found to be vague. Students will ask questions if they need clarification and once a question is asked, all students can hear the response, so the question does not need to be asked again.

In an online course, students rely solely on written material so instructions and explanations must be much more specific and detail oriented. The course material is static and this has its advantages and disadvantages. The lectures can be reviewed by replaying them as many times as a student wishes. This is, of course, not possible with a live lecture. But precisely because the material is static, online courses can also be more frustrating to some students when they encounter problems or concepts that they have difficulty understanding. From my traditional course, I know that providing a different explanation sometimes clarifies the issue, so in my online course, I try to provide students with alternative online sources (such as the SAS Online Documentation) where they might look to find the same information presented in a slightly different way.

Students in online courses also lose the ability to ask questions spontaneously and receive an answer immediately. Students may contact me with questions via email and then wait on my response. When I respond, only the student who sent the message sees the response, so sometimes I end up answering the same question multiple times. Receiving the same questions repeatedly is sometimes indicative of a section of poorly written course material and rewriting the section for future classes sometimes reduces the number of questions. In some courses, I have used the Discussion Board to post the questions sent to me and my answers, so that all students may see these.

INTERACTION WITH CLASSMATES

Students usually feel much more isolated and disconnected from their classmates in online courses. To address this issue, I incorporate the use of the Discussion Board in my courses. The Discussion Board is an area where all course participants may post text for everyone to read, thus simulating a class discussion to some extent. To familiarize students with the Discussion Board area, I have students in all of my courses post an introduction about themselves during the first week of the class. They are asked to comment on their knowledge of any programming languages, reasons for taking the course, and geographic location. In my traditional course, I go through the same exercise at the first class meeting, omitting, of course, the question about geographic location. In subsequent weeks, I post "thought questions" on the Discussion Board. These are questions which do not have right or wrong answers, rather, they are meant to allow students to think about some of the concepts and various programming issues they may encounter while working as a SAS programmer. I developed these topics from discussions that have taken place in my traditional class. Some examples of my Discussion Board topics are as follows:

Program Errors: What things can you do to diagnose errors in your programs? Do errors always generate ERROR lines in the log? If not, what kind of errors do not generate error messages? What other messages in the log should be reviewed? If your code appears to be correct, where else might you look?

Missing Values: If you are writing a program to read in data from a file, what statements or options might you need to use if there are missing values in the data? How might you determine if there are missing values in a data set? Is missing data always a problem that should be investigated? Why or why not? When manipulating data that has missing values, is it better to use an assignment statement or a function? Why?

Once I post these topics, students write up and post responses both to my initial posting and comments left by other students. In my experience, most students write up a response and simply post it to complete the assignment. Only a few take the time to comment upon others' responses, so it is not quite the same as a classroom discussion. However, because students have time to read and reflect upon the questions, responses tend to be more thoughtful than those received in a spur of the moment in-class discussion. In course evaluation comments, some students have noted that they felt more engaged than an in-class discussion simply because they are not quick thinkers and preferred having more time to develop a response. After the due date for each topic, I post my own thoughts about the topics and point out any particularly good student responses.

The graphic below displays the Discussion Board area for my Missing Values topic. My name appears on the top thread since I started the discussion. This is followed by the name of each student who responded and the date of the response. The postings can then be read by expanding the blue topic headers.

The screenshot shows the UC San Diego Extension website interface. The top navigation bar includes links for Home, Help, Logout, Inicio, Ayuda, and Cerrar sesión. Below this is a secondary navigation bar with links for My Courses / Mis cursos, FAQ, Contact, Student, and Instructor. The left sidebar contains a list of course-related links: Announcements, Introduction, Syllabus, Lessons, Discussion Board, Assignments, Resources, and Contacts. The main content area displays a thread titled 'Missing Values' under the path 'SAS PROGRAMMING (072555 FA09) > DISCUSSION BOARD > DISCUSSION TOPIC 3 > THREAD DETAIL'. The thread details include a 'Thread: Missing Values' header, a 'Total posts: 8' and 'Unread posts: 0' status, and a list of posts. The first post is by JUSTINA MARY FLAVIN on 10/4/09 at 3:58 PM. Subsequent posts are replies by WILLIAM GARNER, MANINDER PAL KAUR, YUAN YUAN, TERRY BRABHAM, STEVEN ANUSKIEWICZ, JEFF WANG, and JUSTINA MARY FLAVIN, all dated 10/26/09 or 10/27/09. The interface also includes a search bar, a 'Refresh' button, and a 'Select All' checkbox.

Another way in which I have attempted to encourage student interaction is by allowing group submission of assignments. Students who wish to work together on assignments then submit a single assignment and all group members receive the same grade. Learning is facilitated when students work together because they can help each other understand the material. From a grading standpoint, the collaborative efforts usually result in a better quality assignment submission and reduce the number of assignments that I must grade. In my online courses, I have noticed that students will sometimes choose this option if they already know other students in the course (such as co-workers from the same company or organization). However, I have yet to see any students who did not previously know each other prior to course enrollment choosing to work together. In contrast, in my traditional courses, I have had students who only met each other in the course choosing to work together on assignments. This behavior seems to support the idea that student interaction is more inhibited in online courses than in traditional courses.

INSTRUCTOR TIME SPENT ON COURSE

In an online course, an instructor spends more time developing the course material than in a traditional course. However, once the lectures are developed and recorded for an online course, there is no further time spent on teaching weekly lectures.

I have found that I spend more time grading assignments in my online courses. In my traditional course, students print out and hand in their program code and log each week. The following week, they receive their graded assignment back with my handwritten comments. Additionally, I spent a bit of time in class reviewing and discussing the assignment solutions and answering questions.

In the online course, students upload their code file and log to Blackboard and then I must download all of these files. In a class of 20-30 students, this means I download 40-60 files. The Blackboard system does not have the capability to create zip files and each file must be downloaded individually, so just downloading the files each week can take some time. Additionally, I always have a few students who fail to follow my instructions on file types and file naming conventions, so I have to spend additional time to rename or convert files.

As I review each student's assignment, I use a spreadsheet to record any deductions and comments for each student. Once the assignments are graded, I create a document which has the assignment solution, my general

comments, and a list of the errors and corresponding deductions for each. I send this document out to all students so they can see the grading schema and compare their solution to my solution to understand their grade. Students can then email me if they still have questions about the solution or their grade. Most students seem satisfied with this method of feedback, but there are always some who contact me after each assignment for further feedback on their grades. Using my spreadsheet, I can easily review the grading for those students. For the final comprehensive assignment, I do provide each student with personal feedback because it is a longer assignment and accounts for half of the course grade.

The text box below shows an example of my assignment solutions and comments document.

18-8 (6 points)

2 pts: PROC SORT correct

4 pts: 2 pts per correct statement - var, sum or sumby statements

-1 pt: adding a where clause and not including all animals in output

-1 pt: not using the permanent SAS data set as input to the data step

-0.5 pt: Adding region on VAR statement

-1 pt: not using PROC PRINT

-0.5 pt: addition of extra unnecessary functions

I chose this problem because it simulates a fairly common real life scenario of being given some (partial) output and then being asked to write the code to reproduce it.

The output shown in the book was only a part of the total output and all values of country should have been included in your output (i.e. no where clause should have been included). This could have been verified by comparing your overall total line numbers to those listed in the problem.

There were two ways of generating the totals and subtotals using either a sum or sumby statement.

```
sum count cars;
or
sumby region;
```

The same result could also be obtained using BOTH statements, but that is redundant.

When using both an ID statement and a BY statement having identical variables on the two statements, it is not necessary to include the NOOBS option on the PROC PRINT statement.

It was not necessary to create a temporary data set DOMESTIC before sorting the permanent data set DOMESTIC, rather a temporary, sorted data set can be created by using the OUT= option on PROC SORT.

```
data domestic;
  set learn.domestic;
run;

proc sort data=domestic;
  by Region;
run;
```

can be simplified as:

```
proc sort data=learn.domestic out=domestic;
  by Region;
run;
```

ENROLLMENT

Student enrollment in the online course has been much higher than the traditional course. To date, I have taught the traditional course 7 times with a total enrollment of 62 students, or an average of 9 students per class. I have taught the online version twice, with a total enrollment of 37 students, or an average of 18.5 students per class. I believe that the increased enrollment is due to two reasons. The first and most obvious reason is the increased accessibility

and flexibility of online courses. My enrollment is no longer limited to students who live in the San Diego area and can attend a three hour course one evening per week for nine weeks. The second reason is that the course is no longer a standalone course. It is now a required course for the specialized certificate in Biostatistics as well as a course of interest for the specialized certificate in Data Mining.

STUDENT ACHIEVEMENT

To compare student learning in the classroom setting to the online setting, I used PROC FREQ to perform two different grade comparisons for all courses I have taught.

The output below shows the results for students who took the course for a letter grade:

Frequency Percent Row Pct Col Pct	A	B	C	D	F	Total
Classroom	42 50.60 82.35 60.87	4 4.82 7.84 66.67	0 0.00 0.00 0.00	2 2.41 3.92 50.00	3 3.61 5.88 100.00	51 61.45
Online	27 32.53 84.38 39.13	2 2.41 6.25 33.33	1 1.20 3.13 100.00	2 2.41 6.25 50.00	0 0.00 0.00 0.00	32 38.55
Total	69 83.13	6 7.23	1 1.20	4 4.82	3 3.61	83 100.00

Statistic	DF	Value	Prob
Chi-Square	4	3.7760	0.4372
Likelihood Ratio Chi-Square	4	5.1236	0.2748
Mantel-Haenszel Chi-Square	1	0.2790	0.5974
Phi Coefficient		0.2133	
Contingency Coefficient		0.2086	
Cramer's V		0.2133	

WARNING: 80% of the cells have expected counts less than 5. Chi-Square may not be a valid test.

Fisher's Exact Test

Table Probability (P)	0.0104
Pr <= P	0.5190

Sample Size = 83

For the second comparison, I included both letter grades and Pass/No Pass grades. Letter grades of a "C" or higher were converted to "Pass" and grades of "D" or "F" were converted to "No Pass". The results are shown below:

Frequency Percent Row Pct Col Pct	NP	P	Total
Classroom	9	53	62
	9.09	53.54	62.63
	14.52	85.48	
	75.00	60.92	
Online	3	34	37
	3.03	34.34	37.37
	8.11	91.89	
	25.00	39.08	
Total	12	87	99
	12.12	87.88	100.00

Statistic	DF	Value	Prob
Chi-Square	1	0.8933	0.3446
Likelihood Ratio Chi-Square	1	0.9405	0.3321
Continuity Adj. Chi-Square	1	0.3930	0.5307
Mantel-Haenszel Chi-Square	1	0.8842	0.3470
Phi Coefficient		0.0950	
Contingency Coefficient		0.0946	
Cramer's V		0.0950	

WARNING: 25% of the cells have expected counts less than 5. Chi-Square may not be a valid test.

Fisher's Exact Test

Cell (1,1) Frequency (F)	9
Left-sided Pr <= F	0.8999
Right-sided Pr >= F	0.2707
Table Probability (P)	0.1705
Two-sided Pr <= P	0.5265

Sample Size = 99

For either comparison, the p-values are not significant for either the Chi-Square or Fisher's Exact Test, so it appears that students do equally well in either course setting.

EVALUATION OF SAS® ONDEMAND FOR ACADEMICS

I have used SAS OnDemand for Academics for two online courses and it has worked well. SAS OnDemand for Academics provides an online delivery model by connecting to a SAS-hosted server over the web.

As the instructor, I register the class through an application on the SAS web site. Along with administrative information such as course title and section number, I also provide course start and end dates. This information is presumably used to control the length of time the students have access to the software. Once the course is registered, a directory for the course is created on the SAS server. Course data files can be uploaded to the server using FTP. Students register for the course license on the SAS website. Once they have registered, they receive an email with instructions for downloading and installing the software. When the application is started, a username and password is required to connect to the remote SAS server.

INSTALLATION AND TECHNICAL SUPPORT

As mentioned previously, having all students using the same version of SAS on the same platform is a clear advantage to the instructor because it eliminates issues related to different installations of SAS running on different platforms. The vast majority of students have had no problems downloading and installing the SAS OnDemand software. A few students did encounter some problems either with the installation or their account setup, but in all cases, the students were able to resolve their problems by submitting a Tech Support request. SAS OnDemand for Academics does not run on Macintosh systems and this has inconvenienced a few students.

ENTERPRISE GUIDE VS. DISPLAY MANAGER

SAS OnDemand for Academics uses the Enterprise Guide interface rather than Display Manager. While Enterprise Guide offers the same functionality as Display Manager for writing and running code, if given a choice, I would use Display Manager for two reasons. First, in my own experience, Display Manager is much more widely used in companies and organizations that are performing clinical data analysis. Since a large number of my students are taking the course because they are or wish to become employed as clinical data analysts, teaching the course using Display Manager would be much more beneficial to them. Secondly, most introductory SAS programming books, including the textbook used in my course, *Learning SAS by Example: A Programmer's Guide*, provide information and graphics on Display Manager and do not address the programming environment in Enterprise Guide. And most books on Enterprise Guide focus on the point-and-click environment rather than the programming environment.

SERVER AND OPERATING SYSTEM

The server used for SAS OnDemand for Academics uses the UNIX operating system. For students unfamiliar with UNIX, this has caused some confusion when writing libname or filename statements. Common problems include the use of the forward slash '/' rather than the backward slash '\' and the desire to add a drive letter at the beginning of the pathname.

```
So libname learn "C:\books\learning";
```

needs to become

```
libname learn "/courses/u_ucsd.edu1/i_536036/c_629/saslib";
```

But for some students, it becomes

```
"C:\courses\u_ucsd.edu1\i_536036\c_629\saslib";
```

FILE LOCATION

Once a course is registered or defined in the SAS OnDemand for Academics Control Center by the instructor, a directory for that course is created on the server and data files and data sets can be uploaded using FTP. Only the instructor may upload files to the server. There are limitations on where data files may be stored on the server. SAS data sets must be stored in the /saslib directory. Data in flat files which can be read using an infile statement must be stored in the directory one level above the /saslib directory.

I do not like the limitations of this structure because it leaves many beginning programming students with the false impression that data sets and data files must be stored in different directories. However, having all students accessing the data in exactly the same place does provide one advantage to the instructor because if I wish to run code submitted by students on homework assignments or code sent to me by email, I do not need to change libref or infile statements. In my traditional course, since students use different installations of SAS, every student has a different directory structure, so libref and filename statements are unique to that structure.

FILE TYPES

SAS data sets created using PC SAS can be uploaded to the server, but if these are not regenerated under UNIX, an (annoying) conversion NOTE appears in the log each time each time the data set is accessed in the session.

NOTE: Data file LIBNAME.DATASET is in a format that is native to another host, or the file encoding does not match the session encoding. Cross Environment Data Access will be used, which might require additional CPU resources and might reduce performance.

Flat files may seemingly have any file extension name (or none). However, these types of files, if created on a PC or other non-UNIX operating system, must be converted to a UNIX file format before uploading to the server. Otherwise, these are unreadable and cause errors to be generated in the log.

SAS data sets and flat files are the only type of files which may be stored on the server. Excel spreadsheets (.xls file types) and non-data type files such as SAS program code or Word documents cannot be stored there.

FILE ACCESS IN THE SAS SESSION

In the SAS session, data on local drives cannot be accessed. Only data sets and data files that are stored on the server can be accessed in filename and libref statements. But SAS code in a file must be included from and saved to a local drive.

This causes confusion amongst some students. Many assume that since they must access the local drive for their code files, that SAS OnDemand can also access data files and data sets stored in these local directories.

For assignments requiring students to read a flat file using infile and input statements, if descriptive information about the file structure such as column widths and variable types are not provided, then students must be able to view the file to write the appropriate input statements. While these files reside on the SAS server, and can be read into a SAS program, they cannot be opened or viewed within the SAS OnDemand application. There also is no functionality to view the file on the server using a simple editor such as Notepad since there is no option available to view contents of the course directories via a Windows Explorer type window. Thus I must provide students with copies of these files through Blackboard. Again, this creates confusion for some students because they must view the file locally, but in their program, they must access the version of the file located on the server.

DIRECTORY PERMISSIONS

Only the instructor has write permission to the server directories. This is advantageous because it ensures that students cannot accidentally overwrite permanent data sets that are used by all students. However, a disadvantage is that students cannot create their own permanent data sets because they can neither write to the server nor to a local drive. So in my assignments, I am limited to having students create and modify only work data sets. This causes some students to incorrectly assume that permanent data sets are indeed permanent and cannot ever be changed.

In my traditional course, my course data files are stored on a server directory that is accessible to all students. I have the ability to create a subdirectory for each student. This provides an area where each student can create permanent data sets without cluttering up the course data directory or overwriting permanent course data sets. While I could create subdirectories for each student on the SAS OnDemand server, this would be a pointless exercise, since the students do not have write access to the server.

OVERALL ASSESSMENT

Because there are some fundamental differences in the programming environment in SAS OnDemand for Academics and a regular SAS installation, my assessment is that SAS OnDemand for Academics provides an adequate but less than ideal environment for teaching an introductory SAS programming class. However, I will continue to use it in my courses. Students will also appreciate that as of August 2010, SAS is no longer charging a fee for student accounts and I expect that this will positively impact enrollment in my courses in the future.

Additionally, the SAS employees whom I have contacted about issues or problems with the product have been extremely helpful. In particular, Mike Janes has been instrumental in resolving issues with student accounts. He also has helped me to “recycle” and reuse my course setup each time my course has been offered. By reusing the same physical directory, I do not have to upload my course data to a new directory each time the course is offered, and I do not have to update any of my course materials such as the libref and filename statements in the course code which I provide to the students.

CONCLUSION

Creating and teaching my introductory SAS programming course as an online course has been a positive experience. Having taught the course for a number of years as a traditional course was very helpful in developing the material for the online version of the course and has also served to highlight the differences in teaching both types of courses. Traditional courses provide an advantage in promoting personal interaction that cannot be duplicated in online courses. But online courses provide more freedom and flexibility to both the student and the instructor. With the online version of my course, I do not have to commit to teaching one night per week for nine consecutive weeks which has enabled me to teach my online course more frequently. Online courses have been shown to be as effective as or more effective for learning than traditional courses and the analysis of the grades in my courses seem to support this. Online learning is not for everyone, but for those who are motivated self-reliant problem solvers with good time management skills, online classes can provide a great environment for learning.

REFERENCES

Cody, Ron. 2007. *Learning SAS® by Example: A Programmer's Guide*. Cary, NC: SAS Institute Inc.

ACKNOWLEDGMENTS

Thanks to the staff in Instructor Services at UCSD Extension for the ongoing administrative support. In particular, I would like to recognize Hugo Villar, Director of Life Sciences, who provided guidance for developing the specialized certificate in Biostatistics and Felicia Murray, Program Representative, who provided the grade and enrollment information used in this paper.

Thanks to Mike Janes at SAS Institute for the timely responses to my questions and issues with SAS OnDemand for Academics.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Justina M. Flavin
E-mail: justina.flavin@gmail.com

Justina Flavin is a SAS® programmer working as an independent consultant in San Diego. She is an instructor for courses in SAS programming and statistics at the University of California, San Diego-Extension Division. At Extension, she is also the curriculum developer and advisor for the Biostatistics certificate and advisor for the Data Mining certificate. Justina served as Conference Chair of PharmaSUG '99 and has taught seminars at PharmaSUG and WUSS.

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 are trademarks of their respective companies.