

Office Pooling with SAS® Software

Wayne Woo, Berlex Laboratories, Richmond, CA

Abstract

In April 2000, Indianapolis has the honor of hosting two big events, SUGI and the NCAA Basketball Final Four. During this time of year, countless amounts of office pools crop up to host friendly wagers among co-workers over who will win the tournament. Each pool will have an organizer who manages the pool. Many organizers will use manual and paper-based methods to run their pool. While learning SAS/AF® hoping to pick up additional programming skills, I built an application that enables NCAA pool organizers to more easily manage their office pools. This paper describes the resulting product, its features, and how they are implemented in SAS.

Introduction

For some time, I had wanted to learn SAS/AF software, but lacked an application. Then during the 1999 NCAA Tournament when I was participating in an office pool, I thought about running my own pool. I fancied that there would be a computerized system where pool participants would submit their picks, be able to view standings at any time during the tournament and be able to see other people's picks.

I soon found that this type of application was not a novel idea. I learned that CBS Sportsline.com™ had a free product called Tournament Tracker. Furthermore, while researching for this paper, a Yahoo! internet search using the keywords "office pool software" yielded five commercially available products. In the end, my goal was not to produce a novel application, but to implement some of the best and essential features from the other products in SAS and learn SAS/AF in the process. This paper describes the resulting product, its features, and how they are implemented in SAS.

NCAA Basketball Tournament Primer

Sixty-four (64) of the best collegiate teams are invited to play for a championship at the end of the college basketball season. Teams are assigned into four (4) regions (East, South, Midwest, West) and placed into a single-elimination bracket. Winners of each game advance along the bracket until the final two remaining teams play for the national championship. The Final Four consists of the winners of each region. There is great pressure in each game due to the single-elimination factor. Hence the label "March Madness" is used to alias this tournament.

NCAA Basketball Office Pool Primer

While there are many types of office pools, they invariably involve friendly wagers among co-workers, friends and

family. Underlying the pool is usually a major sporting event. Here we focus on pools run during the NCAA Basketball tournament.

Generally, there is a person ("Organizer") who runs the pool. Participants ("Players") will contribute an entry fee (perhaps \$5) into the pool. Players fill in a bracket with their picks of teams they believe will advance through the tournament. In case the pool ends with a tie, Players usually are asked for the combined number of points scored by both teams in the final game. This is one of many possible tie-breaker scenarios.

Within the pool, point values are assigned to each round of the tournament with earlier rounds receiving less weighting. Correctly picking the winning teams in a particular round adds the corresponding point values to a Player's total. There are higher stakes in correctly predicting winning teams in later rounds.

At the end of the tournament, the Player with the most points wins 1st prize. Depending on the size of the pool, there may be other prize levels. The tie-breaker is used whenever two Players end up with the same score.

Product, Features and Implementation

Now I will show you what I built. You will learn what I considered to be important features to include and you will see their implementation in SAS. You will see how this application can be used to manage an office pool.

Data set Structure

First, I needed to create a data set structure that would store each Player's picks. The design of the data set had performance and convenience implications. I settled on a wide file approach that would contain one observation per Player. Each Player's observation contained all her picks as well as a complete copy of the tournament field. Additionally, the data set contained the point values allocated to each round and the Player's tie-breaker entry. This scheme led to some bulky code and used more storage but meant easier data processing. I was also able to work with a single record most of the time. The data set contains 135 variables.

A special Player called "TOURNAMENT" is the first observation added when the data set is created. The Organizer uses this Player to store the actual results from the tournament. This observation serves as a key for scoring how well the pool participants did.

One essential feature is the interchangeability of pool data sets. This means one can run multiple pools using the application. The opening frame entry (MAIN) allows the Organizer to select a "Pool Data set" to use.

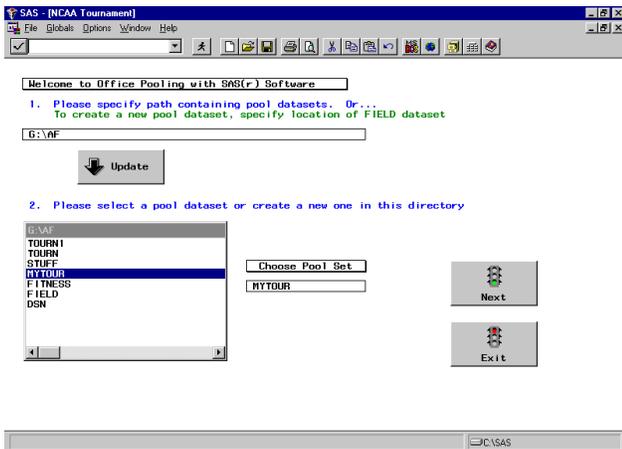


Figure 1. Screen shot: Pool Data set selection

A text entry object allows the Organizer to type a directory location. Pressing the UPDATE button lists all the SAS data sets in the directory. To choose a data set you either select one from the list box or use the text entry box under "Choose Pool Set" to type a valid name for a new data set. Then press NEXT. A check is made to determine whether the data set conforms to a Pool Data set structure. If a new data set is specified, it is created in the directory chosen and will contain an observation for the TOURNAMENT Player.

The FIELD data set (see below) serves as a template for a new Pool Data set. When creating a new Pool Data set, you specify the location of the FIELD data set to enable the application to find this template. Once created, the Pool Data set can be moved to another directory.

Application Capabilities

Once a Pool Data set has been chosen, the frame entry in Figure 2 appears.

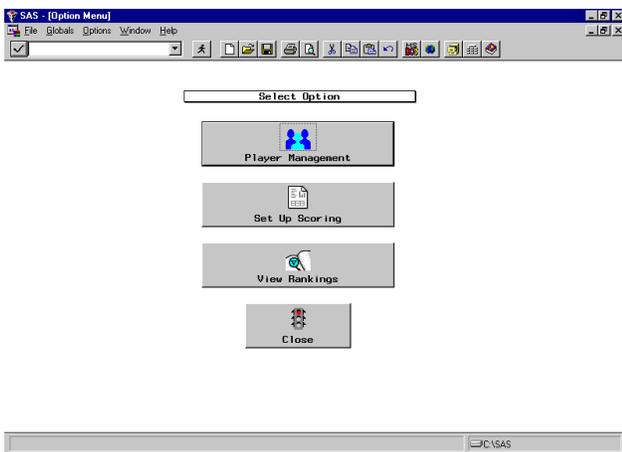


Figure 2. Screen shot: Options Menu

There are basically 3 capabilities in this pool management program. These features are almost universally found in commercially available products and represent the basic functionality needed to manage a basketball pool. Here is a description of where each button takes you.

Player Management is where the Organizer can enter Players' bracket picks. Actual results from the tournament can be tracked via the TOURNAMENT Player.

Set Up Scoring is where points are assigned to each round's games. For example, if one (1) point is assigned to Round 1 games, whenever a Player correctly picks the winner of a Round 1 game, one point is added to the Player's score.

View Rankings is where number crunching gets done. The output shows each Player's score and where he/she ranks relative to other Players.

Below are more details of what you will find upon choosing each of the options.

Player Management

The heart of an application like this lies in the screens used to manage Players and enter their picks.

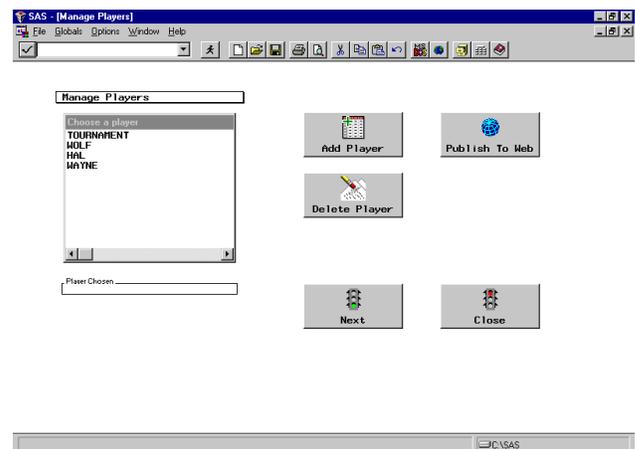


Figure 3. Screen shot: Player Management

The Player Management frame entry gives you the ability to add and delete Players. Adding a Player submits SAS code that copies the structure of an existing observation and creates a new record. Deleting a Player merely issues the SCL DELOBs function on the appropriate observation.

With office intranets widely available these days, there is a feature to write a Player's bracket picks into an HTML document (using SCL functions FPUT and FWRITE). Thus the Organizer can put together a quick web site to allow Players to view each others' pick sheets. The ability to see what others have picked increases the integrity of the pool.

Unlike similar products, I wanted to create bracket screens that were understandable and not cluttered. Instead of one big bracket, there are five screens corresponding to the four regions and the Final Four. Figure 4 depicts the South regional bracket for a fictional Player named "WOLF".

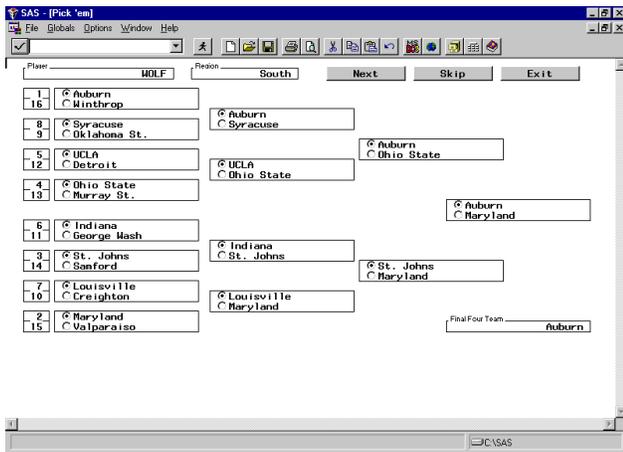


Figure 4. Screen shot: Brackets

Each game is represented by a radio box. This enables selection via a one-click operation. The chosen team is automatically registered in the next round through program statements associated with each radio box. On the Final Four frame, the Player's tie-breaker (combined points scored by both teams in the championship game) can be entered into the text entry box.

At the top of each screen are the name of the Player and the region. Pressing NEXT saves the data to the Pool Data set. SKIPPING or exiting the screen does not save the current screen's data.

Set Up Scoring

On this screen point values are assigned to each round of the tournament. Generally, not all rounds are scored evenly because it is more difficult to predict a winner in a later round.

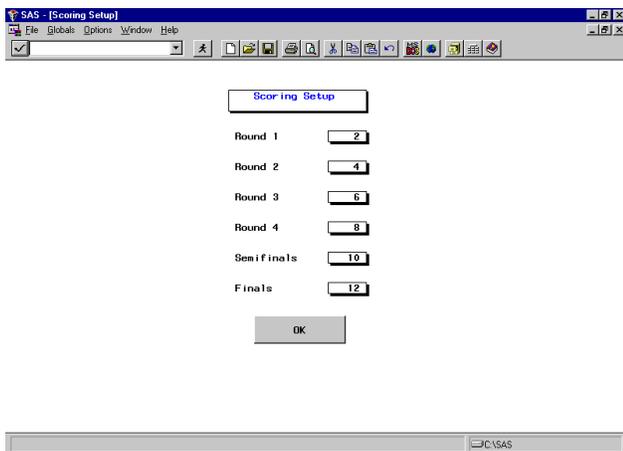


Figure 5. Screen shot: Set Up Scoring

Here the Organizer can use the text entry boxes to enter appropriate point levels. Clicking OK saves the data to each observation in the Pool Data set.

View Rankings

Under this option you will find a scoring mechanism that computes how many points each Player has earned at

any moment during the tournament. SAS statements are submitted (via the SCL SUBMIT) to compare each Player's picks with the TOURNAMENT results to compute a score for the Player. Players are ranked from highest point total to lowest. The viewer shows the output of a PROC PRINT (stored in a temporary catalog) listing the Player's name, score, and tie-breaker. The Organizer determines how the tie-breaker is used if there is a tie among Players at the end of the tournament. For example, one possible tie-breaker scenario would select the Player closest to the total combined points scored in the championship game without going over.

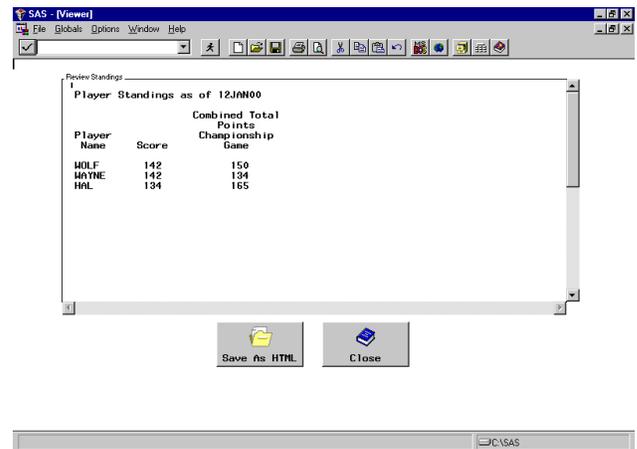


Figure 6. Screen shot: View Rankings

The "SAVE AS HTML" button saves the standings into an HTML file in the directory of your choice.

How To Get Started

An important consideration is year-after-year usage. In order to achieve this, there must be a way to enter each year's tournament teams into the application and make other minor adjustments. In this application, the initial 64 team field is entered into a comma-delimited text file which is then processed by a SAS program into a data set called FIELD. FIELD serves as a subset template for each Pool Data set.

The text file has the following structure: seed, team, region. Figure 7 shows a sample of this file.

The Organizer chooses a directory to store the FIELD data set. When creating a new Pool Data set, it will reside in same directory as FIELD. You are free to move the Pool Data set after its creation.

Since the Final Four regional matchups go through a rotation each year, a facility must be provided to capture this year-to-year nuance. FINALORD.LIST is a catalog entry where this is done. The values EAST, WEST, SOUTH, and MIDWEST are entered into the list with the regions playing each other listed side by side. For example, in the year 2000, East plays South and West plays Midwest. The entries into FINALORD.LIST would be EAST - SOUTH - WEST - MIDWEST. Ordering is important because the Final Four bracket screen relies on this data.

```

1,Auburn,South
2,Maryland,South
3,St. Johns,South
4,Ohio State,South
5,UCLA,South
6,Indiana,South
7,Louisville,South
8,Syracuse,South
9,Oklahoma St.,South
10,Creighton,South
11,George Wash,South
12,Detroit,South

```

Figure 7. Portion of tournament field text file

When all preparations are complete, you invoke the application by issuing an AF command to start the MAIN.FRAME entry.

SAS Implementation Versus Other Products

What makes this SAS implementation more flexible than other products?

- (1) it can be used for any type of 64 team event where an office pool is run
- (2) it can be run on multiple platforms with little modifications due to SAS multiplatform architecture
- (3) it is open source (important buzzword today!)
- (4) it is programmed in SAS...shouldn't that count for anything?

Summary

While it is doubtful that someone would buy SAS to run this application for managing a pool, this was a useful exercise to learn about SAS/AF software. It is often difficult to learn new skills and software without something concrete to apply those lessons toward. It can be worthwhile look for mundane little things in life that can be done in a different way.

References

SAS Institute, Inc., *SAS® Screen Control Language: Reference, Version 6, Second Edition*, Cary, NC: SAS Institute Inc., 1994. 648pp.

SAS Institute, Inc., *Getting Started with the FRAME Entry: Developing Object-Oriented Applications, Version 6, First Edition*, Cary, NC: SAS Institute Inc., 1991. 81 pp.

Notes

All views and opinions expressed in this paper are solely those of the author.

Biography and Contact

Wayne Woo works as a programmer for a pharmaceutical company. He is a SAS Certified Professional and has presented at WUSS. In his spare time, he experiments with lots of new and old technologies to find better ways to accomplish a task.

Project files and source code are available from the author.

Wayne Woo
(510) 262-5045
waynewoo@slip.net

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 registered trademarks or trademarks of their respective companies.