MWSUG 2011 - Paper SA-07

The Full Monty

Joe Roma, Roma Statistical Consulting, Racine, WI
Irvin Snider, Assurant Health, Milwaukee, WI

Abstract

You are on the game show *Let's Make a Deal* and Monty asks you to select one of three doors that hide a prize behind it. Initially, you have a one in three chance of winning a Cadillac and a two in three chance of winning a goat. After you make a selection, Monty opens a door different from the one you have picked and reveals a goat. There are now two doors left, one hiding a goat and the other hiding a Cadillac. Monty now asks you if you want to stay with your original pick or if you want to switch.

In the classic Monty Hall problem, Monty will always open a door with a goat behind it. This being the case, it is always better to switch. By switching, you will win two out of three times as opposed to sticking with your original selection and winning one out of three times.

But what if Monty is drunk? What if Monty forgets where he put the Cadillac? What if Monty is obstinate? Or benevolent? What if Monty is just plain evil? What do you do then?

This paper reviews the classic Monty Hall Problem and several methods to solve the problem including a SAS® Monte Carlo method. The paper will then consider several variations of the problem and a SAS® method to solve each variation.

Classic Monty Hall Problem

Monty Hall was the host of the game show *Let's Make a Deal*. Monty would go into a studio audience filled with potential contestants dressed in outlandish costumes and trade with them. The contestants had to weigh offers by Monty as either being a good prize or a "zonk". For instance, Monty would offer $50 to anyone who had a hard boiled egg. After the exchange of cash for the egg, Monty would then ask the contestant if he would like to trade the money for a potentially better but unknown prize. The contestant could either keep what he had already won or trade it for something that could be a lot better. The catch, of course, is that the trade could result in getting something a lot worse. One could wind up with a goat, the classic zonk.

The classic Monty Hall problem presents you with three identical doors. Two of the doors conceal a goat and one conceals a Cadillac. Your goal is to win the Cadillac. You select a door at random but do not open it. Monty opens a door different from the one that you picked and reveals a goat. At this point of the game, Monty always opens a door containing a goat. It is important to note that Monty knows what prizes are behind the doors and that it is his intention to zonk you. Now, Monty gives you the option of sticking with your original selection or switching to choose the remaining unopened door. You win whatever is behind your final choice. What should you do? Do you switch or do you stick? Does it make a difference?

One would think that the odds are 50/50: there are two doors left after Monty opened a door and revealed a goat. That means that one of the two remaining doors has a Cadillac behind it and you have an even chance of winning it. So it seems that it does not matter if you switch or if you stick to your original choice. You might just as well stick with your initial pick. This seems reasonable.

However, it is actually better to switch rather than stick. The odds are two out of three that if you switch your pick, you will win the Cadillac.

Huh? How can that be? The fact that by merely switching your pick increases the odds of winning seems counterintuitive.

Solutions for Classic Monty

There are several ways to solve the classic Monty Hall problem and prove that it is best to switch your pick. Some of the favorites are listed in the following sections.

Classic Monty Hall Problem Solution Number One

Let's say that instead of three doors there are one million. You pick a door at random since all the doors are identical and there is no indication that any of them differ from any of the others. So you pick door number one. Monty then begins to open doors one at a time, asking you along the way if you want to stick or if you want to switch. You are obstinate and you stick with door number one. Eventually there are two doors left: your original pick and in this example, door number 777,777.

Doesn't it seem pretty obvious that it is better to switch at this point? Your original pick had a one in a million chance of being correct. You'd have to be pretty darn lucky to pick the winning door at the very beginning of this scenario. To make this argument more convincing, instead of one million doors make it one billion. Still not convinced? Try a trillion.

Classic Monty Hall Problem Solution Number Two

Another solution consists of a grid of possible outcomes. In the grid, we assume that you pick door number one. Monty will then open the door containing the goat behind either door two or three. According to the grid, when you switch, you win two times out of three. When you stick, you only win one time out of three.

Choose Door One and Stick with Door One			
Door One	Door Two	Door Three	Outcome
Cadillac	Goat	Goat	Win
Goat	Cadillac	Goat	Lose
Goat	Goat	Cadillac	Lose

Choose Door One and Switch			
Door One	Door Two	Door Three	Outcome
Cadillac	Goat	Goat	Lose
Goat	Cadillac	Goat	Win
Goat	Goat	Cadillac	Win

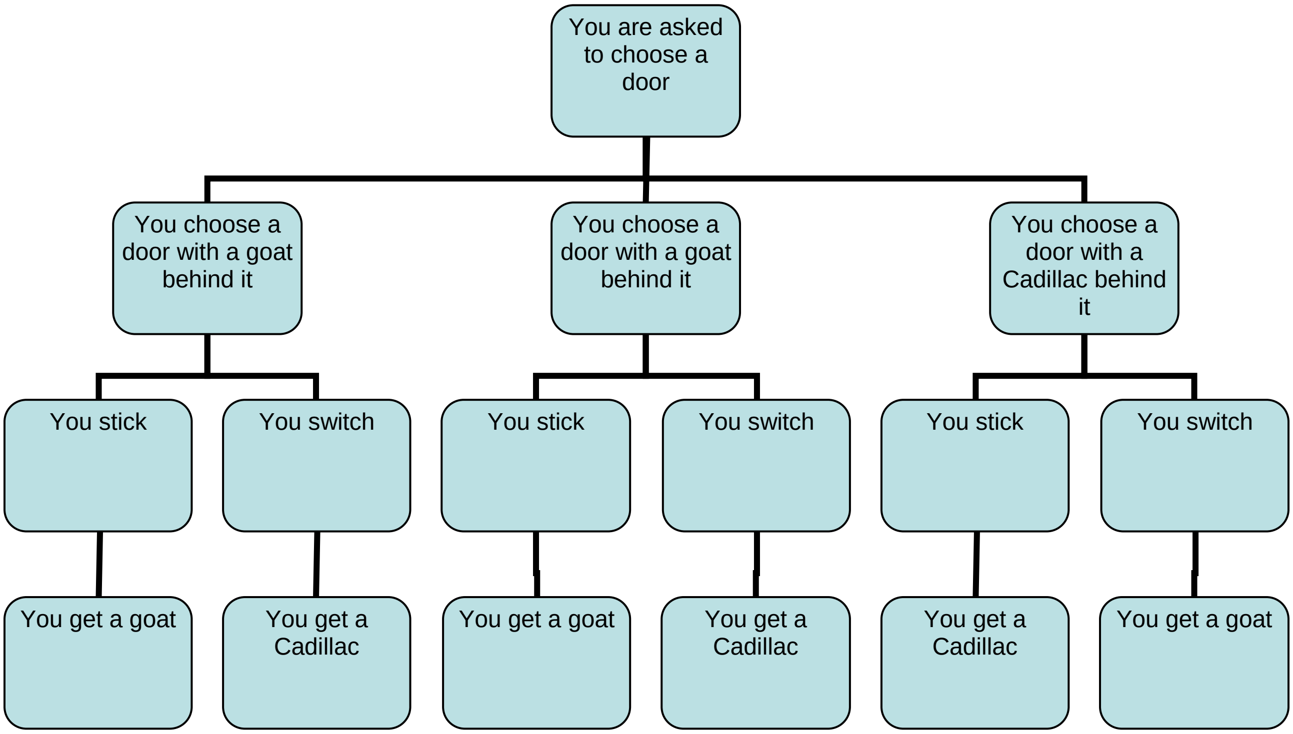
Classic Monty Hall Problem Solution Number Three

Let's say that your original pick is door number one. Now Monty opens a door and reveals a goat. Imagine that just after Monty opened the door and revealed a goat, a UFO lands on the stage and a little green woman emerges. Without knowing what you originally chose, she is asked to pick one of the two unopened doors. The odds of her picking the winning door are fifty-fifty. It is fifty-fifty for her because she lacks the advantage of the original contestant—the help of Monty.

Monty boosts the original contestant's odds by always opening a door that contains a goat. If the Cadillac is behind door number two, Monty reveals the goat behind door number three. If the Cadillac is behind door number three, Monty reveals the goat behind door number two. So when you switch, you win the prize if it's behind either door number two or three. *YOU WIN EITHER WAY*. But if you do not switch, you only win if the prize is behind door number one, your original pick. So your odds of winning are two out of three if you switch. The little green woman did not have this help when she arrived on stage so her odds of winning are merely fifty-fifty.

Classic Monty Hall Problem Solution Number Four

This method involves a decision tree. You start by picking a door at the very top of the tree. The second level of the tree indicates the three possible outcomes of your initial decision. At this point, Monty gives you the option of sticking or switching. The third level of the tree indicates the choices that you can make, which again are switch or stick. The fourth level lists the outcomes of your decisions. As you can hopefully see, if you had switched, you win a Cadillac in two out of three cases.



Classic Monty Hall Problem Solution Number Five

Bayes’ Theorem tells us how to properly recalculate the odds after we are given new information. By opening the door that reveals a goat, Monty is disclosing that the Cadillac is not behind that door. This changes the odds. We now have an advantage that the Little Green Woman did not have.

When you apply Bayes' formula to the Monty Hall problem, you begin with an initial value for the probability attached to a proposition that the prize is behind the not chosen door B, say, namely 1/3. This is the prior probability. Then you modify that probability assessment based on the new information you receive (in this case, the opening of door C to reveal that there is a goat behind it) to give a revised, or posterior probability for that proposition, which works out to be 2/3. Bayes’ Theorem tells us how to properly recalculate the odds after we are given new information. By opening the door that reveals a goat, Monty is disclosing that the Cadillac is not behind that door. This changes the odds.

Bayes’ theorem: $P(A|B) = (P(B|A)*P(A))/P(B)$

$P(A)$ is the prior probability.
$P(B)$ is the marginal probability
$P(A|B)$ is the conditional probability of A, given B
$P(B|A)$ is the conditional probability of B, given A

Initially when the game started, the odds of picking the Cadillac are one out of three. This is known as the prior probability.

Let’s say you pick door number one. You have a one out of three chance in winning a Cadillac. The prior probability is 1/3.

Monty then has the choice of opening doors two or three. Remember that Monty knows which doors the goats and the Cadillac are behind. Keep in mind that Monty will never select the door hiding the Cadillac.

If the Cadillac is behind door number one, then the odds of Monty opening door number two are one in two. It does not matter which door he opens since he will reveal a goat in either case.

If the Cadillac is behind door two, then Monty is not going to open it so the odds of him opening door number two are zero.

If the Cadillac is behind door number two, he is certain to open door number three so the odds of him opening door number three are one.

The probability of Monty then opening door number two is:

(1/3 * 1/2) + (1/3 * 0) + (1/3 * 1) = 1/2

This is the marginal probability.

Plug the marginal probability into Bayes' Theorem in order to calculate if it is better to stick:

The probability of the Cadillac being behind door number one given that door number two has been opened is the prior probability times the probability of Monty opening door number two if the prize is behind door number one divided by the marginal probability

Or

(1/2 * 1/3) / 1/2 or .333

Plug the marginal probability into Bayes' Theorem in order to calculate if it is better to switch:

The probability of the prize being behind door number three given that Monty has opened door number two is the prior probability times the probability of Monty opening door number two if the prize is behind door number three divided by the marginal probability

Or

(1 * 1/3) / 1/2 or .667

As you can see, it is better to switch.

Classic Monty Hall Problem Solution Number Six

We can also just observe the outcomes of the thousands upon thousands of Monty Hall games and add them up to see if it is better to stick or switch. Or we can have a computer simulation do that for us.

What follows is a Monte Carlo simulation written in SAS®. This code produces the classic output.

```
/*Full Monty Macro*/

%macro FULL_MONTY(trials = 1000000);

       data trials;
       do i = 1 to &trials;

              /*determine winning door*/
              PRIZE_RND = ranuni(11);
              if PRIZE_RND <= 1/3 then PRIZE_DOOR = 1;
              else if PRIZE_RND <= 2/3 then PRIZE_DOOR = 2;
              else PRIZE_DOOR = 3;

              /*determine first pick*/
              PICK_RND = ranuni(13);
              if PICK_RND <= 1/3 then FIRST_PICK = 1;
              else if PICK_RND <= 2/3 then FIRST_PICK = 2;
              else FIRST_PICK = 3;

              /*determine revealed door*/
              /*case FIRST_PICK is a winner*/

              REVEAL_RND = ranuni(15);
              if FIRST_PICK = PRIZE_DOOR then do;
              CASE = 'PICK PRIZE DOOR';
                     if REVEAL_RND <= 1/2 then do;
```

```
                if PRIZE_DOOR = 1 then REVEAL_DOOR = 2;
                if PRIZE_DOOR = 2 then REVEAL_DOOR = 1;
                if PRIZE_DOOR = 3 then REVEAL_DOOR = 1;
            end;
            else do;
                if PRIZE_DOOR = 1 then REVEAL_DOOR = 3;
                if PRIZE_DOOR = 2 then REVEAL_DOOR = 3;
                if PRIZE_DOOR = 3 then REVEAL_DOOR = 2;
            end;
        end;

        /*determine revealed door*/
        /*case FIRST_PICK not a winner*/

        if FIRST_PICK not = PRIZE_DOOR then do;
        CASE = 'NOT PICK PRIZE DOOR';
        REVEAL_DOOR = 6 - FIRST_PICK - PRIZE_DOOR;
        end;

        /*output 1 row for switch and 1 row for stay*/
        STRATEGY = 'STAY  ';
        FINAL_DOOR = FIRST_PICK;
        WIN_IND = (FINAL_DOOR = PRIZE_DOOR);
        output;

        STRATEGY = 'SWITCH';
        FINAL_DOOR = 6 - FIRST_PICK - REVEAL_DOOR;
        WIN_IND = (FINAL_DOOR = PRIZE_DOOR);
        output;

    end;
    run;

    /*output results*/

    proc sort data = trials;
    by STRATEGY;
    run;

    proc freq data = trials;
    by STRATEGY;
    tables WIN_IND;
    run;

%mend FULL_MONTY;

%FULL_MONTY(trials = 1000000);
```

Strategy = Stay

WIN_IND	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	665636	66.56	665636	66.56
1	334364	33.44	1000000	100.00

Strategy = Switch

WIN_IND	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	334364	33.44	334364	33.44
1	665636	66.56	1000000	100.00

Once again, the output indicates that it is better to switch.

Monty Variations

As we know, in the classic Monty Hall problem, Monty knows which prizes are hidden behind each door. We also know that Monty will always reveal a goat when given the chance after the contestant makes his initial pick. We know that Monty is trying to zonk the contestant.

Let us now consider variations to the classic Monty Hall problem.

Monty is Drunk!

Monty just got back from the bar, and you can smell the whiskey on his breath. You pick a door, and Monty mumbles that he can't remember where the car is anymore, but shows you a goat anyway (by sheer luck). Now should you switch? The following code adjusts the classic scenario for this variation. Shown below are the modified code sections.

The section to determine the revealed door is modified as follows:

```
/*determine revealed door*/
              /*random between 2 remaining doors*/

              REVEAL_RND = ranuni(15);
                     if REVEAL_RND <= 1/2 then do;
                            if FIRST_PICK = 1 then REVEAL_DOOR = 2;
                            if FIRST_PICK = 2 then REVEAL_DOOR = 1;
                            if FIRST_PICK = 3 then REVEAL_DOOR = 1;
                     end;
                     else do;
                            if FIRST_PICK = 1 then REVEAL_DOOR = 3;
                            if FIRST_PICK = 2 then REVEAL_DOOR = 3;
                            if FIRST_PICK = 3 then REVEAL_DOOR = 2;
                     end;
```

Note that now the revealed door is simply chosen at random between the two remaining doors, merely as a function of FIRST_PICK.

Since we know from the way the variation is stated above that Monty did not show us a car, we also need the following additional data step at the end of the macro, to delete the cases where PRIZE_DOOR = REVEAL_DOOR:

```
/*revealed door = the car*/
       data trials;
       set trials;
              if PRIZE_DOOR = REVEAL_DOOR then delete;
       run;
```

Presumably in those cases Monty revealed the car to us, and we won by default, but we are not in that universe of observations, hence the deletion of them.

We end up with the rather unintuitive result that it no longer matters if we switch, since we lose the benefit of Monty's knowing where the car is when he picks the second door to open. The results demonstrate this below.

Strategy = Stay

WIN_IND	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	333152	49.91	333152	49.91
1	334364	50.09	667516	100.00

Strategy = Switch

WIN_IND	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	334364	50.09	334364	50.09
1	333152	49.91	667516	100.00

Monty Sobers Up

At this point it is becoming clear that the value of switching after the first goat is revealed is directly related to Monty's certainty level with regard to where the goat is. The above "drunk" variation demonstrated that if Monty picks at random from the remaining two doors, it has no impact on the outcome to switch. We now introduce a more generalized form of the classic Monty macro, in which p represents how likely it is Monty will reveal the goat, given that you did not pick the car to begin with. If you did pick the car to begin with, we assume Monty simply picks at random from the remaining goats (perhaps with a slight hunch that you chose correctly as p increases). Note that p = 1 means Monty is 100% certain where the goat is, and reduces to the classic scenario. Further, note that p = .5 means that Monty is picking the door at random which represents the "drunk" scenario we have shown above.

```
%macro MONTY_UNCERTAIN(trials = 1000000, p = 1);

       data trials;
       do i = 1 to &trials;

              /*determine winning door*/
              PRIZE_RND = ranuni(11);
              if PRIZE_RND <= 1/3 then PRIZE_DOOR = 1;
              else if PRIZE_RND <= 2/3 then PRIZE_DOOR = 2;
              else PRIZE_DOOR = 3;

              /*determine first pick*/
              PICK_RND = ranuni(13);
              if PICK_RND <= 1/3 then FIRST_PICK = 1;
              else if PICK_RND <= 2/3 then FIRST_PICK = 2;
              else FIRST_PICK = 3;

              /*determine revealed door*/
              /*case FIRST_PICK is a winner*/

              REVEAL_RND = ranuni(15);
              if FIRST_PICK = PRIZE_DOOR then do;
              CASE = 'PICK PRIZE DOOR';
                     if REVEAL_RND <= 1/2 then do;
                            if PRIZE_DOOR = 1 then REVEAL_DOOR = 2;
                            if PRIZE_DOOR = 2 then REVEAL_DOOR = 1;
                            if PRIZE_DOOR = 3 then REVEAL_DOOR = 1;
                     end;
                     else do;
                            if PRIZE_DOOR = 1 then REVEAL_DOOR = 3;
                            if PRIZE_DOOR = 2 then REVEAL_DOOR = 3;
                            if PRIZE_DOOR = 3 then REVEAL_DOOR = 2;
                     end;
```

```
            end;

            /*determine revealed door*/
            /*case FIRST_PICK not a winner*/

            if FIRST_PICK not = PRIZE_DOOR then do;
                  CASE = 'NOT PICK PRIZE DOOR';
                  if REVEAL_RND <= &p then
                  REVEAL_DOOR = 6 - FIRST_PICK - PRIZE_DOOR;
                  else REVEAL_DOOR = PRIZE_DOOR;
            end;

            /*output 1 row for switch and 1 row for stay*/
            STRATEGY = 'STAY  ';
            FINAL_DOOR = FIRST_PICK;
            WIN_IND = (FINAL_DOOR = PRIZE_DOOR);
            output;

            STRATEGY = 'SWITCH';
            FINAL_DOOR = 6 - FIRST_PICK - REVEAL_DOOR;
            WIN_IND = (FINAL_DOOR = PRIZE_DOOR);
            output;

      end;
      run;

      /*revealed door = the car*/
      data trials;
      set trials;
            if PRIZE_DOOR = REVEAL_DOOR then delete;
      run;

      /*output results*/

      proc sort data = trials;
      by STRATEGY;
      run;

      proc freq data = trials;
      by STRATEGY;
      tables WIN_IND;
      run;

%mend MONTY_UNCERTAIN;
```

In the above, we have introduced a section of code which determines the REVEAL_DOOR as a function of Monty's certainty about the location of the goat. Specifically in the section

```
/*determine revealed door*/
            /*case FIRST_PICK not a winner*/

            if FIRST_PICK not = PRIZE_DOOR then do;
                  CASE = 'NOT PICK PRIZE DOOR';
                  if REVEAL_RND <= &p then
                  REVEAL_DOOR = 6 - FIRST_PICK - PRIZE_DOOR;
                  else REVEAL_DOOR = PRIZE_DOOR;
            end;
```

we cause Monty to reveal the goat in the expression REVEAL_DOOR = 6 - FIRST_PICK - PRIZE_DOOR with probability p. Otherwise we have the case where Monty inadvertently reveals the PRIZE_DOOR. So we see that when p = 1, we are in the classic case in which Monty will always reveal the goat when you have not chosen the car. When p = .5 we have a 50% chance of either door being opened, in both cases of our first pick, hence reducing to the "drunk" variation. To demonstrate, here are the results of

```
%MONTY_UNCERTAIN(trials = 1000000, p = 1);
```

Strategy = Stay

WIN_IND	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	665636	66.56	665636	66.56
1	334364	33.44	1000000	100.00

Strategy = Switch

WIN_IND	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	334364	33.44	334364	33.44
1	665636	66.56	1000000	100.00

Note the conclusion is the same as the classic Monty scenario.

And here are the results of

```
%MONTY_UNCERTAIN(trials = 1000000, p = .5);
```

Strategy = Stay

WIN_IND	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	332410	49.85	332410	49.85
1	334364	50.15	666774	100.00

Strategy = Switch

WIN_IND	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	334364	50.15	334364	50.15
1	332410	49.85	666774	100.00

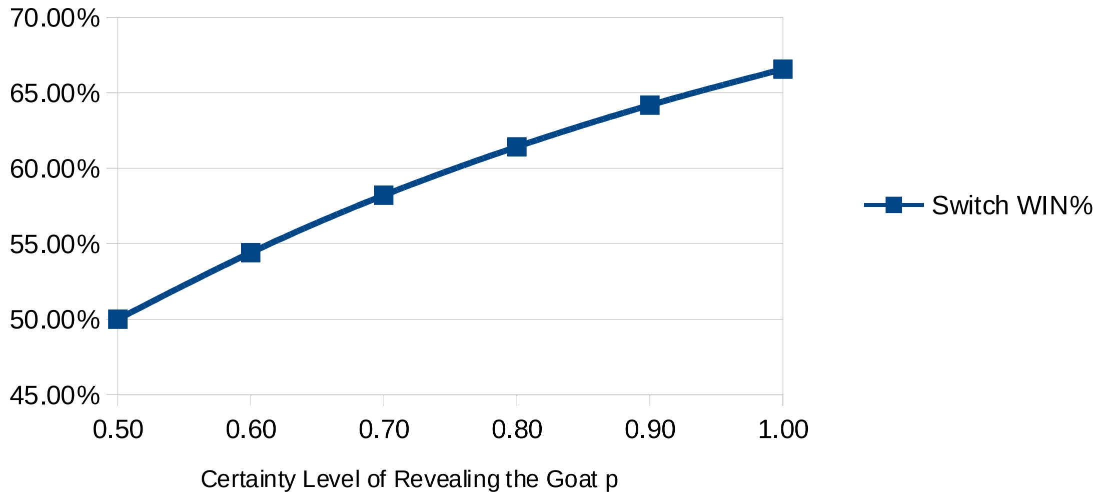
Note the conclusion is the same as the drunken Monty scenario.

Graphic Monty

Now let's look at the value of switching as Monty sobers up. The following is a graph showing the switch winning % as p goes from .5 (completely drunk) to 1 (completely sober). In this way we graphically quantify the relationship of Monty's certainty with the value of switching doors.

```
%MONTY_UNCERTAIN(trials = 1000000, p = .5);
%MONTY_UNCERTAIN(trials = 1000000, p = .6);
%MONTY_UNCERTAIN(trials = 1000000, p = .7);
%MONTY_UNCERTAIN(trials = 1000000, p = .8);
%MONTY_UNCERTAIN(trials = 1000000, p = .9);
%MONTY_UNCERTAIN(trials = 1000000, p = 1);
```

Switching Win % by Monty's Certainty Level for the Goat



Conclusion

Six different solutions to the classic Monty Hall problem were presented. The solutions ranged from common sense to Bayes' Theorem to a Monte Carlo simulation.

Variations to the Monty Hall problem were presented and these variations were analyzed with SAS® Monte Carlo simulations. Monty's uncertainty level with regard to the remaining goat was shown to be directly related to the value of switching upon the revealing of the remaining goat.

Monty Variations

Possible host behaviors in unspecified problem	
Host behavior	**Result**
"Monty from Hell": The host offers the option to switch only when the player's initial choice is the winning door.	Switching always yields a goat.
"Angelic Monty": The host offers the option to switch only when the player has chosen incorrectly.	Switching always wins the car.
"Monty Fall" or "Ignorant Monty": The host does not know what lies behind the doors, and opens one at random that happens not to reveal the car.	Switching wins the car half of the time.
The host knows what lies behind the doors, and (before the player's choice) chooses at random which goat to reveal. He offers the option to switch only when the player's choice happens to differ from his.	Switching wins the car half of the time.
The host always reveals a goat and always offers a switch. If he has a choice, he chooses the leftmost goat with probability p (which may depend on the player's initial choice) and the rightmost door with probability $q=1-p$.	If the host opens the rightmost door, switching wins with probability $1/(1+q)$.
The host acts as noted in the classic version of the problem.	Switching wins the car two-thirds of the time. (Special case of the above with $p=q=½$)
The host opens a door and makes the offer to switch 100% of the time if the contestant initially picked the car, and 50% the time if she didn't.	Switching wins 1/2 the time at the Nash equilibrium.
Four-stage two-player game-theoretic. The player is	Minimax solution Nash equilibrium: car is first hidden

playing against the show organizers (TV station) which includes the host. First stage: organizers choose a door (choice kept secret from player). Second stage: player makes a preliminary choice of door. Third stage: host opens a door. Fourth stage: player makes a final choice. The player wants to win the car, the TV station wants to keep it. This is a zero-sum two-person game. By von Neumann's theorem from game theory, if we allow both parties fully randomized strategies there exists a minimax solution or Nash equilibrium.	uniformly at random and host later chooses uniform random door to open without revealing the car and different from player's door; player first chooses uniform random door and later always switches to other closed door. With his strategy, the player has a win-chance of at least 2/3, however the TV station plays; with the TV station's strategy, the TV station will lose with probability at most 2/3, however the player plays. The fact that these two strategies match (at least 2/3, at most 2/3) proves that they form the minimax solution.
As previous, but now host has option not to open a door at all.	Minimax solution: car is first hidden uniformly at random and host later never opens a door; player first chooses a door uniformly at random and later never switches. Player's strategy guarantees a win-chance of at least 1/3. TV station's strategy guarantees a lose-chance of at most 1/3.

References

Fan, Xitao, Felsovalyi, Ákos, Sivo, Stephen A., Keenan, Sean C., SAS® for Monte Carlo Studies, 2001, SAS Institute Inc., Cary, NC

Haddon, Mark, The Curious Incident of the Dog in the Night-time, 2003, Random House, Inc., New York, NY

Hoffman, Paul., The Man Who Loved Only Numbers: The Story of Paul Erdös and the Search for Mathematical Truth, 1998, Hyperion, New York, NY

McGrayne, Sharon Bertsch, The Theory That Would Not Die: How Bayes' Rule Cracked the Enigma Code, Hunted Down Russian Submarines, and Emerged Triumphant from Two Centuries of Controversy, 2011, Yale University Press, New Haven, CT

Rosenhouse, Jason., The Monty Hall Problem: The Remarkable Story of Math's Most Contentious Brainteaser, 2009, Oxford University Press, Inc., New York, NY

Vazsonyi, Andrew., Which Door has the Cadillac, 2002, Writers Club Press, Lincoln, NE

For a discussion of the Monty Hall Problem and Bayes Theorem visit:
http://www.maa.org/devlin/devlin_12_05.html

Acknowledgements

Thanks to Emma Snider, Editorial/Marketing Assistant at Pearson Education for proofreading and editorial suggestions.

Irvin thanks Deanna Nguyen for her encouragement and assistance in Irvin's ever continuing pursuit to improve his communication skills.

Irvin thanks Andrea Everling for her help with the section on Bayes' Theorem.

Contact Information

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

Joe Roma
Roma Statistical Consulting
3414 Oakwood Dr
Racine, WI 53406
Phone: 262-995-8195
E-mail: joeyroma@gmail.com
Web: www.linkedin.com/pub/joseph-roma/1/2b7/973

Irvin Snider
Assurant Health
501 West Michigan
Milwaukee, WI 53201-3050
Phone: 414-299-6979
Fax: 414-299-8043
E-mail: irv.snider@assurant.com
Web: www.linkedin.com/in/irvinsnider

Certified Advanced Programmer for SAS 9