Social Influences Involved in Teenage Gambling Behavior

New Mexico Adventures in
Supercomputing Challenge
Final Report

April 6, 2005

Team Number 081
Sliver High School, Silver City, NM

Team Members:
Benjamin Fox, Sophomore
Kyle Youngs, Sophomore

Teacher: Mrs. Peggy Larisch
Mentor: Mr. Berry F. Estes

Acknowledgements

The authors wish to acknowledge the following individual for their guidance and assistance provided during this project:

- Mrs. Peggy Larisch – Teacher, Silver High School, Advanced Computers
- Mr. Berry F. Estes – Nuclear Engineer (Retired), Sandia National Laboratories

Table of Contents

E.0 Executive Summery

Gambling is a major problem facing today's society. In recent years, the increase in teenage gambling has caused many concerns. Peer models have been suggested as important etiological factors in the development of high-risk behavior including gambling.

This study contains three hypotheses: (1) group involvement will increase the betting patterns; (2) males will be more influential in gambling activities; and (3) males will wager more than females as they are greater risk-takers by nature.

The team created three different gambling simulation games (roulette, 21, and slots) in order to observe participants' playing behavior. An IBM laptop was used in order to render the games as realistic as possible.

The population selected for this study consisted of 60 high school students from southern New Mexico. The participants were 30 males and 30 females. All students were between the ages of 14 and 18 and were all defined as inexperienced gamblers (e.g. defined as never playing in a casino before). The students participated in the gambling simulations and the results were recorded by using C++ programs created by members of the team and sent to a text file to allow for the data to be read.

The student participants were then randomly placed in groups: (1) with a member of the same gender and (2) with a member of the opposite gender. The gambling simulations were conducted again with the results being recorded by the C++ programs.

Using Microsoft Excel, a Chi-squared statistical test was performed along with finding the mean and standard deviation of each game. It was found by summing the amount bet divided by the times played. In a previous evaluation of this project, it was determined that the data be checked for statistical significance by using the Chi-squared statistical test.

After analyzing the data and checking its validity with the Chi-square test, all three hypotheses were found to be correct because the results showed statistical significance.

1.0 Introduction

1.1 Purpose

Gambling is a major problem facing today's society. Risky gambling is observed in more than 15 million United States citizens and is very common in young people [1]. In the results of a survey conducted through the United States and Canada, 15%, of all surveyed teens admitted to having problems related to gambling. Gambling is the fastest growing form of entertainment in the United States [2]. Between the years of 1997 and 1998, the number of online gamblers increased from 6.9 million to 14.5 million and grossed over $652 million dollars for the Internet gambling houses [1]. Gambling is legal in all but two states and costs the U.S. government an estimated $5 billion dollars in social services (health care, bankruptcy, criminal costs) [1]. Gambling is also legal in all of the Canadian provinces and 90 other countries thought the world [3]. American citizens have gambled with over $482 billion dollars in casinos, Internet gambling houses, and racetracks [1]. Another $31.5 billion was spent gambling on state lottery games and casinos' influences are growing [1].

In recent years, there have been growing concerns over the increasing occurrence of teenage gambling problems. This is likely the results of availability and accessibility of legalized gambling. Gambling is legal in 48 of the 50 United States [1]. There is also evidence that gambling causes personal and financial difficulties in at least 2% of the adult population [3].

Current prevalence rates of adolescent pathological gambling indicate that between 4% and 8% of adolescents exhibit compulsive or pathological patterns of gambling activity [4]. Gambling behavior seems to be established earlier than other addictions in young people today [5]. Since there are few observable signs of gambling dependence among young people, these problems have gone unnoticed compared to other addictions [5]. Although betting in casinos and

electronic gambling machines, and lotteries in general are illegal for adolescents; the enforcement of these laws can be difficult.

Peer models have been suggested as important etiological factors in the development of high-risk behavior including gambling [6]. These findings indicate that there may be a relationship between gambling and other addictions as well as a strong social learning component involved in the acquisition of such behaviors.

Gambling activities have a social reinforcement. Gambling behaviors can be encouraged and strengthened by peers depending on the individual's developmental level and social status. From a social learning theory perspective individuals learn, acquire, and preserve with behaviors that are attractive and reinforcing. A social learning model of gambling behavior provides an explanation for why many youth are attracted to the gambling environment. Being known as a gambler or risk-taker gives a teenager the social recognition among his/her peers [6]. Gambling becomes a form of an arcade for many young people today. It is a "hangout" activity where they can meet friends.

Studies suggest that gambling is more popular with males than females. The principal goal of this study was to examine the social influences involved in the gambling behavior of a typical teenager via interactive play. Given that modeling has been shown to be a strong learning component in teenagers, it is likely that it is also instrumental in the acquisition and maintenance of gambling behavior. The fact that gambling has many reinforcing qualities, especially those that are social in nature, makes gambling a very rewarding pastime. With the increase in adolescent gambling activities these notions are a vital concern.

The hypotheses of this study are: (1) group involvement will increase the betting patterns; (2) males will be more influential in gambling activities; and (3) males will wager more than females as they are greater risk-takers by nature.

1.2 Scope

The group used in this study included 60 high school students from southern New Mexico (30 females and 30 males) [Appendix A]. The students participated in the gambling simulations and the results were recorded by using C++ programs [Appendix B] created by members of the team and sent to a text file.

The student participants were then randomly placed in groups: (1) with a member of the same gender and (2) with a member of the opposite gender. The gambling simulations were conducted again with the results being recorded by the C++ programs.

1.3 Computer Program

The team created three different gambling simulation games (roulette, 21, and slots) [Appendix B] in order to observe participants' playing behavior. An IBM laptop was used in order to render the games as realistic as possible. All three games involved changed as well as active choices. The participants were provided with simple simulations so that explanation of the games was easy for all participants. There was no limitation on the betting but once the participant had used all of his allowed bankroll, the games would end.

2.0 Project Proposal

2.1 Description of Project

The population for this project consisted of 60 high school students from southern New Mexico. The participants were 30 males and 30 females. All students were between the ages of 14 and 18 and were all defined as inexperienced gamblers (e.g. defined as never playing in a casino before). The

participants then played one or more of the three-origonal programs written in C++ (21, roulette, and slots) individually. Each participant then played in two groups, one with a member of the same sex as one group and one member of the opposite sex for the other. These groups simulated a gambling environment based in a social atmosphere. The groups were forced to cooperate while gambling and agree on decisions that had to be made (e.g. bets, number of cards wanted)

Each participating individual played one or all of the gamboling simulations (21,slots, and roulette) written in C++. An IBM Think Pad laptop was used to run these programs. The players ID number for baseline, same, and opposite groups were fed into a text file along with the number of times bet and how much bet each time. A spreadsheet was then crated to store the data and to find the mean of each participant, the mean, and standard deviation for each game, and to perform a Chi-square statistic test to verify the validity of the data for each game.

3.0 Analytical Methodology

3.1 Mathematical Bases

The participants of the sample population were not limited to how many times they were allowed to play each of the gambling games (Game 21, Slots, and Roulette); the mean (average) of all the bets were used for each game played [Appendix C].

The mean for the bets of the sample group was found by using the mathematics formula:

$$\text{Mean} = (\text{bet}(1)+\text{bet}(2)+\text{bet}(3)+\ldots.+\text{bet}(n))/n$$

where n is the number of times played.

An Excel spreadsheet was used to record and calculate all of the participants' data (e.g. number of and how much each bet was) on all games played [Appendix A]. Each participant had three means (playing alone, playing with a participant of the same gender, and playing with a participant of the opposite gender) for each game played [Appendix C]. Sixty worksheets (one per participant) were created for documenting this data [Appendix A].

In the Excel spreadsheet, three other worksheets were created to analyze the data for each game played. The means from each participant were copied to these worksheets along with the gender and identity number given each participant. From this data, the mean and standard deviation were calculated.

The mean was found by using the mathematic formula:

$\bar{X} = (X(1)+ X(2)+ X(3)+\ldots.+X(n))/n$

where n is the number of participants and, $X_{()}$ is the mean of each participant.

Since this model is dealing with a sample that is being used to estimate properties of a population, the standard deviation formula is:

Standard deviation = $(((X_1- \bar{X})+(X_2 - \bar{X}) + (X_3 - \bar{X})\ldots.+(X_n- \bar{X})) / (n-1))x(1/2)$

Where n is the number of participants, $X_{()}$ is the mean of each participant, and $\bar{X}$ is the mean of all of the means

3.2 Chi-squared Statistic

During prior evaluation for this project, it was determined that the data be checked for statistical significance by using the Chi-squared statistical test [Appendix D]. This test is a measure that combines the strength of the relationship with information about the size of the sample to give one summary number (e.g. the validity of your data). If the summary number is equal to or larger than the “magic” number, the relationship in the table is considered to be statistical significant. The origin of the “magic” number comes from a table of percentiles representing what should happen by chance. The interpretation of the value of this “magic” number is straightforward. If a relationship that has a chi-squared statistic larger than the “magic” number is observed, one can assume that the relationship in the sample did not occur by chance. In this case, one can say that the relationship is statistically significant. Of all relationships that have occurred just by chance, 5%of them will erroneously earn the title of statistically significant. However, if the size of the sample is too small, a real relationship may not be detected. The chi-squared statistic depends on both the strength of the relationship and the size of the sample.

The actual computation and assessment of statistical significance was tedious but not mathematically difficult. There are many methods. The method chosen for this model comes from the book, Seeing Through Statistics by Jessica M Utts [7].

There are four basic steps:

- Compute the expected numbers.
- Compare the observed and expected numbers.
- Compute the chi-squared statistic.
- Make a decision.

Using Excel, worksheets were added to calculate the Chi-squared statistic. To help the team to understand the mathematical behavior of the process, simple functions were used in the Excel program. The following steps were followed in sequence:

Step1: Compute the Expected Numbers:

The formula used for finding the expected number in any row and column combination is:

Expected number = (row total)x(column total) / (table total)

Step2: Compare the Observed and Expected Numbers:

Compute for each of the cells of the table:

(Observed number – Expected number)^2/(Expected number)

Step3: Compute the Chi-Squared Statistic:

Compute the chi-squared statistic, by adding the numbers in all of the cells from step 2. This result is the chi-squared statistic.

Step4: Make the Decision:

Using a statistics textbook [8], one would need to look up the appropriate number in a table called “percentiles of the chi-squared distribution” using the frequency table:

Frequency = (row-1) x (column-1)

number of participants minus one as the degrees of freedom value and a probability (p) of 0.99. The value located on the chart is the "magic" number used for decision making.

3.3 Computer Applications

All three programs (21, slots, roulette) were constructed using the C++ programming language. All of the programs are original and written specifically for this project. The programs call for the user to input their identification number (baseline #: 101-160, same #: 201-260, opposite #: 301-360) and if they would like to hear the rules. The user then continues on with the program and reacts to their oscillation in performance by changing the amount bet or to stop playing. The program outputs simple instructions between betting turns to allow the user to understand how to play (how to bet, spin quit ect.) and allow for smoother play. The programs are expected to output the participants' player ID number, the amount the participant bet each round played, along with how many times they played.

In each game, the participant began with a limited amount of money to allow for a more realistic gambling atmosphere. Once the participant has completed one betting round of the game, they are asked if they would like to play again. If they reply "yes" then the program calls up the "main" function through recursion to execute the game again. If the participant chooses "no", the program closes. Once the simulation has ended, the participant's performance data (player ID number, the amount the participant bet each round played, how many times they played) was recorded into a text file. A function was included in each game to output this data into text file. The results of the project can be obtained by looking at the results of the text file that was gathered from the programs.

4.0 Results

4.1 Computation of Data

Raw Frequency (defined as the initial variable string of data) is required in a Chi-Square test. The frequency of each game (21, slots, roulette) used baseline, same group , and opposite group bets to tally both male and female participants' results for increase in betting in social atmospheres. Raw frequency is directly related to two or more mutually exclusive categories. The categories used were (1) the comparison of male and female bets, (2) and that the interaction of males and females resulting in a social atmosphere causes the amount bet to increase from the baseline bet (see Tables 1-3).

Each game (21, slots, roulette, baseline, same, and opposite), an average amount bet was calculated by summing the amount bet divided by the times played. This was done for male and female groups. The average bet of the participants varied greatly between male and female baselines and the same and opposite groups related to the sex of the members of the group (sees Tables 4-6). Next, the average mean of female baseline, same, and opposite bets were calculated. The same was also found for the male bets (see Table 7). The average bets of both males and females were also calculated (see Table 8). Along with mean, the standard deviation of male and female baselines, same, and opposite bets were found (see Tables 9-11) [Appendix C].

A Chi-square statistics test was performed on the data to check the validity of the data and to help the team to understand the mathematical behavior of the process (see Tables 12-14).

4.2 Data Tables

Table 1 21-Frequency Chart

Frequency of social interaction of betting increasing average bet made by teenagers

	Yes (same)	No (same)	Yes (opposite)	No (opposite)	Total
female	13	17	22	8	60
male	26	4	15	15	60
	39	21	37	23	120

Table 2 Roulette-Frequency Chart

Frequency of social interaction of betting increasing average bet made by teenagers

	Yes (same)	No (same)	Yes (opposite)	No (opposite)	Total
female	7	23	23	7	60
male	26	4	23	7	60
	33	27	46	14	120

Table 3 Slots-Frequency Chart

Frequency of social interaction of betting increasing average bet made by teenagers

	Yes (same)	No (same)	Yes (opposite)	No (opposite)	Total
female	16	14	29	1	60
male	23	7	14	16	60
	39	21	43	17	120

Table 4 21-Mean Chart

Mean	Female		
	Baseline	Same	Opposite
	36	40	1120
	Male		
	Baseline	Same	Opposite
	380	46239	1303

Table 5 Roulette-Mean Chart

Mean	Female		
	Baseline	Same	Opposite
	28	20	114
	Male		
	Baseline	Same	Opposite
	37	250	115

Table 6 Slots-Mean Chat

Mean	Female		
	Baseline	Same	Opposite
	28	55	477
	Male		
	Baseline	Same	Opposite
	217	41234	468

Table 7 Female Male Average Bets

Total Gender Means		Baseline	Same	Opposite
	f	30.67	38.33	570.33
	m	211.33	20241	628.67

Table 8 Total Means

Total Means	Baseline	Same	Opposite
	121	14639.67	600

Table 9 21-Satandard Deviation Chart

Standard Deviation			
Female			
	Baseline	Same	Opposite
	46.47	71.72	2710.16
Male			
	Baseline	Same	Opposite
	737.91	167945.9	2787.45

Table 10 Roulette-Standard Deviation Chart

Standard Deviation

Female		
Baseline	Same	Opposite
10.33	4.47	130.62
Male		
Baseline	Same	Opposite
27.65	384.27	130.24

Table 11 Slots-Standard Deviation Chart

Standard Deviation

Female		
Baseline	Same	Opposite
15.58	73.55	676.01
Male		
Baseline	Same	Opposite
162.15	154230.2	677.15

Table 12 21-Chi-Square Test

Chi-Square Stat

	15.835711	df=(r-1)*(c-1)
P=.99	11.3	df=3

Table 13 Roulette-Chi-Square Test

Chi-Square Stat

	34.12458	df=(r-1)*(c-1)
P=.99	11.3	df=3

Table 14 Slots-Chi-Square Test

Chi-Square Stat

	22.0576	df=(r-1)*(c-1)
P=.99	11.3	df=3

5.0 Conclusion

5.1 Results

After analyzing the data, the following hypotheses were found to be correct:

1. Group involvement will increase the betting patterns.

   The mean of the average baseline bets was less than the mean average of the same and opposite bets (see Table 8). This data proves that when in a social atmosphere, teens bet higher then when playing as and individual.

2. Males will be more influential in gambling activities.

   When placed with a male group member, a females betting average wet up compared to when the same participant played with a member of the same sex (see Tables 4-7).

3. Males will wager more than females as they are greater risk-takers by nature.

   Average bets placed by male participants in their baseline and same games was higher then the average bets placed by female participants in their baseline and same games (see Table 4-7).

6.0 Future Work

6.1 Recommendations

To expand on the progress to date, it is recommended that the population be expanded to get a more accurate representation teenage gambling. It is also recommended that the population be extended to include adults in real gambling situations (e.g. casinos). Another recommendation is to expand the project to other activities (e.g. driving) to see if social influence is a factor in actions other then gambling. This would allow for a better understanding of how social atmospheres affect human behavior.

References

1. Unwin, Brian K.. Pathologic Gambling. American Family Physician. 2/14/05 <http://www.aafp.org/afp/20000201/741.html>.

2. NA, Pathological gambling: An addiction embracing the nation. Illinois Institute for Addiction Recovery. 11/23/04 <http://www.addictionrecov.org/aboutgam.htm>.

3. Lesiur, H. R., & Rosenthal, R. J. (1991). Pathological gambling: A review of the literature (prepared for the American Association Task Force on DSM-IV committee on disorders of impulse control not elsewhere classified).*Journal of Gambling Studies, 7 (1),439.*

4. Gupta, R., & Derevensky, J.L. (1998). Adolescent gambling behavior: A prevalence study and examination of the correlates associated with problem gambling. *Journal of Gambling Studies, 14 (4), 319345*

5. Arcuri, A. F., Lester, D., & Smith, F. O. (1985). Shaping adolescent gambling behavior. *Adolosce 20, 935938*

6. Shaffer, H.J., & Hall, M.N. (1996). Estimating the prevalence of adolescent gambling disorders: A quantitative synthesis and guide toward standard gambling nomenclature. *Journal of Gambling Studies , 12 (2), 193214.*

7. Utts, Jessica M.. Seeing Through Statistics. Pacific Grove: Brooks/Cole Publiching Company, 1999.

8. Downing, Douglas, Ph. D, and Clark Jeffrey, Ph. D.. Forgotten Statistics. Hauppauge: Barron's Educational Series, Inc., 1996.

Appendix A

Excel Spreadsheets

Appendix B

C++ Programs

Appendix B

```
// Game of 21
#include <iostream.h>
#include <stdlib.h>
#include <time.h>


//**********************************
// function
int random()
{
        //  Returns a random card value
        //Post: returns a random # from 1 to 10
  int card =0;
  int max =10;
  card =1+rand()%max;
  return (card);
}

//************************************
//**********************************
// function
int how_many()
{
        // ask user how_many cards
        // Post: gives how many cards the user wants
        int numCards;
        cout<<"How many cards do you want?";
        cin>>numCards;
        return (numCards);

}

//************************************
//**********************************
// function
int User( int NumberOfCards)
{
        //  PRE: receives the number of cards
        //uses a loop to get cards and add them up
        // print out cards
        //Post: outputs the total count of player's hand
    int x;
    int Ptotal=0;
      cout<<"Player:  ";
```

```
    for (int i = 0;i<NumberOfCards;i++)
    {   //begin for loop
        x= random();
        cout<<x<<" ";
        Ptotal = Ptotal +x;
     }  // end for loop
     cout<<endl;
     return (Ptotal);
}
//*********************************
// function
int Computer( int NumberOfCards)
{
          // PRE: receives the number of cards
          //uses a loop to get cards and add them up
          // prints out computers cards
          //Post: outputs the total count of computer's hand
    int x;
    int Ctotal=0;
    cout<<"Computer:   ";
    for (int i = 0;i<NumberOfCards;i++)
    {   //begin for loop
        x= random();

        cout<<x<<" ";
        Ctotal = Ctotal +x;
     }  // end for loop
     cout<<endl;
     return (Ctotal);
}        //end function computer
//*********************************
void Winner(int PT, int CT)
{       // begin function winner
          //Pre: brings in the players total and the computers total
          //Post: do not need to return a value
          if ((PT >21)&&(CT>21))
             cout<< "You both went bust"<<endl;
          if ((PT>CT)&&(PT<=21))
             cout<<"You win!!!"<<endl;
          if ((CT>PT)&&(CT<=21))
             cout<<"You lose, Computer wins!"<<endl;
          if (CT==PT)
             cout <<"Draw"<<endl;
          if ((CT>21)&&(CT>PT))
             cout<<"You win!!!"<<endl;
          if ((PT>21)&&(PT>CT))
```

```
            cout<<"You lose, Computer wins!"<<endl;
cout<<endl;
}      // end function winner
 //**********************************
 //**********************************
 // this function simply asks the participant if they would like to play the
 // game again. the integer "w" is the answer that the participant gives.  The
 // answer is then returned by the function for use in other functions.
char P_again()
 {
  char w;
 cout << "Do you want to play again?";
  cin>>w;
  return(w);
  }
//*************************************************
// this function is where all of the participants totals were added up.
// it also gets the integer "w" from the function above and if the reply was
// yes then the program goes through recursion and plays the game again until
// the participant chooses to stop playing the game in the function above.
void play ()
{
 cout << "CARD GAME "<<endl;
      int PlayerTotal = 0;
      int ComputerTotal =0;
      int x =0;
      int n=0;
      char ask;
      x = how_many ();
      PlayerTotal = User(x);
      ComputerTotal = Computer(x);
      Winner(PlayerTotal, ComputerTotal);
       ask =P_again();
     if ((ask =='y')||(ask =='Y'))
      {
        play();  // this is recursion - where a function calls itself

      }
 }   // end Play function
//***********************************
int main()
{
    srand ( (unsigned) time ( NULL ) ); // init Random() function

     play();
      system("PAUSE");
```

```
    return 0;
}
// Slots Program 1/12/05
//Kyle Youngs and Ben Fox
#include <iostream.h>
#include <stdlib.h>
#include <time.h>
#include <fstream.h>
int game();
int PrintRules();
int playernum();
//function*************************************************************
// this function prints the rules of the slot machine game to the participant.
// it outputs all of the rules that are to be followed by the participant and
// answers some of the questions the participant may have.
// the integer "an" is simply the answer that the user reply's when the program
// asks them if they would "like to pull" if the participant answers with a
// yes response then it goes into another function for the game. Otherwise the
// program ends and all data is recorded by the text file.
int PrintRules()
{
int an;
cout<<endl;
cout<<"Would you like to pull?: ";
cout<<endl;
cin>>an;
if ((an='y')||(an='Y'))
{
playernum();
}
}
//function*************************************************************
// this function just asks the user for they're player id number. the integer
//"playernum" is the answer that the participant responds to when they are
// prompted to enter they're player ID. after this is entered by the participant
// then the function goes to the actual game itself.
int playerID ()
{
int playernum;
cout<<"Please enter your playerID#: ";
cin>>playernum;
cout<<endl;
cout<<endl;
{
game();
}
```

```
}
//function*************************************************************
// this function actually simulates the slot machine that the participant is
// playing on. there are many integers in this function because of all of the
// random numbers that are needed by the machine to determine if the participant
// wins or loses.  The integer "c" is the amount of money the participant starts
// with.  This integer is set equal to 200. integers "R11-R33" are all single
// numbers that will be outputted be the function to simulate the slot machine
// face.  All of these integers are randomly selected between the numbers 1-5.
// all of these integers have different spots in the slot itself. for example
// the integer "R11" stands for "R column 1 and row 1" and "R22" stands for
// "R" column 2 and row 2" or "R23" stands for "R coulmn 2 row 3". After all of
// these numbers are randomly selected by the function they are outputted by the
// program.  If anyo of these numbers match diagonally, straight down, straight
// up vertical or horizontal the participant is a winner.  All three of the
// numbers have to be the same for the participant to win the game.
int game ()
{
int c=200;
srand( (unsigned)time( NULL )); // init Rand() function
int R11=0+rand()%5;
int R12=0+rand()%5;
int R13=0+rand()%5;
int R21=0+rand()%5;
int R22=0+rand()%5;
int R23=0+rand()%5;
int R31=0+rand()%5;
int R32=0+rand()%5;
int R33=0+rand()%5;
cout<<endl;
cout<<"[   "<<R11<<   "]   [  "<<R12<<  "]  [  "<<R13<<  "]"<<endl;
cout<<"[   "<<R21<<   "]   [  "<<R22<<  "]  [  "<<R23<<  "]"<<endl;
cout<<"[   "<<R31<<   "]   [  "<<R32<<  "]  [  "<<R33<<  "]"<<endl;
cout<<endl;
cout<<endl;
 if ((R11==R12)&&(R12==R13))
{
cout<<endl;
cout<<endl;
cout<<"You win across in first row!"<<endl;
c=c+4;

}
if ((R21==R22)&&(R22==R23))
{
cout<<endl;
```

```
cout<<endl;
cout<<"You win across in middle row!"<<endl;
c=c+4;

}
if ((R31==R32)&&(R32==R33))
{
cout<<endl;
cout<<endl;
cout<<"You win across in last row!"<<endl;
c=c+4 ;

}
if ((R13==R22)&&(R22==R31))
{
cout<<endl;
cout<<endl;
cout<<"You win diagonally!"<<endl;
c=c+4;

}
if ((R11==R22)&&(R22==R33))
{
cout<<endl;
cout<<endl;
cout<<"You win diagonally!"<<endl;
c=c+4 ;

}
if ((R11==R21)&&(R21==R31))
{
cout<<endl;
cout<<endl;
cout<<"You win going down!"<<endl;
c=c+4;

}
if ((R12==R22)&&(R22==R32))
{
cout<<endl;
cout<<endl;
cout<<"You win going down!"<<endl;
c=c+4 ;

}
if ((R13==R23)&&(R23==R33))
```

```
{
cout<<endl;
cout<<endl;
cout<<"You win going down!"<<endl;
c=c+4;

cout<<endl;
}
}
//function*************************************************************
// this function asks the participant if they would like to see the rules.  If
// the participant answers the "rules" integer with a yes answer then the
// program goes to the rules function.  If the participant answers no then the
// program simply closes itself down.
int main()
{
int rules;
cout<<endl;
cout<<endl;
cout<<"Would you like to see the rules?:  ";
cin>>rules;
cout<<endl;
cout<<endl;
if ((rules='y')||(rules='Y'))
{
PrintRules ();
}
else
{
return 0;
}
   system("PAUSE");
      return 0;
      }
```

```
// Roulette wheel
//
//
//
#include <iostream.h>
#include <stdlib.h>
#include <time.h>
#include <fstream.h>
//*********************
// function
// this function is the result of one of the two options that the player chose
// in the 'playgame' function ( number being spin_number and color being
//spin_color.) this function asks the user which of the three
// colors (red, black, green) they would like to bet on. After that the program
// assigns a number between 1-3 to the three colors.  Then the program
generates
// a randomly selected number between 1-3 and if the "color number" the player
// chose is the same "color number" that the program randomly generated then
the
// individual wins. If the players "color number" does not match the generated
// number the player loses.
// in this function the integer "r" is color that the participant is entering
// when the program prompts them to.
// the participant chooses between the three colors (green, black, red) and each
// color is assigned one letter (r=red, b=black, g=green) (The program outputs
// the instructions to choose one letter instead of the whole name of the color)
// from there the participant enters one letter as integer r.
// integer "n" in this function is the number assigned to each color so it can
// be randomly generated by the program.  if the r integer is equal to b (black)
// then the "n" integer is equal to 2. if the r integer is equal to r (red) then
// the "n" integer is equal to 1.  If the r integer is equal to g (green) then
// the "n" integer is equal to 3.
// integer "m" in this function is the money that the participant choses to bet
// with each time. If the players 'r' integer is equal to the randomly generated
// number then they're money bet (m) is doubled.  If the nubmers do not match
// then they're money bet (m) is counted against them.
//*************************************************************************
int SpinColor()
{
char r;
int m,n;
int color_spin;
cout<<"What color(r = red, b= black, g= green)do you want to bet on?"<<endl;
```

```
cin>> r;
if ((r=='r')||(r=='R'))
   n=1;
if ((r=='b')||(r=='B'))
   n=2;
if ((r=='g')||(r=='G'))
   n=3;
color_spin = 1+rand()%3;
cout<<" You chose the color number:  "<<n<<endl;
cout<<" and the roulette wheel landed on the color number "<<color_spin<<endl;
if (color_spin ==n)
m=2;
else
m=-1;
return (m);
}
//*************************************************************************
//function
// This function is also the result of a choice made by the participant at the
// main menu.  This function asks the user to chose a number between 1-33 that
// they would like to bet on.  After this the program generates a random number
// between 1-33 and then compares the one it generated to the one that the
// player chose.  If these two numbers match then the player wins.  If these
// two numbers differ then the player loses.
// in this function integer "m" is the money that the player chooses to bet each
// time they decide to play the game. If the players number matches up with the
// computers random number then their money (m) is doubled. If the players
// number differs from the computers number then their bet is counted against
// them.
// In this function the integer "k" is the number that the participant chooses
// to bet on.  This number has to be no less than 1 and no greater than 33. If
// this number is not in those boundaries then the program will repeat its
// request for the participant to enter a number between 1-33.
// In this function the integer "number_spin" is the number generated randomly
// by the program.  The program has to randomly generate a number between 1-
33
// for the intger "number_spin". Once this number is generated it is compared
// to "k" which is the number the player chose.  If the numbers match the
// participant wins. otherwise the participant loses.
//*************************************************************************
int SpinNumber()
{
int m,k;
int number_spin;
cout<<"What Number (0-33) do you want to bet on?"<<endl;
cin >>k;
```

```
if ((k<0)||(k>33))
{
SpinNumber();
number_spin = 0+rand()%33;
cout <<"You chose # "<<k<<" the roulette wheel landed on
"<<number_spin<<endl;
}
if (k>33)
{
return 0;
}
if (number_spin ==k)
{
m=5;
}
else
{
m=-1;
return (m);
}
}

//*************************************************************************
// function
// This function is where the player decides to chose whether to bet on either
// color or number or both.
// In this function integer "choice" is the selection the participant makes from
// the main menu.  The program outputs the selections the participant can choose
// and assigns each choice a number.  The participant is then asked to enter one
// of these numbers as their preferred selection.
// In this function the integer "choice" is the number that the participant
// chooses from the menu that gives them the selection of things to bet on.
// The integer "spin" in this function is the variable used to distinguish the
// functions that branch off of this function.  You will notice that there are
// 3 spins integers (spin,spin1,spin2) basically if the "choice" integer is
// equal to 1 then the integer "spin" is equal spin. if the integer "choice" is
// 2 then the "spin" integer is equal to spin1. if the integer "choice" is
// is equal to 3 then the integer "spin" is equal to spin2.
// all of these integers (spin,spin1,spin2) are all assigned to different
// functions. The integer "spin" will go to the "spincolor" function.  The
// integer "spin1" will go to the "spinnumber" function. The integer 3 will go
// to the "spincolor" and "spinnumber" functions.
int PlayGame()
// plays one spin of the roulette wheel after asking what they want to bet on
// choices are color only
//             number only
```

```
// or both
{
int choice;
int spin;
int spin1,spin2;
cout<<"Which of the folowing do you want to bet on:"<<endl;
cout<<"          1. Color only "<<endl;
cout<<"          2. Number only "<<endl;
cout<<"          3. Both color and number "<<endl;
cin >>choice;
if (choice ==1)
   spin = SpinColor();
if (choice ==2)
   spin =SpinNumber();
if (choice ==3)
{
   spin1=SpinColor();
   spin2=SpinNumber();
   if ((spin1>0)||(spin2>0))
   spin =1;
   if ((spin1<0)&&(spin2<0))
   spin =-1;
   else
   spin = 5;
}
return(spin);
}
//*********************************
// this functions role is to simply ask the participant how much he/she would
// like to bet on. The participant has to bet between 1 dollar and 200 dollars.
// the integer "b" is the amount of money that the participant wants to bet on
// either color or number.  If the number that the participant entered for "b"
// is greater than 200 or less than 1 then the function will repeat itself until
// the participant enters a valid number.  After this the "b" integer (or bet)
// is returned for the other functions to use.  This is returned mainly for the
// purpose of the text files statistics that we used to perform the chi-squared
// test.
// function
int Bet()
{
int b;
cout<<"How much money do you want to bet ($1 - $200)"<<endl;
cin>> b;
if ((b<1)||(b>200))
Bet();
return (b);
```

```
}
//*********************************
// this function finds the total of all of the winnings that the participant
// "racked up".  the integer "total" is the final amount of coins/money that the
// player ends up with.  the integer "x" is set equal to the integer "b" which
// is returned from the function above so it can be subtracted or added from the
// coins that the player started with.
// the intger "winnings" simply multiplies the two integers "x" and "winnings"
// to come up with the final coin amount. This amount is returned for use of the
// text file in the main function.
//function
int FindTotal()
{
int total;
int x =Bet();
int winnings = PlayGame();
total =winnings*x;
return (total);
}
//*********************************
// this function brings all of the functions into play by asking simple
// questions.  In this function the integer "count" is the number of times the
// participant played the game.  This integer is set equal to zero until the
// person answers the question in this function that asks if they would " like
// to spin the wheel" if the participant replays with a yes answer then the
// "count" integer adds 1. If the participant answers with a no then the \
// program shuts down completely.  The integer "money" is the amount of money
// that the participant starts out with at the beginning of the game.  The
// program makes the statement that they only have 200 coins at the beginning of
// the program. the integer "playerID" is used to record the participants
// id that was given to them before they played the games.  This integer is
// very important because it is read into the text file along with the amount
// that the person bet each time. the character "an" in this function is the
// answer that the participant gives to the statement "are you ready to spin the
// wheel" if the person answers yes the "count" integer is increased by one.
// Then the "money" integer is added with the "find total" integer that was
// returned in a earlier function.  After this the program outputs how many
// times the participant played the game and what they ended up with.  When the
// particpant decides to stop playing and answers the question "are you ready to
// spin the wheel" with a no answer then the numbers gathered from the integers
// "money" "playerID" and "count" are all put into a text file which outputs
// all three final totals.  the text file displays the playerID first. Then the
// amount of times they played "count" and the amount they bet each time
"money"
// after looking at the information gathered from the text file from this
// function then we can test our hypothesis and prove it correct or incorrect.000
```

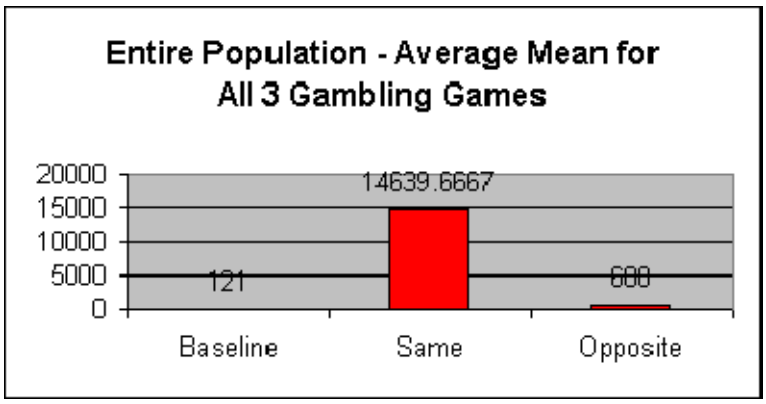
```
//MAIN Program
int main()
{
srand( (unsigned)time(NULL)); //init Rand() function
int count=0;
int money = 200;
char an;
int playerID;
cout<<"Please enter your player ID # from your survey sheet"<<endl;
cin>>playerID;
cout<<endl;
cout<<"Are you ready to Spin the Wheel?"<<endl;
cin >> an;
while ((money>0)&&((an =='y')||(an =='Y')))
{
count++;
money = money+FindTotal();
cout<<"You have spinned the wheel "<<count<<"   time/s"<<endl;
cout<<endl;
cout<<"your total is = $ "<<money<<endl;
cout<<endl;
cout<<"Are you ready to Spin the Wheel?"<<endl;
cin>> an;
cout<<endl;
{
return (playerID);
}
}
//**** add ID #, total times played and total of winnings at end of play
fstream RouletteFile("Roulette.txt", ios::in | ios::out);
if(RouletteFile.fail())
{
 cout<<"File could not be opened"<<endl;
 system ("Pause");
 return 0;
}
else
{
 RouletteFile.seekp(0,ios::end);                    // go to end of datafile
 RouletteFile<<playerID;                            //insert playerid
 RouletteFile.width(10); RouletteFile<<count;           //insert count
 RouletteFile.width(20);RouletteFile<<money<<endl;    //insert money
      system("PAUSE");
      return 0;
}
}
```

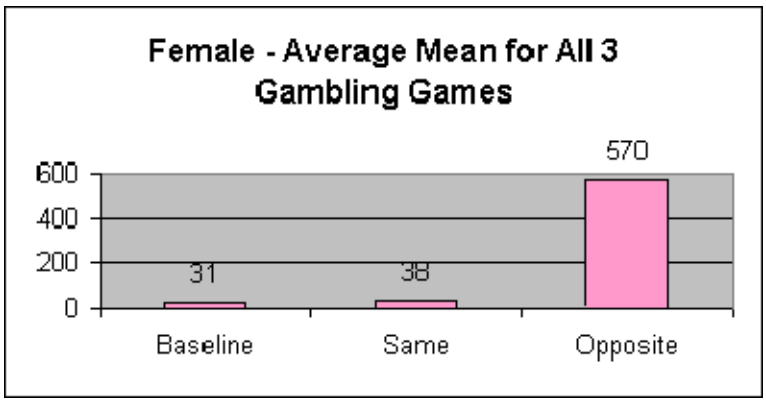
Appendix C

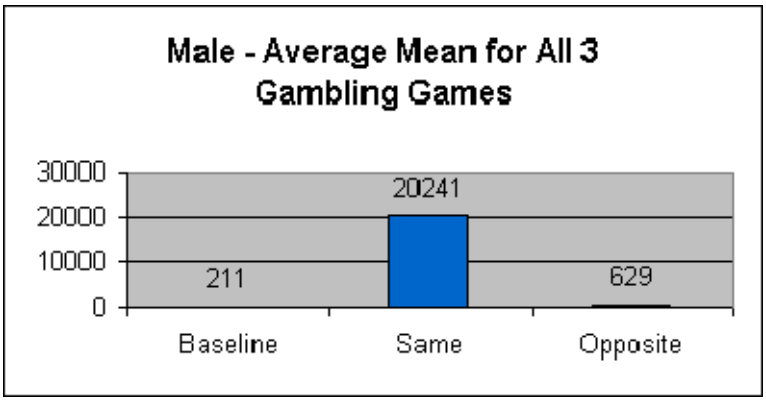
Mean Graphs

Total Means	Baseline	Same	Opposite
	121	14639.67	600

Total Gender Means	Baseline	Same	Opposite
f	30.66667	38.33333	570.3333
m	211.3333	20241	628.6667





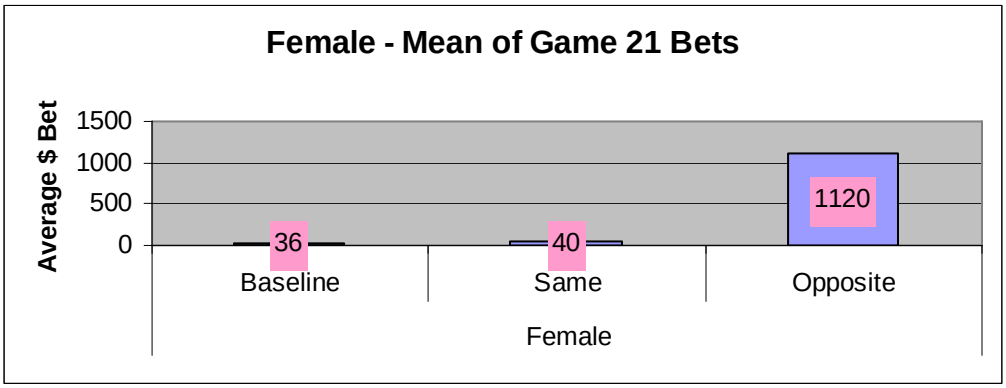


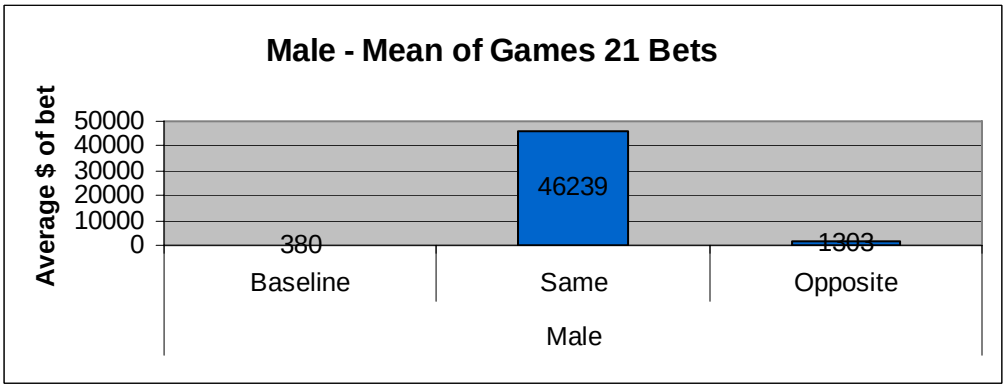
Appendix D
Chi-Squared Test

Game 21 Average Bets

Participants#	Gender	Baseline	Same	Opposite
103	f	19	26	0
104	f	10	26	23
106	f	27	20	23
107	f	23	20	5793
108	f	21	27	911
109	f	8	361	314
110	f	24	30	200
111	f	13	30	1650
112	f	24	23	33
113	f	51	23	20
114	f	47	232	1000
115	f	42	23	1500
116	f	21	21	26
117	f	13	21	34
118	f	20	21	225
119	f	30	21	18
120	f	40	19	760
121	f	42	19	38
122	f	16	19	13850
123	f	42	19	21
124	f	27	23	22
125	f	50	20	26
126	f	273	20	1691
127	f	42	21	22
128	f	18	21	26
129	f	20	22	41
130	f	29	22	1780
131	f	43	23	1000
132	f	16	19	24
133	f	25	19	1429
101	m	462	663946	5793
102	m	4225	663946	820
105	m	20	23	23
106	m	60	23	5793
107	m	417	2155	1650
108	m	377	2155	200
109	m	192	775	33
110	m	261	775	20
111	m	218	1500	1000
112	m	159	1500	1500
113	m	193	200	26
114	m	233	200	34
115	m	417	1780	13850
116	m	417	1780	911

117m		252	775	760
118m		400	775	225
119m		308	1518	21
120m		231	1518	18
121m		113	257	22
122m		224	257	26
123m		529	6280	1691
124m		220	820	22
125m		206	6280	1780
126m		175	236	26
127m		200	236	1000
128m		29	25	41
129m		56	25	24
130m		144	13291	314
131m		300	13291	38
132m		364	820	1429
	Total	12478	1388372	71567
	Mean	208	23140	1213





Frequency of social interaction of betting increasing average bet made by teenagers

	Yes (same)	No (same)	Yes (opposite)	No (opposite)	
female	13	17	22	8	60
male	26	4	15	15	60
	39	21	37	23	120

Step 1 Expected

	Yes (same)	No (same)	Yes (opposite)	No (opposite)
Female	19.5	10.5	18.5	11.5
Male	19.5	10.5	18.5	11.5

Step 2 Observed

	Yes (same)	No (same)	Yes (opposite)	No (opposite)
Female	2.1666667	4.02381	0.662162162	1.065217391
Male	2.1666667	4.02381	0.662162162	1.065217391

Chi-Square Stat

	15.835711	df=(r-1)*(c-1)
P=.99	11.3	df=3

Mean Female

Baseline	Same	Opposite
36	40	1120

Male

Baseline	Same	Opposite
380	46239	1303

Standard Deviation

Female

Baseline	Same	Opposite
46.473938	71.7235	2710.15902

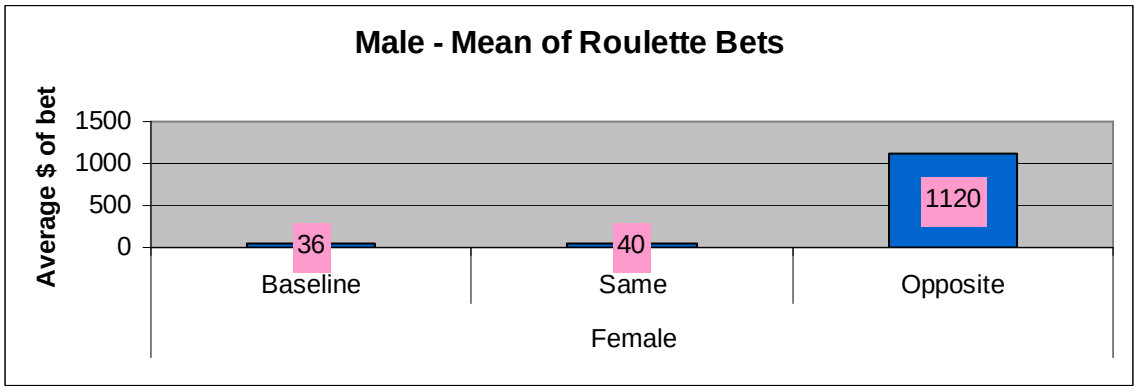
Male

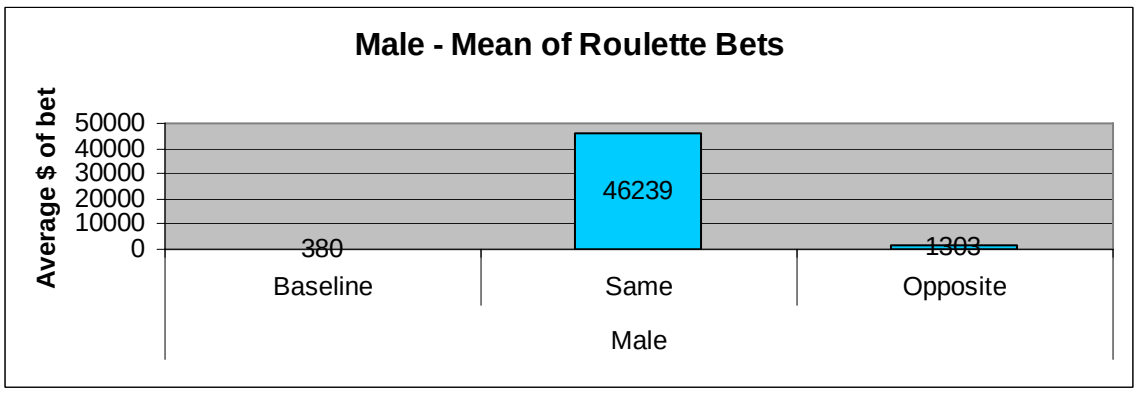
Baseline	Same	Opposite
737.90903	167945.9	2787.452459

Roulette Average Bets

Participants#	Gender	Baseline	Same	Opposite
103	f	51	21	157
104	f	20	21	317
106	f	20	34	140
107	f	30	20	157
109	f	20	14	85
110	f	11	14	260
113	f	38	30	20
114	f	20	30	168
115	f	31	23	38
116	f	25	23	20
118	f	30	22	100
122	f	37	22	200
123	f	26	21	26
124	f	13	21	36
125	f	20	22	22
126	f	22	22	20
128	f	40	18	76
130	f	34	18	50
131	f	17	16	185
133	f	43	16	203
136	f	31	17	22
144	f	45	20	26
145	f	28	20	114
148	f	40	20	22
150	f	20	20	26
151	f	16	18	40
152	f	28	18	113
156	f	40	17	100
157	f	16	17	28
159	f	25	17	660
101	m	103	128	157
102	m	68	128	317
105	m	30	38	140
108	m	53	38	157
111	m	118	37	168
112	m	50	37	20
117	m	28	775	38
119	m	25	775	20
120	m	23	1500	100
121	m	15	1500	200
127	m	35	200	26
129	m	25	200	36
132	m	102	203	185
134	m	41	147	85
135	m	24	100	76
137	m	22	100	22

138m	38	124	147
139m	22	144	20
140m	13	225	22
141m	22	225	26
142m	52	76	177
143m	22	100	22
146m	20	76	113
147m	17	21	26
149m	20	21	100
153m	10	40	40
154m	51	40	28
155m	14	194	260
158m	22	200	50
160m	39	100	660
Total	1961	8103	6869
Mean	33	135	114





Frequency of social interaction of betting increasing average bet made by teenagers

	Yes (same)	No (same)	Yes (opposite)	No (opposite)		
female	7	23	23	7	60	
male	26	4	23	7	60	

	33	27	46	14	120

Step 1 Expected

	Yes (same)	No (same)	Yes (opposite)	No (opposite)
Female	16.5	13.5	13.5	13.5
Male	16.5	13.5	23	7

Step 2 Observed

	Yes (same)	No (same)	Yes (opposite)	No (opposite)
Female	5.469697	6.685185	6.685185	3.12962963
Male	5.469697	6.685185	0	0

Chi-Square Stat

	34.12458	df=(r-1)*(c-1)
P=.99	11.3	df=3

Mean

Female

Baseline	Same	Opposite
28	20	114

Male

Baseline	Same	Opposite
37	250	115

Standard Deviation

Female

Baseline	Same	Opposite
10.32959	4.468994	130.6219

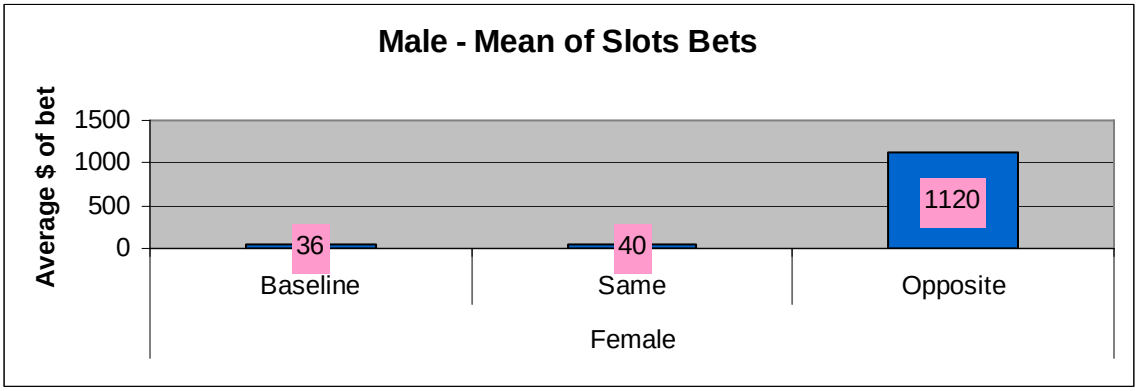
Male

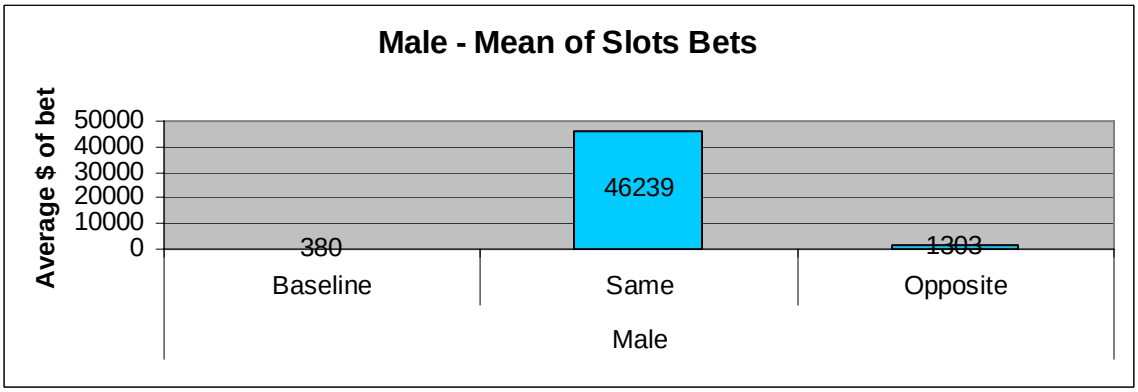
Baseline	Same	Opposite
27.65129	384.2743	130.2425

Slots Average Bets

Participants#	Gender	Baseline	Same	Opposite
103	f	46	29	60
104	f	6	29	174
106	f	8	12	13
107	f	9	12	60
109	f	5	17	833
110	f	8	17	300
113	f	28	30	260
114	f	12	30	2171
115	f	21	35	43
116	f	56	35	43
118	f	51	29	1000
122	f	33	28	1500
123	f	25	20	45
124	f	25	20	52
125	f	20	20	75
126	f	30	20	38
128	f	51	11	780
130	f	40	11	46
131	f	15	30	1516
133	f	30	30	48
136	f	30	38	60
144	f	55	23	58
145	f	47	23	871
148	f	44	221	46
150	f	18	221	48
151	f	20	252	52
152	f	28	252	1257
156	f	43	38	2400
157	f	16	58	30
159	f	23	48	433
101	m	20	608607	60
102	m	205	608607	174
105	m	4	10	13
108	m	33	12	60
111	m	622	1133	2171
112	m	390	1780	260
117	m	292	325	43
119	m	30	325	43
120	m	21	200	1000
121	m	16	200	1500
127	m	188	175	45
129	m	213	175	52
132	m	290	220	1516
134	m	364	220	833
135	m	261	550	78
137	m	445	550	375
138	m	320	1250	48
139	m	305	1000	38

140m		172	1614	60
141m		230	1383	58
142m		600	1780	1012
143m		240	1100	46
146m		48	1133	1257
147m		175	225	48
149m		200	225	2400
153m		110	25	52
154m		56	25	30
155m		137	1614	300
158m		300	1383	46
160m		211	1166	433
	Total	7339	1238650	28363
	Mean	122	20644	473





Frequency of social interaction of betting increasing average bet made by teenagers

	Yes (same)	No (same)	Yes (opposite)	No (opposite)		
female		16	14	29	1	60
male		23	7	14	16	60
		39	21	43	17	120

Step 1 Expected

	Yes (same)	No (same)	Yes (opposite)	No (opposite)
Female	19.5	10.5	21.5	8.5
Male	19.5	10.5	21.5	8.5

Step 2 Observed

	Yes (same)	No (same)	Yes (opposite)	No (opposite)
Female	0.628205	1.166667	2.61627907	6.617647059
Male	0.628205	1.166667	2.61627907	6.617647059

Chi-Square Stat

	22.0576	df=(r-1)*(c-1)
P=.99	11.3	df=3

Mean Female

Baseline	Same	Opposite
28	55	477

Male

Baseline	Same	Opposite
217	41234	468

Standard Deviation

Female

Baseline	Same	Opposite
15.58126	73.54514	676.0082402

Male

Baseline	Same	Opposite
162.1564	154230.2	677.1534857