Monte Carlo On Ice: How Simulation Can Help Figure Skating Medalist

James Lee (jlee73@ncsu.edu)

2018/04/25

1 Introduction

Figure skating has a long history, it was the first winter sport to be included in the Olympics in 1908. The scoring of figure skating is based on the performance of jumps, spins, step sequence and aesthetic component in one's program. To make every elements accountable, modern figure skating has a very complicated scoring system. However, the system lies with two major flaws: First, it prefers skaters challenging high base value jump over perfecting slightly easier ones. Second, the extra base value can be earned by performing jumps in second half, leads to a loophole for skaters to exploit. ISU rule is looking to change recently to stride a better balance between on technical and aesthetic side of the sport.

The goal of this project is to use Monte-Carlo method to simulate the difference in scoring distribution before and after the proposing rule change. Therefore giving advice to all level of skaters, judges, and ISU. The report will cover the background knowledge, method and parameter used, results, discussions and comments on the proposing plan.

2 ISU Judging System and the recent proposal

In single figure skating events, a skater must complete two programs: Short Program and Free Skating. Short Program has maximum time of 2 minutes 50 seconds, requires 7 elements including an axel jump, triple or quadruple jump with connecting steps, combination jumps, and other spins/sequences. Free Skating has time limit of 4 minutes (senior ladies) or 4 minutes and 30 seconds (senior men/ pairs) with fewer limitations.

The scoring of both programs will be decomposed to two components: Technical Elements Score (TES) and Program Component Score (PCS). In TES, each elements of the program is assigned to a base value (BV), and 9 judges will evaluate the Grade of Execution (GOE) based on the quality of the elements and is finalized by averaging the middle 7 scores. TES is the sum of all elements' BV and GOE. Program component scores is the sum of five parts: skating skills, transitions, performance, composition, and music interpretation. Each with a mark from 0.25 to 10.00 with 0.25 increments.

Recent proposing rule change includes lowering the base values of quadruple jumps (See Figure) and extending GOE from -3 +3 to -5 +5, which could go into effect for the 2018-19 season.

3 Method and Parameter

For the purpose of demonstration and simplicity, I used all historical free skating results of Yuzuru Hanyu, two-time Olympic Gold Medalist, as example. Also, the program use to define the elements simulated are based on Yuzuru Hanyu's real program from 2018 winter Olympics and his early career. The elements for both program are shown as following chart.

Judges Details per Skater

선수별 심판 세부채점 정보 / Notation détaillée des juges par patineur

Rank	Name	NOC Code	Starting Number	Total Segment Score	Total Element Score	Total Program Component Score (factored)	Total Deductions
2	HANYU Yuzuru	JPN	22	206.17	109.55	96.62	0.00

#	Executed Elements	Info	Base Value		GOE	J1	J2	J3	J4	J5	J6	J7	J8	J9	Ref.	Scores of Panel
1	4S		10.50		3.00	3	2	3	3	3	3	3	3	3		13.50
2	4T		10.30		3.00	3	2	3	3	3	3	3	3	3		13.30
3	3F		5.30		1.60	3	2	3	3	2	2	2	2	2		6.90
4	FCCoSp4		3.50		1.00	2	2	2	2	2	2	2	2	1		4.50
5	StSq3		3.30		1.43	3	2	3	3	3	2	3	3	3		4.73
6	4S+3T		16.28	x	2.71	3	1	3	3	2	2	3	3	3		18.99
7	4T+REP		7.93	x	-2.06	-1	-2	-2	-2	-2	-1	-2	-2	-1		5.87
8	3A+1Lo+3S		14.74	x	2.14	3	1	2	3	2	2	3	2	1		16.88
9	3Lo		5.61	x	1.20	3	1	2	2	1	2	2	2	1		6.81
10	3Lz		6.60	x	-1.10	-1	-2	-2	-1	-2	-2	-1	-2	-1		5.50
11	FCSSp4		3.00		0.93	2	2	2	2	1	1	2	3	2		3.93
12	ChSq1		2.00		2.00	3	2	3	3	3	2	3	3	3		4.00
13	CCoSp4		3.50		1.14	3	2	2	3	2	2	3	2	2		4.64
			92.56													**109.55**

Program Components	Factor	J1	J2	J3	J4	J5	J6	J7	J8	J9	Scores of Panel
Skating Skills	2.00	9.75	9.50	9.75	9.75	9.50	9.75	9.75	9.75	9.75	9.71
Transitions	2.00	9.50	9.25	9.75	9.75	9.25	9.50	9.50	9.50	9.50	9.50
Performance	2.00	9.75	9.50	9.75	9.75	9.00	9.50	9.75	9.75	9.50	9.64
Composition	2.00	9.75	9.50	9.75	10.00	9.50	9.75	9.50	10.00	9.75	9.71
Interpretation of the Music	2.00	10.00	9.50	9.75	9.75	9.75	9.75	9.75	9.75	9.75	9.75
Judges Total Program Component Score (factored)											**96.62**
Deductions:											**0.00**

Figure 1: An example scoring sheet from Yuzuru Hanyu in 2018 Winter Olympics

Jump/Pairs Throw	2017-18 Base Value	Proposed Base Value*
Quad Toe	10.3	9.5
Quad Salchow	10.5	9.7
Quad Loop	12.0	10.5
Quad Flip	12.3	11.0
Quad Lutz	13.6	11.5
Quad Axel	15.0	12.5
Triple Toe	4.3	4.2
Triple Salchow	4.4	4.3
Triple Loop	5.1	4.9
Triple Flip	5.3	5.3
Triple Lutz	6.0	5.9
Triple Axel	8.5	8.0
Throw Quad Toe	8.2	6.5
Throw Quad Salchow	8.2	6.5
Throw Quad Loop	8.7	7.0
Throw Quad Flip/Lutz	9.0	7.5

Figure 2: Proposed Base Value Change

```
(easier,safer program)
-4T
2S
3F
FCCoSp4
StSq3
-3A+3T
-3A+2T
-3Lo
-3Lz+2T+2T
-3Lz
ChSq1
CCoSp4
FCSSp4

78.51BV+13.45GOE

(more intense program)
-4S
-4T
3F
FCCoSp4
StSq3
-4S+3T
-4T+REP
-3A+1Lo+3S
-3Lo
-3Lz
FCSSp4
ChSq1
CCoSp4

92.56BV+16.99GOE
```

Figure 3: Programs used in simulation

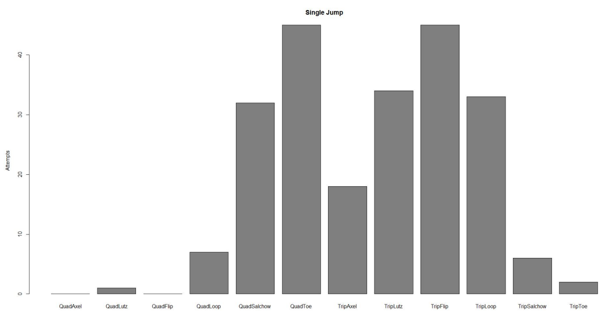

Figure 4: Single jumps attempts in data

First, we can generate empirical distribution for every elements on both BV and GOE. After the base value is decided, we simulate falls by giving a probability based on historical sample mean of falls on elements. If a fall is simulated, we immediately assigned GOE as -3 or -5 by rule and give 1 point deduction on whole program, if not, then generate GOE from the empirical distribution. Lastly,after all elements are generated, the five components are generated by 9 identical discrete uniform distributions (8.25 9.5 for "hard" program and 8.75 10.00 for "easy" program) and set to be the average of the middle 7 scores.

The flow chart of this process can be seen in below and the R code in appendix.

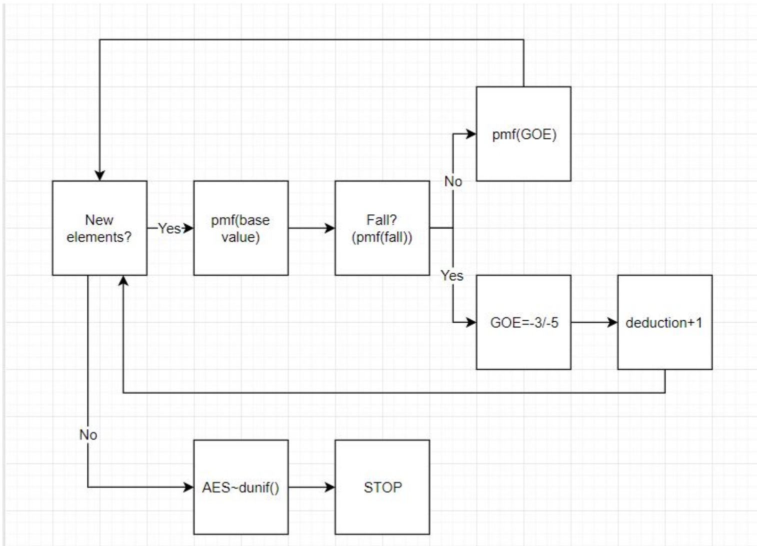

Figure 5: Flow chart of the simulation

4 Results

As the comparisons bar chart below shows, the hard program still has a significant point lead in both before and after the score change. The difference, however, is the percentage winning by the harder program has drop nearly 7 percent due to the higher variance caused by extended GOE. After proposed rule change the maximum score also increase in both program, due to most first-tier skaters like Hanyu have positive total GOE, getting extra points after GOE is extended.

Although the simulation is relatively conservative on the AES scoring distribution, we can still say the program score of Yuzuru Hanyu in Winter Olympics is higher than most simulations (Hanyu score 206.17 while the maximum of the simluation is 207.35)

5 Discussion and Comments

This simulation can be used on all level of skaters if with enough historical data. Top tier skaters can understand the risk and potential of doing a high base value jump in the future; Middle tier skaters can maximize their chance of standing on podium; Also the estimation can be made for the skater is moved from junior league to senior league.

The new rule still gives higher BV, harder jumps like quad or 3A an advantage, but the skaters had to bear more risk to it. New jumps like 4A may be highly discouraged because of the unbearable high risk. However, it also gives incentive to top skaters to achieve personal best and even world record since the average score will be higher than before.

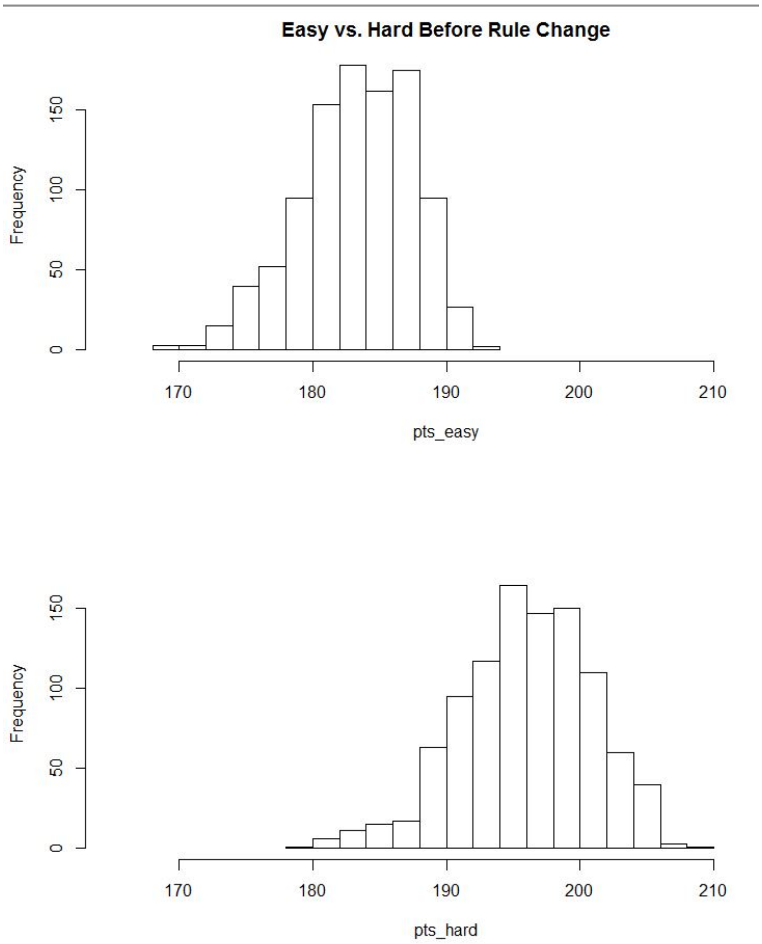

Figure 6: n=1000 simulation before rule change, Mean on Easy program=183.35 Hard program=195.8

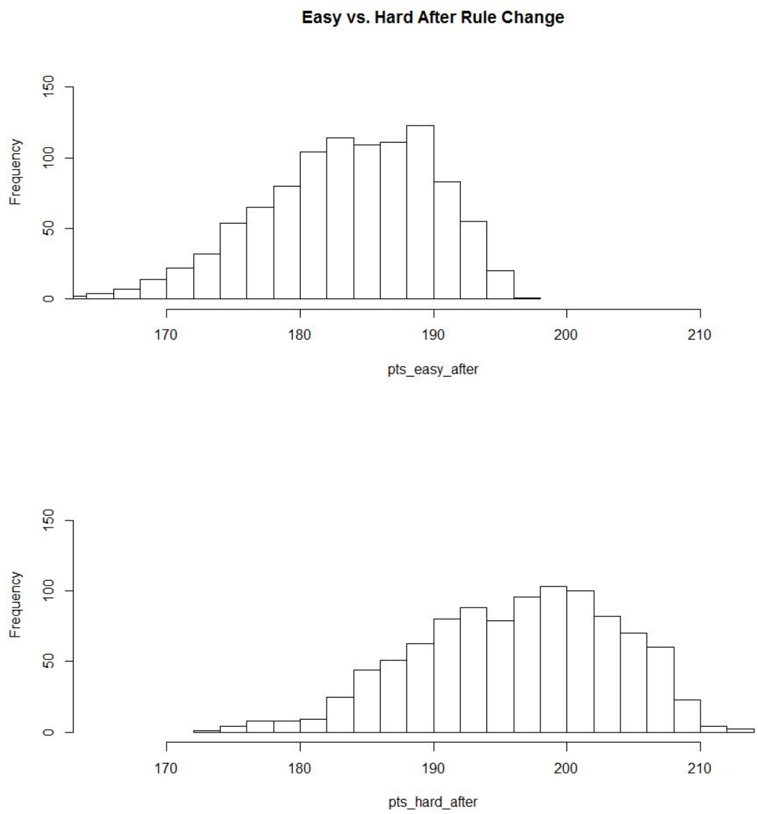

Figure 7: n=1000 simulation before rule change, Mean on Easy program=183.76 Hard program=196.1

Overall, I think the rule change is positive since the main issue on balancing aesthetic and technical is solved: technical, high base value jumps still have incentive on breaking new record, but riskier than before if skaters wants to stand on the podium. Aesthetic-based program has a chance to catch up if performed perfectly. However, there are some issue may not be solved by changing rules objectively. Back-loading program will still be an issue since cancel the compensation would only lead to front-loading, as why this rule is firstly applied. And adjusting it can't help either, since different skaters will have different concern and risk and stamina. Lastly deduction on AES also won't help since it would be a matter of being subjective and putting more restrictions on free skating.

6 Appendix: R Code

```
###########################################################################################
#Figure Skating Analysis Using Monte Carlo Simulation
#Author: James Lee(jlee73@ncsu.edu)
#Abstract: The goal of this project is to understand FS program scoring and
#          simulate the distribution of score before and after IJS rule change.
###########################################################################################

####load csv file
#change directory if needed
Player <- read.csv(file="D:/ISU dataset/Monte-Carlo-On-Ice/Yuzuru Hanyu.csv", header=FALSE, sep=",")

####The data is scrapped from ISU website, which needs data-cleaning

#Type,exclude some irrelavent names
```

```
Type<-unique(Player[,1])
rm_type<-c("Elements","Program components","Total","Skating Skills","Transitions","Performance/Execu
           "Composition","Interpretation","Deduction","Program total","Transition/Linking Footwork",
           "Choreography/Composition","Transition/Linking Footwork/Movement","Choreography")
rm_index<-which(Type %in% rm_type)

##compute how many times each type of jumps have perform and plot a barplot
#table type
 Type<-Type[-rm_index]
 par(mar = c(7, 4, 2, 2) + 0.2)
 Type_count<-table(Player[,1],exclude=rm_type)

##Drawing barplot
#decide end point, the frame of the plot
end_point = 0.5 + nrow(Type_count) + nrow(Type_count)-1
#barplot-delete axisnames and put on rotated ones using text() syntax
barplot(Type_count,main="Number of Attempts",col="grey50", ylim=c(0,5+max(Type_count))
         ,ylab="Attempts",xlab="",space=1,axisnames = FALSE)
text(seq(1.5,end_point,by=2), par("usr")[3]-0.25,
      srt = 60, adj= 1, xpd = TRUE,
      labels = paste(rownames(Type_count)), cex=0.65)


#define all jump

S1A_M<-Player[Player$V1 %in% c("1A"),1:3]
C1A1Lo3S_M<-Player[Player$V1 %in% c("1A+1Lo+3S"),1:3]
S1A1T_M<-Player[Player$V1 %in% c("1A+1T"),1:3]
S1Lo_M<-Player[Player$V1 %in% c("1Lo"),1:3]
S1Lz_M<-Player[Player$V1 %in% c("1Lz"),1:3]
C1Lz2T_M<-Player[Player$V1 %in% c("1Lz+2T"),1:3]
S1S_M<-Player[Player$V1 %in% c("1S"),1:3]
S2Lz_M<-Player[Player$V1 %in% c("2Lz"),1:3]
S2S_M<-Player[Player$V1 %in% c("2S"),1:3]
C2S1Lo_M<-Player[Player$V1 %in% c("2S+1Lo"),1:3]
C2S3T_M<-Player[Player$V1 %in% c("2S+3T"),1:3]
S2T_M<-Player[Player$V1 %in% c("2T"),1:3]
C2T1Lo2S_M<-Player[Player$V1 %in% c("2T+1Lo+2S"),1:3]
S3A_M<-Player[Player$V1 %in% c("3A","3A<"),1:3]
C3A1Lo2S_M<-Player[Player$V1 %in% c("3A+1Lo+2S","3A+1Lo<<+2S"),1:3]
C3A1Lo3S_M<-Player[Player$V1 %in% c("3A+1Lo+3S","3A+1Lo<+3S"),1:3]
C3A1T_M<-Player[Player$V1 %in% c("3A+1T"),1:3]
C3A1T1Lo_M<-Player[Player$V1 %in% c("3A+1T+1Lo"),1:3]
C3A2T_M<-Player[Player$V1 %in% c("3A+2T"),1:3]
C3A2T2T_M<-Player[Player$V1 %in% c("3A+2T+2T"),1:3]
C3A3T_M<-Player[Player$V1 %in% c("3A+3T","3A+3T<"),1:3]
C3A3T2T_M<-Player[Player$V1 %in% c("3A+3T+2T"),1:3]
S3F_M<-Player[Player$V1 %in% c("3F","3Fe"),1:3]
S3Lo_M<-Player[Player$V1 %in% c("3Lo","3Lo<"),1:3]
C3Lo1Lo3S_M<-Player[Player$V1 %in% c("3Lo+1Lo+3S"),1:3]
C3Lo1T_M<-Player[Player$V1 %in% c("3Lo+1T"),1:3]
C3Lo2T_M<-Player[Player$V1 %in% c("3Lo+2T"),1:3]
S3Lz_M<-Player[Player$V1 %in% c("3Lz","3Lz<"),1:3]
C3Lz1Lo3S_M<-Player[Player$V1 %in% c("3Lz+1Lo+3S","3Lz+1Lo+3S<"),1:3]
C3Lz1T_M<-Player[Player$V1 %in% c("3Lz+1T"),1:3]
C3Lz2T_M<-Player[Player$V1 %in% c("3Lz+2T"),1:3]
```

```
C3Lz2T2T_M<-Player[Player$V1 %in% c("3Lz+2T+2T"),1:3]
S3S_M<-Player[Player$V1 %in% c("3S"),1:3]
C3S2T_M<-Player[Player$V1 %in% c("3S+2T"),1:3]
S3T_M<-Player[Player$V1 %in% c("3T"),1:3]
S4Lo_M<-Player[Player$V1 %in% c("4Lo","4Lo<","4Lo<<","4Lo+REP"),1:3]
S4Lz_M<-Player[Player$V1 %in% c("4Lz","4Lz<","4Lz<<","4Lze","4Lz+REP"),1:3]
S4S_M<-Player[Player$V1 %in% c("4S","4S<","4S<<","4S+REP"),1:3]
C4S3T_M<-Player[Player$V1 %in% c("4S+3T"),1:3]
S4T_M<-Player[Player$V1 %in% c("4T","4T<","4T<<","4T+REP","4T<+REP"),1:3]
C4T1Lo3S_M<-Player[Player$V1 %in% c("4T+1Lo+3S"),1:3]
C4T2T_M<-Player[Player$V1 %in% c("4T+2T"),1:3]
C4T3T_M<-Player[Player$V1 %in% c("4T+3T"),1:3]
CCoSp1_M<-Player[Player$V1 %in% c("CCoSp1"),1:3]
CCoSp2_M<-Player[Player$V1 %in% c("CCoSp2"),1:3]
CCoSp3_M<-Player[Player$V1 %in% c("CCoSp3"),1:3]
CCoSp3p3_M<-Player[Player$V1 %in% c("CCoSp3p3"),1:3]
CCoSp3p4_M<-Player[Player$V1 %in% c("CCoSp3p4"),1:3]
CCoSp4_M<-Player[Player$V1 %in% c("CCoSp4"),1:3]
ChSq_M<-Player[Player$V1 %in% c("ChSq"),1:3]
ChSq1_M<-Player[Player$V1 %in% c("ChSq1"),1:3]
ChSt1_M<-Player[Player$V1 %in% c("ChSt1"),1:3]
CiSt1_M<-Player[Player$V1 %in% c("CiSt1"),1:3]
CiSt3_M<-Player[Player$V1 %in% c("CiSt3"),1:3]
CiSt4_M<-Player[Player$V1 %in% c("CiSt4"),1:3]
CSSp4_M<-Player[Player$V1 %in% c("CSSp4"),1:3]
FCCoSp2_M<-Player[Player$V1 %in% c("FCCoSp2"),1:3]
FCCoSp2p2_M<-Player[Player$V1 %in% c("FCCoSp2p2"),1:3]
FCCoSp3_M<-Player[Player$V1 %in% c("FCCoSp3"),1:3]
FCCoSp3p3_M<-Player[Player$V1 %in% c("FCCoSp3p3"),1:3]
FCCoSp3p4_M<-Player[Player$V1 %in% c("FCCoSp3p4"),1:3]
FCCoSp4_M<-Player[Player$V1 %in% c("FCCoSp4"),1:3]
FCSSp3_M<-Player[Player$V1 %in% c("FCSSp3"),1:3]
FCSSp4_M<-Player[Player$V1 %in% c("FCSSp4"),1:3]
SlSt2_M<-Player[Player$V1 %in% c("SlSt2"),1:3]
SlSt3_M<-Player[Player$V1 %in% c("SlSt3"),1:3]
StSq2_M<-Player[Player$V1 %in% c("StSq2"),1:3]
StSq3_M<-Player[Player$V1 %in% c("StSq3"),1:3]
StSq4_M<-Player[Player$V1 %in% c("StSq4"),1:3]

##creating table for specific barplot on single jumps
SingleJump <- matrix(c(0,1,0,7,32,45,
                       18,34,45,33,6,2),ncol=1,byrow=TRUE)
colnames(SingleJump) <- c("Attempts")
rownames(SingleJump) <- c("QuadAxel","QuadLutz","QuadFlip","QuadLoop","QuadSalchow","QuadToe"
                          ,"TripAxel","TripLutz","TripFlip","TripLoop","TripSalchow","TripToe")
SingleJump <- as.table(SingleJump)
#create barplot
barplot(t(SingleJump),main="Single Jump",col="grey50",
        ylab="Attempts",xlab="")

# ##creating table for specific barplot on combination jumps
#
# CombinationJump <- table(Player[,1],exclude=rm_type)
#
# #create barplot
# barplot(t(CombinationJump),main="Single Jump",col="grey50",
```

```
#         ylab="Attempts",xlab="")
#
# ##creating table for specific barplot on Spins
#
# Spins <- table()
#
# #create barplot
# barplot(Spins,main="Spins",col="grey50",
#         ylab="Attempts",xlab="")

###data preprocessing for the simulation

jump_new<-Player[,1:3]
jump_new<-jump_new[!(jump_new$V1=="Elements"|jump_new$V1=="Program components"|jump_new$V1=="Total"|
                       jump_new$V1=="Skating Skills"|jump_new$V1=="Transitions"|jump_new$V1=="Perfor
                       jump_new$V1=="Composition"|jump_new$V1=="Interpretation"|jump_new$V1=="Deduct
                       jump_new$V1=="Program total"|jump_new$V1=="Transition/Linking Footwork"|
                       jump_new$V1=="Choreography/Composition"|jump_new$V1=="Transition/Linking Foot
                       jump_new$V1=="Choreography"),]
jump_cleaned<-jump_new[order(jump_new$V1),]

#See how GOE is diversed in every jump
GOE<-as.numeric(as.character(jump_cleaned$V3))

barplot(as.vector(GOE),main="GOE",col="grey50",
                ylab="",xlab="")

################################Here is the end of the data proprocessing##########################

### Simulation

n<-1000
#the average difference in
AESavg_easy<-8.75
AESavg_hard<-8.25
pts_easy<-rep(0,n)
pts_hard<-rep(0,n)

#AES scoring:simulate 9 judge, average the middle two for five components
AES_easy<-function(){
  total_AES<-0
  for (i in 1:5){
    total_AES<-total_AES+AESavg_easy+0.25*mean(sort(sample(1:5,9,replace=T))[2:8])
  }
  return(2*total_AES)
}

AES_hard<-function(){
  total_AES<-0
  for (i in 1:5){
    total_AES<-total_AES+AESavg_hard+0.25*mean(sort(sample(1:5,9,replace=T))[2:8])
  }
  return(2*total_AES)
}
for (i in 1:n){
  pts_easy[i]<-sample(as.numeric(as.character(S4T_M$V2)),1)+sample(as.numeric(as.character(S4T_M$V3)
```

```
    sample(as.numeric(as.character(S2S_M$V2)),1)+sample(as.numeric(as.character(S2S_M$V3)),1)+
    sample(as.numeric(as.character(S3F_M$V2)),1)+sample(as.numeric(as.character(S3F_M$V3)),1)+
    sample(as.numeric(as.character(FCCoSp4_M$V2)),1)+sample(as.numeric(as.character(FCCoSp4_M$V3)),1
    sample(as.numeric(as.character(StSq3_M$V2)),1)+sample(as.numeric(as.character(StSq3_M$V3)),1)+
    sample(as.numeric(as.character(C3A3T_M$V2)),1)+sample(as.numeric(as.character(C3A3T_M$V3)),1)+
    sample(as.numeric(as.character(C3A2T_M$V2)),1)+sample(as.numeric(as.character(C3A2T_M$V3)),1)+
    sample(as.numeric(as.character(S3Lo_M$V2)),1)+sample(as.numeric(as.character(S4T_M$V3)),1)+
    sample(as.numeric(as.character(C3Lz2T2T_M$V2)),1)+sample(as.numeric(as.character(C3Lz2T2T_M$V3))
    sample(as.numeric(as.character(S3Lz_M$V2)),1)+sample(as.numeric(as.character(S3Lz_M$V3)),1)+
    sample(as.numeric(as.character(ChSq1_M$V2)),1)+sample(as.numeric(as.character(ChSq1_M$V3)),1)+
    sample(as.numeric(as.character(CCoSp4_M$V2)),1)+sample(as.numeric(as.character(CCoSp4_M$V3)),1)+
    sample(as.numeric(as.character(FCSSp4_M$V2)),1)+sample(as.numeric(as.character(FCSSp4_M$V3)),1)+

  pts_hard[i]<-sample(as.numeric(as.character(S4T_M$V2)),1)+sample(as.numeric(as.character(S4T_M$V3)
    sample(as.numeric(as.character(S4T_M$V2)),1)+sample(as.numeric(as.character(S4T_M$V3)),1)+
    sample(as.numeric(as.character(S3F_M$V2)),1)+sample(as.numeric(as.character(S3F_M$V3)),1)+
    sample(as.numeric(as.character(FCCoSp4_M$V2)),1)+sample(as.numeric(as.character(FCCoSp4_M$V3)),1
    sample(as.numeric(as.character(StSq3_M$V2)),1)+sample(as.numeric(as.character(StSq3_M$V3)),1)+
    sample(as.numeric(as.character(C4S3T_M$V2)),1)+sample(as.numeric(as.character(C4S3T_M$V3)),1)+
    sample(as.numeric(as.character(S4T_M$V2)),1)+sample(as.numeric(as.character(S4T_M$V3)),1)+
    sample(as.numeric(as.character(S3Lz_M$V2)),1)+sample(as.numeric(as.character(S3Lz_M$V3)),1)+
    sample(as.numeric(as.character(C3A1Lo3S_M$V2)),1)+sample(as.numeric(as.character(C3A1Lo3S_M$V3))
    sample(as.numeric(as.character(S3Lo_M$V2)),1)+sample(as.numeric(as.character(S3Lo_M$V3)),1)+
    sample(as.numeric(as.character(ChSq1_M$V2)),1)+sample(as.numeric(as.character(ChSq1_M$V3)),1)+
    sample(as.numeric(as.character(CCoSp4_M$V2)),1)+sample(as.numeric(as.character(CCoSp4_M$V3)),1)+
    sample(as.numeric(as.character(FCSSp4_M$V2)),1)+sample(as.numeric(as.character(FCSSp4_M$V3)),1)+
}
par(mfrow=c(2,1))
hist(pts_easy,xlim=range(165,215),ylim=range(0,180),main="Easy vs. Hard Before Rule Change")
hist(pts_hard,xlim=range(165,215),ylim=range(0,180),main="")
sum(as.numeric(pts_easy<pts_hard))

##change BV/ GOE based on ISU rule change
#deduct pts simply by subtractions
#which may be different if the jump is downgraded or in the 2nd half
S4T_M["V4"]<-as.numeric(as.character(S4T_M$V2))-0.8
S4T_M["V5"]<-as.numeric(as.character(S4T_M$V3))*5/3

C3A3T_M["V4"]<-as.numeric(as.character(C3A3T_M$V2))-0.6
C3A3T_M["V5"]<-as.numeric(as.character(C3A3T_M$V3))*5/3

C3A2T_M["V4"]<-as.numeric(as.character(C3A2T_M$V2))-0.5
C3A2T_M["V5"]<-as.numeric(as.character(C3A2T_M$V3))*5/3

S3Lo_M["V4"]<-as.numeric(as.character(S3Lo_M$V2))-0.2
S3Lo_M["V5"]<-as.numeric(as.character(S3Lo_M$V3))*5/3

C3Lz2T2T_M["V4"]<-as.numeric(as.character(C3Lz2T2T_M$V2))-0.8
C3Lz2T2T_M["V5"]<-as.numeric(as.character(C3Lz2T2T_M$V3))*5/3

S3Lz_M["V4"]<-as.numeric(as.character(S3Lz_M$V2))-0.1
S3Lz_M["V5"]<-as.numeric(as.character(S3Lz_M$V3))*5/3

S4S_M["V4"]<-as.numeric(as.character(S4S_M$V2))-0.8
S4S_M["V5"]<-as.numeric(as.character(S4S_M$V3))*5/3
```

```
C4S3T_M["V4"]<-as.numeric(as.character(C4S3T_M$V2))-0.9
C4S3T_M["V5"]<-as.numeric(as.character(C4S3T_M$V3))*5/3

C3A1Lo3S_M["V4"]<-as.numeric(as.character(C3A1Lo3S_M$V2))-0.6
C3A1Lo3S_M["V5"]<-as.numeric(as.character(C3A1Lo3S_M$V3))*5/3

pts_easy_after<-rep(0,n)
pts_hard_after<-rep(0,n)
for (i in 1:n){
  pts_easy_after[i]<-sample(as.numeric(as.character(S4T_M$V4)),1)+sample(as.numeric(as.character(S4T
    sample(as.numeric(as.character(S2S_M$V2)),1)+sample(as.numeric(as.character(S2S_M$V3)),1)+
    sample(as.numeric(as.character(S3F_M$V2)),1)+sample(as.numeric(as.character(S3F_M$V3)),1)+
    sample(as.numeric(as.character(FCCoSp4_M$V2)),1)+sample(as.numeric(as.character(FCCoSp4_M$V3)),1
    sample(as.numeric(as.character(StSq3_M$V2)),1)+sample(as.numeric(as.character(StSq3_M$V3)),1)+
    sample(as.numeric(as.character(C3A3T_M$V4)),1)+sample(as.numeric(as.character(C3A3T_M$V5)),1)+
    sample(as.numeric(as.character(C3A2T_M$V4)),1)+sample(as.numeric(as.character(C3A2T_M$V5)),1)+
    sample(as.numeric(as.character(S3Lo_M$V4)),1)+sample(as.numeric(as.character(S4T_M$V5)),1)+
    sample(as.numeric(as.character(C3Lz2T2T_M$V4)),1)+sample(as.numeric(as.character(C3Lz2T2T_M$V5))
    sample(as.numeric(as.character(S3Lz_M$V4)),1)+sample(as.numeric(as.character(S3Lz_M$V5)),1)+
    sample(as.numeric(as.character(ChSq1_M$V2)),1)+sample(as.numeric(as.character(ChSq1_M$V3)),1)+
    sample(as.numeric(as.character(CCoSp4_M$V2)),1)+sample(as.numeric(as.character(CCoSp4_M$V3)),1)+
    sample(as.numeric(as.character(FCSSp4_M$V2)),1)+sample(as.numeric(as.character(FCSSp4_M$V3)),1)+

  pts_hard_after[i]<-sample(as.numeric(as.character(S4T_M$V4)),1)+sample(as.numeric(as.character(S4T
    sample(as.numeric(as.character(S4T_M$V4)),1)+sample(as.numeric(as.character(S4T_M$V5)),1)+
    sample(as.numeric(as.character(S3F_M$V2)),1)+sample(as.numeric(as.character(S3F_M$V3)),1)+
    sample(as.numeric(as.character(FCCoSp4_M$V2)),1)+sample(as.numeric(as.character(FCCoSp4_M$V3)),1
    sample(as.numeric(as.character(StSq3_M$V2)),1)+sample(as.numeric(as.character(StSq3_M$V3)),1)+
    sample(as.numeric(as.character(C4S3T_M$V4)),1)+sample(as.numeric(as.character(C4S3T_M$V5)),1)+
    sample(as.numeric(as.character(S4T_M$V4)),1)+sample(as.numeric(as.character(S4T_M$V5)),1)+
    sample(as.numeric(as.character(S3Lz_M$V4)),1)+sample(as.numeric(as.character(S3Lz_M$V5)),1)+
    sample(as.numeric(as.character(C3A1Lo3S_M$V4)),1)+sample(as.numeric(as.character(C3A1Lo3S_M$V5))
    sample(as.numeric(as.character(S3Lo_M$V4)),1)+sample(as.numeric(as.character(S3Lo_M$V5)),1)+
    sample(as.numeric(as.character(ChSq1_M$V2)),1)+sample(as.numeric(as.character(ChSq1_M$V3)),1)+
    sample(as.numeric(as.character(CCoSp4_M$V2)),1)+sample(as.numeric(as.character(CCoSp4_M$V3)),1)+
    sample(as.numeric(as.character(FCSSp4_M$V2)),1)+sample(as.numeric(as.character(FCSSp4_M$V3)),1)+
}
par(mfrow=c(2,1))
hist(pts_easy_after,breaks=15,xlim=range(150,230),ylim=range(0,180),main="Easy vs. Hard After Rule C
hist(pts_hard_after,breaks=15,xlim=range(150,230),ylim=range(0,180),main="")
```