Central European University
Statistical Methods for Network Science
Final Project

PageRank Analysis and Motif Detection of “Svoya Igra” Players Network

by

Chyngyz Davletov

Professor: Rosario Nunzio Mantegna
Teaching Assistant: Luca Marotta

October 2015

Table of Contents

List of Tables

List of Figures

1 Introduction

"Svoya Igra" (further italicized) is the Russian analogue of the American intellectual TV game called "Jeopardy!". *Svoya Igra* has been broadcasted weekly across the post-soviet countries since 1994. The core of the game is that there are three players who compete in answering questions of different difficulty and earn/lose points for the right/wrong answers; the amount of points for a question increases with the difficulty of the question. The questions are taken from a variety of fields and are supposed to be answered by a person with a wide range of knowledge and ken. In this paper I perform the PageRank analysis of players for the network of *Svoya Igra* with different weighting settings and ways of network construction. Also, I implement the motif detection to see whether the network is over- or under expressed in terms of motifs.

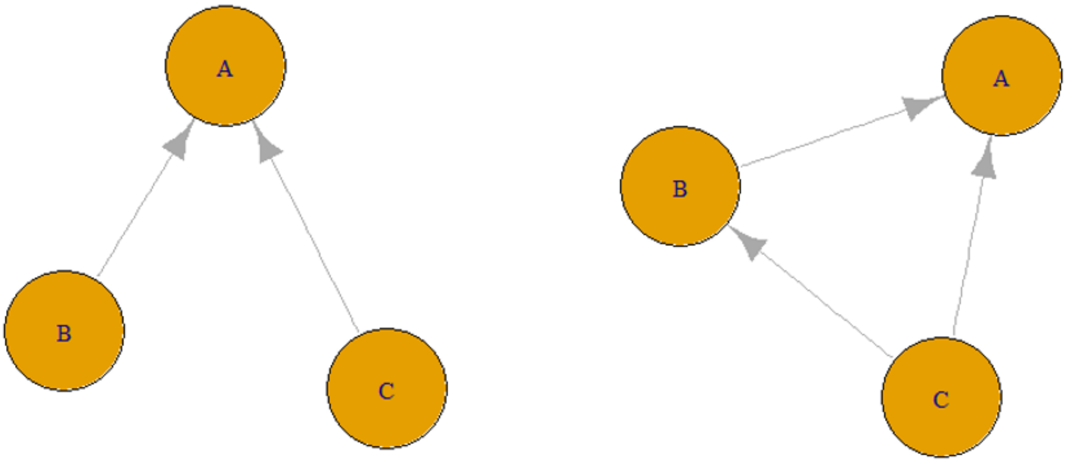
2 Data Set

Data were collected from the official web site of the game (2015)[1]. I downloaded the results for the games played from the early 1994 up until 2015. The first game dated as of April 7, 1994 and the last – November 21, 2015. Starting from 2001, each question is worth 10 times it was before 2001, and, besides, the rules of the game became a little different, hence I divide the data set into two samples – from 1994 to 2000 and from 2001 to 2015. This is important because in the analysis I use the difference in the earned points as the weights in the network. There was a certain amount of missing data, especially for the early period (such as there were missing cells for points earned by the end of the game by a player), hence those observations were deleted from the sample. Overall, there are 199 played games for the first (1994-2000) period and 1111 games for the second (2001-2015) period that had no missing data and, thus, used in this work. The reason for such difference in the amount of games played between two periods is, first of all, number of years, which is almost three times larger in the second period, and the accuracy with which the results of games were recorded, which is much more precise in the second period as well, which might be due to the increasing popularity of *Svoya Igra* over the years. The number of different players who participated in the first and second period is 159 and 673 respectively.

3 Network Representation

I represent the data set as a network of contacts between players. Suppose that in each game there are players A, B, and C. Each time player A earns more points than B by the end of the game (A wins B), I draw a link from B to A (B→A). So, if A wins the game, it means that A has won both B and C, thus we have two links: B→A and C→A. We can also think of the link between B and C (B→ C or C→B) if we are interested in the overall position of each player rather than concentrating on who is the winner only. Thus, I construct two networks: one, in which the links associated with the winner only are present (*network 1*), and the other, which takes into consideration the link between 2nd and 3rd player in the game (*network 2*). In other words, both networks are constructed by two types of motifs of size 3: one which has only two links between 3 nodes, and the other one, which is transitive and has 3 links between the nodes as shown in Figure 1.

[1] http://svoya-igra.org



(a) *Network 1* **(b)** *Network 2*

Figure 1: Motifs of which the networks were built-up: (a) only links from winner to losers are taken into account; (b) the link between 2^{nd} and 3^{rd} players is also present

4 Raw Statistics

It is always useful to start the analysis with observing some raw data. Table 1 contains the names of winners of some tournaments between 1994 and 2015. Observing the names of tournaments winners gives an insight on who might be the best player. Simple count reveals that Alexander Lieber has the most tournaments won – 7. Thus we expect him to be at least in the top three in the rankings.

Table 2 reports general statistics for the top ten players in terms of the games won. It can be inferred from the table that the largest number of games were won by Alexander Lieber who won 55 games of the 83 he participated in. It is also worth noting that players listed in the table win about 60 percent of games they participate in, except Alexander Druz, who has an incredibly high winning percentage of 83. However, he played only 35 games, whereas other listed players played almost twice as many games as he did. It is very likely that Druz's winning percentage would fall if he would play more games, because it is difficult to maintain such winning behaviour, in particular when playing against other strong players.

Finally, Table 3 provides the list with the maximum points earned by the end of a game. Alexander Druz here is the absolute record holder winning 120,001 points just for the one game. In general Druz earned a lot of points per game which is reflected in the table by several entries for him. There are other effective players such as Alexander Ediger, Anatoly Belkin, and Yuri Khashimov who earned substantial amount of points as well.

Table 1: Winners of Different Tournaments by Year (1994–2015)

Year, Tournament	Player	Year, Tournament	Player
1994	Alexei Tugarev	2005 I half	Alexander Uspanov
1996 I Cycle of Golden Dozen	Alexander Lieber	2005 II half	Andrei Zhdanov
1997 II Cycle of Golden Dozen	Alexander Lieber	2006 I Cycle of Golden Dozen	Yuri Khashimov
1997 III Cycle of Golden Dozen	Tatiana Bespalova	2006 II Cycle of Golden Dozen	Stanislav Mereminsky
1997 IV Cycle of Golden Dozen	Anatoly Belkin	2006 III Cycle of Golden Dozen	Valery Ovchinnikov
1997 Super Final of 1997	Anatoly Belkin	2006 IV Cycle of Golden Dozen	Vladislav Pristinsky
1998 V Cycle of Golden Dozen	Tatiana Bespalova	2006 V Cycle of Golden Dozen	Andrei Shtefan
1998 VI Cycle of Golden Dozen	Jacov Podolny	2006 Super Final	Andrei Zhdanov
1998 VII Cycle of Golden Dozen	Jacov Podolny	2007 VI Cycle of Golden Dozen	Dmitry Lurie
1998 Christmas Super Final	Mikhail Sakharov	2007 VII Cycle of Golden Dozen	Anatoly Belkin
1999 VIII Cycle of Golden Dozen	Alexander Lieber	2007 VIII Cycle of Golden Dozen	Dmitry Lurie
1999 IX Cycle of Golden Dozen	Yuri Bershidsky	2007 IX Cycle of Golden Dozen	Olga Uspanova
1999 X Cycle of Golden Dozen	Alexander Lieber	2007 X Cycle of Golden Dozen	Alexander Lieber
1999 Super Final of 1999	Yuri Bershidsky	2007 Super Final	Alexander Lieber
2000 XI Cycle of Golden Dozen	Yuri Bershidsky	2008 Final Games	Andrei Cherniavsky
2000 XII Cycle of Golden Dozen	Andrei Zhdanov	2009 Final Games	Stanislav Mereminsky
2000 XIII Cycle of Golden Dozen	Mikhail Sakharov	2010 Final Games	Yuri Khashimov
2001-2002 Cup 1	Anatoly Wasserman	2011 New-Year Tournament	Dmitry Lurie
2002 Cup 2	Yuri Bershidsky	2012 Championship	Alexander Lieber
2002 Cup 3	NA	2013 Super Final	Andrei Cherniavsky
2003 Cup 4	Alexander Druz	2014 I Cycle of Golden Dozen	Alexander Korobeinikov
2004	Anatoly Wasserman	2015 II Cycle of Golden Dozen	Alexander Korobeinikov

Table 2: Leaders by the Number of Wins

Rank	Player	Number of wins	Number of games	Percentage of wins
1	Alexander Lieber	55	83	66.27
2	Andrei Zhdanov	48	72	66.67
3	Jacov Podolny	44	67	65.67
4	Yuri Khashimov	41	62	66.13
5	Anatoly Wasserman	40	60	66.67
6	Stanislav Mereminsky	38	64	59.38
7	Anatoly Belkin	34	63	53.97
8	Alexander Ediger	32	50	64
9	Vladislav Pristinsky	32	54	59.26
10	Alexander Druz	29	35	82.86

Table 3: Maximum Earned Points for One Game

Rank	Player	Points
1	Alexander Druz	120 001
2	Alexander Druz	80 074
3	Anatoly Belkin	70 001
4	Yuri Khashimov	67 400
5	Alexander Ediger	63 401
6	Alexander Ediger	63400
7	Alexander Druz	60 501
8	Alexander Druz	60 000
9	Alexander Ediger	55 300
10	Alexei Markin	54 400
11	Anatoly Belkin	51 000
12	Jacov Podolny	50 300
13	Yuri Khashimov	50 000
14	Alexander Ediger	48 400
15	Yuri Khashimov	47 400
16	Dmitry Lurie	46 000
17	Alexander Ediger	45 800
18	Dmitry Lurie	45 401
19	Yuri Khashimov	45 399
20	Konstantin Shlykov	45 399

5 PageRank Analysis

To perform the PageRank analysis I used the built-in function page.rank() in R from igraph library. I perform two types of weighting for both periods' networks: 1) each time player A wins against player B he gets 1 as a weight; 2) each time player A wins against player B he gets the difference in the earned points between A and B as a weight.

Table 4 presents the results obtained for *network1* and *network2* of the PageRank for the first period (1994-2000) with both weighting options. Table 5 corresponds to the second period (2001-2015). From Table 4 it can be inferred that Alexander Lieber is ranked #1 in both networks and with both weighting options, thus he can be called the best player in the period from 1994 to 2000 according to the PageRank algorithm. Further, it is possible to observe some pattern in the table, namely the fact that there are players who go up in the ranking or fall down when the weighting option switches from 1 to DE (difference in earned points), that is, compare columns 2 and 3. For example, Jacov Podolny and Alexei Tugarev improved after this switch in weights with Podolny going from 3$^{\text{rd}}$ to 2$^{\text{nd}}$ position in the ranking and Tugarev from 4$^{\text{th}}$ to 3$^{\text{rd}}$. Those who fell down are Mikhail Sakharov (from 6 to 9), Irina Sobolevskaya (from7 to 8), Anatoly Belkin (from 10 to out-of-top-ten), and Tatiana Bespalova (from 2 to 4). How can this be explained? With weights equal to 1, each win is worth 1. But with weights equal to the difference in earned points, each win is worth much more, because difference in earned points almost always exceeds 1 by an order of hundred or thousand times, thus W=DE makes the link much stronger. It means that Podolny and Tugarev often won their games with a big advantage in the earned points over their opponents contrary to Sakharov, Sobolevskaya, Belkin, and Bespalova. Now, let us discuss what happens when there is a switch from network1 to *network2*. In *network2* there is now a link that represents the contact between players finished 2$^{\text{nd}}$ and 3$^{\text{rd}}$ – the link, which did not exist in *network1*. It turns out, that this is an important change in the network structure, because now players who did not win the game, but finished 2$^{\text{nd}}$, get a weight for beating a player who finished 3$^{\text{rd}}$, whereas in *network1* they did not get any. Therefore, this is a more accurate measure of a player's quality. Let us compare 2$^{\text{nd}}$ and 4$^{\text{th}}$ columns of Table 4. There is a fantastic improvement by Anatoly Belkin who grew in ranking from 10$^{\text{th}}$ to 2$^{\text{nd}}$ just because of this change in the network structure. This means that even when he would not win a game, he would more likely be 2$^{\text{nd}}$, but not 3$^{\text{rd}}$. The same logic applies in the explanation of other movements in the ranking due to the switch from *network1* to *network2*. Finally, let us discuss the 5th column, which contains top ten players for *network2* with difference in earned points as weights. Basically, the intuition is the same as when I compared columns 2 and 3, that is, each win weights much more than

Table 4: Top 10 Players in *Network 1* and *Network 2* for the 1$^{\text{st}}$ Period (1994–2000)

Rank	Network 1,W=1	Network 1,W=DE	Network 2,W=1	Network 2,W=DE
1	Alexander Lieber	Alexander Lieber	Alexander Lieber	Alexander Lieber
2	Tatiana Bespalova	Jacov Podolny	Anatoly Belkin	Jacov Podolny
3	Jacov Podolny	Alexei Tugarev	Jacov Podolny	Anatoly Belkin
4	Alexei Tugarev	Tatiana Bespalova	Mikhail Sakharov	Mikhail Sakharov
5	Valery Brun-Cekhovoi	Valery Brun-Cekhovoi	Tatiana Bespalova	Alexei Tugarev
6	Mikhail Sakharov	Andrei Abramov	Irina Sobolevskaya	Tatiana Bespalova
7	Irina Sobolevskaya	Jacov Zaydelman	Valery Brun-Cekhovoi	Dmitry Bielawski
8	Natalia Marfina	Irina Sobolevskaya	Alexei Tugarev	Jacov Zaydelman
9	Andrei Zhdanov	Mikhail Sakharov	Levon Hakobian	Valery Brun-Cekhovoi
10	Anatoly Belkin	Natalia Marfina	Andrei Zhdanov	Yuri Bershidsky

Notes: W=weights; DE=difference in earned points it was when weights were equal to 1. We observe that players who went up or down in the ranking when the switch was from column 2 to column 3 are moving in the same direction as in the switch from column 4 to column 5, as it should be.

The first thing to notice in Table 5 is that, in contrast with the first period, Alexander Lieber is not ranked #1 in any setting for the second period. However, he is still a top-player being 2$^{\text{nd}}$ in both networks with weights equal to 1 and 3$^{\text{rd}}$ in *network 2* with difference in earned points as weights. As before, it is better to compare the columns rather than just observing the rankings for different settings separately. So, we observe that Stanislav Merminsky fell from being ranked #1 to the 3$^{\text{rd}}$ position in the ranking when I change the weighting from 1 for the win to the difference in earned points between winner and losers for *network 1*. Applying the same reasoning as before, this change in rankings takes place because now the extent of the advantage a winner has over the losers is taken into account when ranking the players. Therefore it can be concluded that Mereminsky's victories were won under hard fight rather than with a distinct and confident advantage, and the same can be said about other players who fell down in the ranking, such as: Vladislav Pristinsky (from 3 to 10), Andrei Zhdanov (from 4 to 9), Jacov Podolny (from 9 to out-of-top-ten), Anatoly Belkin (from 10 to out-of-top-ten) and other players. The same pattern is observed if we compare columns 4 and 5. The player for whom the effect of the change in weights from 1 to DE is the largest in the top 10 is Anatoly Wasserman, who is considered as one of the biggest brains on the territory of CIS countries; the latitude and deepness of his knowledge gave birth to numerous anecdotes and nicknames about his extremely high erudition, beard, distinctive style of dress, and other features. In both networks when I change the weights from 1 to DE, Anatoly Wasserman reaches the top position in the ranking, which means that he usually won the games with a clear advantage over the losers. As it was mentioned above, `page.rank()` applied to *network 2* is a more accurate measure of a player's quality. However, looking at columns 4 and 5, there appears a question who is the best player: Mereminsky or Wasserman? There is no one-and-for-all answer to this question – for some it only matters whether you win or not, but for others it also matters the way you win. Hence, Mereminsky can be called the best player if we do not take into account how big his advantage was when he won his games, and Wasserman is the best player if we do care about it. It is also worth noting that Alexander Druz, another well-known brain together with Anatoly Wasserman, improved a lot in the ranking moving from out-of-top-ten to 5$^{\text{th}}$ and 6$^{\text{th}}$ positions in the ranking in both networks when weights were changed from 1 to DE, meaning that he often won games with a big advantage in terms of earned points. Indeed, we saw in Table 3 that Druz often earned big amounts of points per game.

Table 5: Top 10 Players in *Network 1* and *Network 2* for the 2$^{\text{nd}}$ Period (2001–2015)

Rank	Network 1,W=1	Network 1,W=DE	Network 2,W=1	Network 2,W=DE
1	Stanislav Mereminsky	Anatoly Wasserman	Stanislav Mereminsky	Anatoly Wasserman
2	Alexander Lieber	Alexander Lieber	Alexander Lieber	Stanislav Mereminsky
3	Vladislav Pristinsky	Stanislav Mereminsky	Anatoly Wasserman	Alexander Lieber
4	Andrei Zhdanov	Yuri Khashimov	Vladislav Pristinsky	Yuri Khashimov
5	Anatoly Wasserman	Alexander Druz	Andrei Zhdanov	Vladislav Pristinsky
6	Yuri Khashimov	Dmitry Lurie	Yuri Khashimov	Alexander Druz
7	Dmitry Lurie	Alexander Ediger	Jacov Podolny	Alexander Ediger
8	Valery Ovchinnikov	Pavel Shevchenko	Anatoly Belkin	Dmirty Lurie
9	Jacov Podolny	Andrei Zhdanov	Dmitry Lurie	Pavel Schevchenko
10	Anatoly Belkin	Vladislav Pristinsky	Valery Ovchinnikov	Jacov Podolny

Notes: W=weights; DE=difference in earned points

6 Motif Detection

In this section I present the results of motif detection that were obtained with the help of FANMOD program made by S.Wernicke and F.Rasche, which was based on the algorithms described in Wernicke (2005). Table 6 shows the number of different isomorphic motifs for both networks for each period. Tables 7–10 present first 5 most frequent 3-motifs for *network 1* and *network 2* for both periods, tables 11–14 provide same type of information on 4-motifs.

It can be inferred from Table 6 that there are up to 13 different 3-motifs and a lot more different 4-motifs present in the networks, which is how it should be, because there are many more ways of connections among 4 nodes than among just 3. It is worth noting that in *network 1* for 2001–2015 period and in *network 2* for both periods there is a maximum possible number of 3-nodes motifs present according to Uri Alon's motif dictionary with at least two arcs.

Table 6: Number of Distinct Isomorphic Motifs

Years Network	1994-2000 Network 1	2001-2015 Network 1	1994-2000 Network 2	2001-2015 Network 2
# of 3-motifs	10	13	13	13
# of 4-motifs	68	192	150	197

6.1 Size 3 Motifs

Although there are almost all different types of 3-motifs in *network 1* for 1994–2000 period and, simply, all – in *network 1* for 2001–2015 period and in both periods of *network 2*, many of those 3-motifs are not very frequent. Hence, I restrict the discussion to just several of them.

From tables 7–10 it is clear that the most frequent 3-motifs are those that have ID-s 36, 12, 6, 164, 38, and 14. The most frequent 3-motif is the one with ID 36, which is depicted in panel (a) Figure 1. Its frequency is the highest for *network 1* with values of 51.5% and 46.5% for first and second periods respectively, which is not a surprise, because *network 1* was constructed by this type of motif, i.e. without taking into account the link between players finished 2$^{\text{nd}}$ and 3$^{\text{rd}}$ in a game. Frequency of ID36 motif falls a little in *network 2* (still remaining the most frequent 3-motif) with corresponding values of 37.5% and 40.4% for first and second periods, which is also an expected outcome, because *network 2* was constructed from motifs, which are depicted in panel (b) Figure 1 or ID38 in FANMOD representation, i.e. in this network there is a link between players who finished 2$^{\text{nd}}$ and 3$^{\text{rd}}$ in a game. Thus, I also expected ID38 motif to become as frequent in *network 2* as ID36 was in *network 1*, which is, surprisingly, not the case. Moreover, although ID38 becomes more frequent in *network 2* than it was in *network 1*, the increase in frequency is not that serious. However, this is no longer a surprise, if we recall how *network 2* was constructed. Each game was transformed into 2 links (A wins B, A wins C) to obtain *network 1* and into 3 links (A wins B, A wins C, and C(B) wins B(C)) to obtain *network 2*. This means that the link between 2$^{\text{nd}}$ and 3$^{\text{rd}}$ players is just one of the total 3 links generated by each game to obtain *network 2*, thus ID36 is still the dominant 3-motif in *network 2*. It is also worth noting that ID38 is highly overrepresented with z-scores of around 30 in *network 2*.

Another very frequent 3-motif is the one with ID12. It has somewhat stable frequency around 33% and it is underrepresented in both networks for both periods. Its interpretation is quite natural. High frequency of ID12 reflects that there are many triads of players, say, A, B, and C, in which player A lost to B in one game and player B lost to C in another game, which means that there is no connection between players A and C in this type of a triad. If we draw arrows from A to B, and from B to C, the picture we get is exactly ID12 3-motif.

In both networks first 5 most frequent motifs have extreme p-values meaning that over- or under-representation of motifs is statistically significant. However the fact that z-scores are not very high means that motifs are slightly over- or underrepresented. ID36 is overrepresented in *network 1* in both periods, whereas in *network 2* it is overrepresented in the second period and underrepresented in the first.

Table 7: 3-Motifs for 1994–2000 Period for *Network 1*

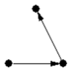
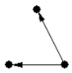
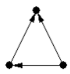
ID	Motif	Frequency[Original]	Mean-Freq[Random]	Z-Score	p-Value
36		51.5%	50.7%	4.1	0
12		34.3%	35%	-2.2	1
6		7.2%	8.3%	-4.5	1
164		3.4%	4%	-3.3	1
38		1.8%	0.7%	3.8	0

Table 8: 3-Motifs for 2001–2015 Period for *Network 1*

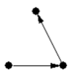
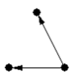
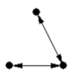
ID	Motif	Frequency[Original]	Mean-Freq[Random]	Z-Score	p-Value
36		46.5%	46.2%	6	0
12		32.8%	33%	-3	1
164		8.1%	8.3%	-3.6	1
6		8%	8.3%	-4.7	1
14		2%	2.2%	-4.5	1

Table 9: 3-Motifs for 1994–2000 Period for *Network 2*

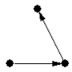
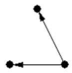
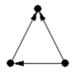
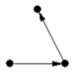
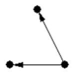
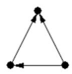
ID	Motif	Frequency[Original]	Mean-Freq[Random]	Z-Score	p-Value
36		37.5%	37.8%	-1.9	0.97
12		34.2%	35.8%	-7.6	1
6		9.1%	13.3%	-26.6	1
164		7.85%	8.45%	-3.3	1
38		6.7%	1.1%	28.1	0

Table 10: 3-Motifs for 2001–2015 Period for *Network 2*

ID	Motif	Frequency[Original]	Mean-Freq[Random]	Z-Score	p-Value
36		40.4%	40.3%	1.8	0.04
12		35%	35.4%	-6.2	1
6		9.7%	11.3%	-32	1
164		7.9%	8.2%	-6.6	1
38		3%	1%	32	0

6.2 Size 4 Motifs

Contrary to 3-motif detection, in each setting for the two networks there are the same 5 most frequent size 4 motifs[2], which have ID-s 392, 2184, 2116, 140, and 536. Again, contrary to 3-motif analysis, the frequency of 4-motifs is distributed quite evenly among them, so that the most frequent 4-motif in all network settings appears once in four motifs (25%, see Table 11), and the least frequent out of the five occurs only in 6.5% of cases (Table 12).

Table 11: 4-Motifs for 1994–2000 Period for *Network 1*

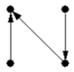
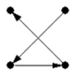
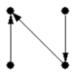
ID	Motif	Frequency[Original]	Mean-Freq[Random]	Z-Score	p-Value
392		25%	22.6%	1.4	0.09
2184		18.1%	19.6%	-2	1
2116		14.3%	16.5%	-2.8	1
140		10.7%	10.7%	0.02	0.5
536		7.5%	7.4%	0.13	0.5

Table 12: 4-Motifs for 2001–2015 Period for *Network 1*

ID	Motif	Frequency[Original]	Mean-Freq[Random]	Z-Score	p-Value
392		20.9%	19.7%	2.7	0
2184		17.2%	17.5%	-1.8	0.97
2116		13%	13.6%	-3.3	1
140		9.4%	9.7%	-2.1	0.98
536		6.5%	6.9%	-3.1	1

Interpretation of 4-motifs is more complicated compared to 3-motifs, because 4-motifs contain the contacts between players from (usually, 3 games) up to **4 different games**[3]. As an example, let us discuss the most frequent 4-motif – ID392. Let A be lower left node, B – upper left, C – upper right, and D – lower right. Then such direction of links as in ID392 means that A lost to B in one game,

[2] There were 6 most frequent, distinct 3-motifs.
[3] 3-motifs usually have contacts between 3 players from up to 3 different games.

D lost to B in another game, C lost to D in a third game, and A and C have no contact, i.e. did not participate in a common game. Thus there are 3 games reflected in this motif. Also, 4-motifs can reflect 4 different games as well, but, as we see it, this is a rare case in practice.

Finally, over- and underrepresentation of 4-motifs exhibits a visible trend for some motifs. Thus IDs 392, 2184, and 2116 keep the same direction in all network settings with ID392 being overrepresented, and the rest two – underrepresented. All three are statistically significantly over or underrepresented. For ID-s 140, 536 there is no such clear pattern. Again z-scores are not very high in absolute value except for some rare cases, meaning the over- or underrepresentation is not serious in terms of its size.

Table 13: 4-Motifs for 1994–2000 Period for *Network 2*

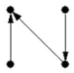
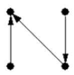
ID	Motif	Frequency[Original]	Mean-Freq[Random]	Z-Score	p-Value
392		18%	15.8%	2.85	0
2184		11%	12.6%	-6.4	1
2116		10.6%	12.2%	-5.6	1
140		9.2%	12%	-7.1	1
536		7.75%	7.66%	0.27	0.381

Table 14: 4-Motifs for 2001–2015 Period for *Network 2*

ID	Motif	Frequency[Original]	Mean-Freq[Random]	Z-Score	p-Value
392		19.2%	17.8%	4.1	0
2184		13.4%	14%	-5.7	1
2116		12.1%	12.8%	-5	1
140		9.8%	11.3%	-12.3	1
536		7.6%	7.8%	-2.1	0.98

7 Conclusion

In this paper I have performed PageRank analysis and motif detection for the network of "Svoya Igra" players. It turns out that, according to PageRank, the best player for the first period (1994–2000) is Alexander Lieber. As for the second period, there are two players identified as the best – Anatoly Wasserman and Stanislav Mereminsky, where the former is considered the best if we care about the advantage that a winner has over his/her opponents and the latter is ranked #1 if we care just about the fact of a win. Motif detection has revealed that for 3-motifs networks consist mainly from the motifs of the type which is depicted on panel (a) Figure 1 or ID36 in FANMOD notation with a little shift towards the motif which is shown on panel (b) Figure 1 or ID38 in *network 2*, which is an expected result. As for size 4 motifs, interpretation might be cumbersome, because such motifs may reflect the contacts between different players from up to 4 different games. Representation is significantly different from the mean of random graph simulation, however, in general, the difference is not large.

References

Svoya Igra (2015). Official Web Site. `http://www.svoya-igra.org`.

Wernicke, S. (2005). *A Faster Algorithm for Detecting Network Motifs*, pages 165–177. Springer Berlin Heidelberg, Berlin, Heidelberg.

Appendix A

```
setwd("C:/Users/User/Desktop/Чина/CEU/2nd Year/1.Fall2015/Statistical Methods/Project")
library(igraph)
library(taRifx)

### This script is for the network of type 2nd -> winner, 3rd -> winner ###
### i.e. it is about "absolute winner" of the game, without taking into account the link from player who finished 3rd to player who finished 2nd
###

#### Preparing the data ####

a1 <- read.csv("data1.csv",stringsAsFactors = FALSE)
a2 <- read.csv("data2.csv",stringsAsFactors = FALSE)
b1 <- data.frame()
b2 <- data.frame()
c1 <- data.frame()
c2 <- data.frame()

for (i in seq(from = 1, to = 1112)){
  b1[i,1] <- a1[2*i-1,3]
  b1[i,2] <- a1[2*i-1,4]
  b1[i,3] <- a1[2*i-1,5]
  c1[i,1] <- a1[2*i,3]
  c1[i,2] <- a1[2*i,4]
  c1[i,3] <- a1[2*i,5]
}

b1 <- b1[-59,] # empty cell => DELETE IT!!! =)
c1 <- c1[-59,] # empty cell => DELETE IT!!! =)

for (i in seq(from = 1, to = 199)){
  b2[i,1] <- a2[2*i-1,3]
  b2[i,2] <- a2[2*i-1,4]
  b2[i,3] <- a2[2*i-1,5]
  c2[i,1] <- a2[2*i,3]
  c2[i,2] <- a2[2*i,4]
  c2[i,3] <- a2[2*i,5]
}

# m1 and m2 to see what we got in previous loops
m1 <- cbind(b1,c1)
m2 <- cbind(b2,c2)

t1.1 <- destring(c1[,1])
t2.1 <- destring(c1[,2])
t3.1 <- destring(c1[,3])
p1.1 <- as.matrix(t1.1)
p2.1 <- as.matrix(t2.1)
p3.1 <- as.matrix(t3.1)

t1.2 <- destring(c2[,1])
t2.2 <- destring(c2[,2])
t3.2 <- destring(c2[,3])
p1.2 <- as.matrix(t1.2)
p2.2 <- as.matrix(t2.2)
p3.2 <- as.matrix(t3.2)

d1 <- cbind(p1.1,p2.1,p3.1)
d2 <- cbind(p1.2,p2.2,p3.2)

### Games from 2001 to 2015

e1.1 <- data.frame()
e2.1 <- data.frame()

f <- NA

for (i in seq(from = 1, to = 1111, by = 2)){
  if (d1[i,1] > d1[i,2] & d1[i,1] > d1[i,3] & d1[i,2] > d1[i,3]){
    e1.1[i,1] <- b1[i,1] #A
    e1.1[i,2] <- b1[i,2] #B
    e1.1[i,3] <- d1[i,1] - d1[i,2]
    e1.1[i+1,1] <- b1[i,1] #A
    e1.1[i+1,2] <- b1[i,3] #C
    e1.1[i+1,3] <- d1[i,1] - d1[i,3]
  } else {
    if (d1[i,1] > d1[i,2] & d1[i,1] > d1[i,3] & d1[i,3] > d1[i,2]){
      e1.1[i,1] <- b1[i,1] #A
      e1.1[i,2] <- b1[i,2] #B
      e1.1[i,3] <- d1[i,1] - d1[i,2]
      e1.1[i+1,1] <- b1[i,1] #A
      e1.1[i+1,2] <- b1[i,3] #C
      e1.1[i+1,3] <- d1[i,1] - d1[i,3]
    } else {
      if (d1[i,2] > d1[i,1] & d1[i,1] > d1[i,3] & d1[i,2] > d1[i,3]){
        e1.1[i,1] <- b1[i,2] #B
        e1.1[i,2] <- b1[i,1] #A
        e1.1[i,3] <- d1[i,2] - d1[i,1]
        e1.1[i+1,1] <- b1[i,2] #B
        e1.1[i+1,2] <- b1[i,3] #C
        e1.1[i+1,3] <- d1[i,2] - d1[i,3]
      } else {
        if (d1[i,2] > d1[i,1] & d1[i,1] > d1[i,3] & d1[i,3] > d1[i,2]){
          e1.1[i,1] <- b1[i,2] #B
          e1.1[i,2] <- b1[i,1] #A
          e1.1[i,3] <- d1[i,2] - d1[i,1]
          e1.1[i+1,1] <- b1[i,2] #B
          e1.1[i+1,2] <- b1[i,3] #C
          e1.1[i+1,3] <- d1[i,2] - d1[i,3]
        } else {
          if (d1[i,1] > d1[i,2] & d1[i,3] > d1[i,1] & d1[i,3] > d1[i,2]){
            e1.1[i,1] <- b1[i,3] #C
            e1.1[i,2] <- b1[i,1] #A
            e1.1[i,3] <- d1[i,3] - d1[i,1]
            e1.1[i+1,1] <- b1[i,3] #C
            e1.1[i+1,2] <- b1[i,2] #B
            e1.1[i+1,3] <- d1[i,3] - d1[i,2]
```

```
          } else {
            if (d1[i,2] > d1[i,1] & d1[i,3] > d1[i,1] & d1[i,3] > d1[i,2]){
              e1.1[i,1] <- b1[i,3] #C
              e1.1[i,2] <- b1[i,1] #A
              e1.1[i,3] <- d1[i,3] - d1[i,1]
              e1.1[i+1,1] <- b1[i,3] #C
              e1.1[i+1,2] <- b1[i,2] #B
              e1.1[i+1,3] <- d1[i,3] - d1[i,2]
            }
          }
        }
      }
    }
  }
}

for (i in seq(from = 2, to = 1111, by = 2)){
  if (d1[i,1] > d1[i,2] & d1[i,1] > d1[i,3] & d1[i,2] > d1[i,3]){
    e2.1[i,1] <- b1[i,1] #A
    e2.1[i,2] <- b1[i,2] #B
    e2.1[i,3] <- d1[i,1] - d1[i,2]
    e2.1[i+1,1] <- b1[i,1] #A
    e2.1[i+1,2] <- b1[i,3] #C
    e2.1[i+1,3] <- d1[i,1] - d1[i,3]
  } else {
    if (d1[i,1] > d1[i,2] & d1[i,1] > d1[i,3] & d1[i,3] > d1[i,2]){
      e2.1[i,1] <- b1[i,1] #A
      e2.1[i,2] <- b1[i,2] #B
      e2.1[i,3] <- d1[i,1] - d1[i,2]
      e2.1[i+1,1] <- b1[i,1] #A
      e2.1[i+1,2] <- b1[i,3] #C
      e2.1[i+1,3] <- d1[i,1] - d1[i,3]
    } else {
      if (d1[i,2] > d1[i,1] & d1[i,1] > d1[i,3] & d1[i,2] > d1[i,3]){
        e2.1[i,1] <- b1[i,2] #B
        e2.1[i,2] <- b1[i,1] #A
        e2.1[i,3] <- d1[i,2] - d1[i,1]
        e2.1[i+1,1] <- b1[i,2] #B
        e2.1[i+1,2] <- b1[i,3] #C
        e2.1[i+1,3] <- d1[i,2] - d1[i,3]
      } else {
        if (d1[i,2] > d1[i,1] & d1[i,1] > d1[i,3] & d1[i,3] > d1[i,2]){
          e2.1[i,1] <- b1[i,2] #B
          e2.1[i,2] <- b1[i,1] #A
          e2.1[i,3] <- d1[i,2] - d1[i,1]
          e2.1[i+1,1] <- b1[i,2] #B
          e2.1[i+1,2] <- b1[i,3] #C
          e2.1[i+1,3] <- d1[i,2] - d1[i,3]
        } else {
          if (d1[i,1] > d1[i,2] & d1[i,3] > d1[i,1] & d1[i,3] > d1[i,2]){
            e2.1[i,1] <- b1[i,3] #C
            e2.1[i,2] <- b1[i,1] #A
            e2.1[i,3] <- d1[i,3] - d1[i,1]
            e2.1[i+1,1] <- b1[i,3] #C
            e2.1[i+1,2] <- b1[i,2] #B
            e2.1[i+1,3] <- d1[i,3] - d1[i,2]
          } else {
            if (d1[i,2] > d1[i,1] & d1[i,3] > d1[i,1] & d1[i,3] > d1[i,2]){
              e2.1[i,1] <- b1[i,3] #C
              e2.1[i,2] <- b1[i,1] #A
              e2.1[i,3] <- d1[i,3] - d1[i,1]
              e2.1[i+1,1] <- b1[i,3] #C
              e2.1[i+1,2] <- b1[i,2] #B
              e2.1[i+1,3] <- d1[i,3] - d1[i,2]
            }
          }
        }
      }
    }
  }
}

### Games from 1994 to 2000 ###

e1.2 <- data.frame()
e2.2 <- data.frame()

for (i in seq(from = 1, to = 199, by = 2)){
  if (d2[i,1] > d2[i,2] & d2[i,1] > d2[i,3] & d2[i,2] > d2[i,3]){
    e1.2[i,1] <- b2[i,1] #A
    e1.2[i,2] <- b2[i,2] #B
    e1.2[i,3] <- d2[i,1] - d2[i,2]
    e1.2[i+1,1] <- b2[i,1] #A
    e1.2[i+1,2] <- b2[i,3] #C
    e1.2[i+1,3] <- d2[i,1] - d2[i,3]
  } else {
    if (d2[i,1] > d2[i,2] & d2[i,1] > d2[i,3] & d2[i,3] > d2[i,2]){
      e1.2[i,1] <- b2[i,1] #A
      e1.2[i,2] <- b2[i,2] #B
      e1.2[i,3] <- d2[i,1] - d2[i,2]
      e1.2[i+1,1] <- b2[i,1] #A
      e1.2[i+1,2] <- b2[i,3] #C
      e1.2[i+1,3] <- d2[i,1] - d2[i,3]
    } else {
      if (d2[i,2] > d2[i,1] & d2[i,1] > d2[i,3] & d2[i,2] > d2[i,3]){
        e1.2[i,1] <- b2[i,2] #B
        e1.2[i,2] <- b2[i,1] #A
        e1.2[i,3] <- d2[i,2] - d2[i,1]
        e1.2[i+1,1] <- b2[i,2] #B
        e1.2[i+1,2] <- b2[i,3] #C
        e1.2[i+1,3] <- d2[i,2] - d2[i,3]
      } else {
        if (d2[i,2] > d2[i,1] & d2[i,1] > d2[i,3] & d2[i,3] > d2[i,2]){
          e1.2[i,1] <- b2[i,2] #B
          e1.2[i,2] <- b2[i,1] #A
          e1.2[i,3] <- d2[i,2] - d2[i,1]
```

```
          e1.2[i+1,1] <- b2[i,2] #B
          e1.2[i+1,2] <- b2[i,3] #C
          e1.2[i+1,3] <- d2[i,2] - d2[i,3]
        } else {
          if (d2[i,1] > d2[i,2] & d2[i,3] > d2[i,1] & d2[i,3] > d2[i,2]){
            e1.2[i,1] <- b2[i,3] #C
            e1.2[i,2] <- b2[i,1] #A
            e1.2[i,3] <- d2[i,3] - d2[i,1]
            e1.2[i+1,1] <- b2[i,3] #C
            e1.2[i+1,2] <- b2[i,2] #B
            e1.2[i+1,3] <- d2[i,3] - d2[i,2]
          } else {
            if (d2[i,2] > d2[i,1] & d2[i,3] > d2[i,1] & d2[i,3] > d2[i,2]){
              e1.2[i,1] <- b2[i,3] #C
              e1.2[i,2] <- b2[i,1] #A
              e1.2[i,3] <- d2[i,3] - d2[i,1]
              e1.2[i+1,1] <- b2[i,3] #C
              e1.2[i+1,2] <- b2[i,2] #B
              e1.2[i+1,3] <- d2[i,3] - d2[i,2]
            }
          }
        }
      }
    }
  }
}

for (i in seq(from = 2, to = 199, by = 3)){
  if (d2[i,1] > d2[i,2] & d2[i,1] > d2[i,3] & d2[i,2] > d2[i,3]){
    e2.2[i,1] <- b2[i,1] #A
    e2.2[i,2] <- b2[i,2] #B
    e2.2[i,3] <- d2[i,1] - d2[i,2]
    e2.2[i+1,1] <- b2[i,1] #A
    e2.2[i+1,2] <- b2[i,3] #C
    e2.2[i+1,3] <- d2[i,1] - d1[i,3]
  } else {
    if (d2[i,1] > d2[i,2] & d2[i,1] > d2[i,3] & d2[i,3] > d2[i,2]){
      e2.2[i,1] <- b2[i,1] #A
      e2.2[i,2] <- b2[i,2] #B
      e2.2[i,3] <- d2[i,1] - d1[i,2]
      e2.2[i+1,1] <- b2[i,1] #A
      e2.2[i+1,2] <- b2[i,3] #C
      e2.2[i+1,3] <- d2[i,1] - d2[i,3]
    } else {
      if (d2[i,2] > d2[i,1] & d2[i,1] > d2[i,3] & d2[i,2] > d2[i,3]){
        e2.2[i,1] <- b2[i,2] #B
        e2.2[i,2] <- b2[i,1] #A
        e2.2[i,3] <- d2[i,2] - d2[i,1]
        e2.2[i+1,1] <- b2[i,2] #B
        e2.2[i+1,2] <- b2[i,3] #C
        e2.2[i+1,3] <- d2[i,2] - d2[i,3]
      } else {
        if (d2[i,2] > d2[i,1] & d2[i,1] > d2[i,3] & d2[i,3] > d2[i,2]){
          e2.2[i,1] <- b2[i,2] #B
          e2.2[i,2] <- b2[i,1] #A
          e2.2[i,3] <- d2[i,2] - d2[i,1]
          e2.2[i+1,1] <- b2[i,2] #B
          e2.2[i+1,2] <- b2[i,3] #C
          e2.2[i+1,3] <- d2[i,2] - d2[i,3]
        } else {
          if (d2[i,1] > d2[i,2] & d2[i,3] > d2[i,1] & d2[i,3] > d2[i,2]){
            e2.2[i,1] <- b2[i,3] #C
            e2.2[i,2] <- b2[i,1] #A
            e2.2[i,3] <- d2[i,3] - d2[i,1]
            e2.2[i+1,1] <- b2[i,3] #C
            e2.2[i+1,2] <- b2[i,2] #B
            e2.2[i+1,3] <- d2[i,3] - d2[i,2]
          } else {
            if (d2[i,2] > d2[i,1] & d2[i,3] > d2[i,1] & d2[i,3] > d2[i,2]){
              e2.2[i,1] <- b2[i,3] #C
              e2.2[i,2] <- b2[i,1] #A
              e2.2[i,3] <- d2[i,3] - d2[i,1]
              e2.2[i+1,1] <- b2[i,3] #C
              e2.2[i+1,2] <- b2[i,2] #B
              e2.2[i+1,3] <- d2[i,3] - d2[i,2]
            }
          }
        }
      }
    }
  }
}

r1.1 <- rbind(e1.1,e2.1)
r2.1 <- na.omit(r1.1)
r3.1 <- data.frame(r2.1[,2],r2.1[,1])

r1.2 <- rbind(e1.2,e2.2)
r2.2 <- na.omit(r1.2)
r3.2 <- data.frame(r2.2[,2],r2.2[,1])

g1 <- graph.data.frame(r3.1, directed = TRUE)
g1_ <- graph.data.frame(r3.1, directed = TRUE)
E(g1)$weight <- r2.1[,3]
E(g1_)$weight <- 1

g2 <- graph.data.frame(r3.2, directed = TRUE)
g2_ <- graph.data.frame(r3.2, directed = TRUE)
E(g2)$weight <- r2.2[,3]
E(g2_)$weight <- 1

g1.final <- simplify(g1)
g1_final <- simplify(g1_)
g2.final <- simplify(g2)
g2_final <- simplify(g2_)
```

```
#### Doing PageRank ####

pagerank.g1.final <- page.rank(g1.final, vids = V(g1.final), directed = TRUE) # with simplify() applied to g1; weights = difference in earnings for
each row and are set so by default,because -> line 302
pagerank.g1_final <- page.rank(g1_final, vids = V(g1_final), directed = TRUE) # with simplify() applied to g1_; weights = 1 for each row and are
set so by default,because -> line 303
pagerank.g2.final <- page.rank(g2.final, vids = V(g2.final), directed = TRUE) # with simplify() applied to g2; weights = difference in earnings for
each row and are set so by default,because -> line 307
pagerank.g2_final <- page.rank(g2_final, vids = V(g2_final), directed = TRUE) # with simplify() applied to g2_; weights = 1 for each row and are
set so by default,because -> line 308

head(sort(pagerank.g1.final$vector, decreasing=TRUE), 10)
head(sort(pagerank.g1_final$vector, decreasing=TRUE), 10)
head(sort(pagerank.g2.final$vector, decreasing=TRUE), 10)
head(sort(pagerank.g2_final$vector, decreasing=TRUE), 10)

#### Doing Motif Detection ####

q1 <- get.edgelist(g1.final,names = FALSE) # for period 2001-2015
q2 <- get.edgelist(g2.final,names = FALSE) # for period 1994-2000

w1 <- q1 - 1 # for subgraph size of 3; for period 2001-2015
w2 <- q2 - 1 # for subgraph size of 3; for period 1994-2000
w3 <- q1 - 1 # for subgraph size of 4; for period 2001-2015
w4 <- q2 - 1 # for subgraph size of 4; for period 1994-2000

write.table(w1, file = "w1.txt", sep = " ", row.names = FALSE, col.names = FALSE, append = FALSE)
write.table(w2, file = "w2.txt", sep = " ", row.names = FALSE, col.names = FALSE, append = FALSE)
write.table(w3, file = "w3.txt", sep = " ", row.names = FALSE, col.names = FALSE, append = FALSE)
write.table(w4, file = "w4.txt", sep = " ", row.names = FALSE, col.names = FALSE, append = FALSE)

### The rest is done using the "Fanmod" program ###
```

Appendix B

```
setwd("C:/Users/User/Desktop/Чина/CEU/2nd Year/1.Fall2015/Statistical Methods/Project")
library(igraph)
library(taRifx)

### This script is for the network of type 2nd -> winner, 3rd -> winner and 3rd -> 2nd ###
### i.e. it takes into account of the link beteen 3rd and 2nd players, contrary to the "FinalProject_a" ###

#### Preparing the data ####

a1 <- read.csv("data1.csv",stringsAsFactors = FALSE)
a2 <- read.csv("data2.csv",stringsAsFactors = FALSE)
b1 <- data.frame()
b2 <- data.frame()
c1 <- data.frame()
c2 <- data.frame()

for (i in seq(from = 1, to = 1112)){
  b1[i,1] <- a1[2*i-1,3]
  b1[i,2] <- a1[2*i-1,4]
  b1[i,3] <- a1[2*i-1,5]
  c1[i,1] <- a1[2*i,3]
  c1[i,2] <- a1[2*i,4]
  c1[i,3] <- a1[2*i,5]
}

b1 <- b1[-59,] # empty cell => DELETE IT!!! =)
c1 <- c1[-59,] # empty cell => DELETE IT!!! =)

for (i in seq(from = 1, to = 199)){
  b2[i,1] <- a2[2*i-1,3]
  b2[i,2] <- a2[2*i-1,4]
  b2[i,3] <- a2[2*i-1,5]
  c2[i,1] <- a2[2*i,3]
  c2[i,2] <- a2[2*i,4]
  c2[i,3] <- a2[2*i,5]
}

# m1 and m2 to see what we got in previous loops
m1 <- cbind(b1,c1)
m2 <- cbind(b2,c2)

t1.1 <- destring(c1[,1])
t2.1 <- destring(c1[,2])
t3.1 <- destring(c1[,3])
p1.1 <- as.matrix(t1.1)
p2.1 <- as.matrix(t2.1)
p3.1 <- as.matrix(t3.1)

t1.2 <- destring(c2[,1])
t2.2 <- destring(c2[,2])
t3.2 <- destring(c2[,3])
p1.2 <- as.matrix(t1.2)
p2.2 <- as.matrix(t2.2)
p3.2 <- as.matrix(t3.2)

d1 <- cbind(p1.1,p2.1,p3.1)
d2 <- cbind(p1.2,p2.2,p3.2)

### Games from 2001 to 2015

e1.1 <- data.frame()
e2.1 <- data.frame()
e3.1 <- data.frame()

f <- NA

for (i in seq(from = 1, to = 1111, by = 3)){
  if (d1[i,1] > d1[i,2] & d1[i,1] > d1[i,3] & d1[i,2] > d1[i,3]){
    e1.1[i,1] <- b1[i,1] #A
    e1.1[i,2] <- b1[i,2] #B
    e1.1[i,3] <- d1[i,1] - d1[i,2]
    e1.1[i+1,1] <- b1[i,1] #A
    e1.1[i+1,2] <- b1[i,3] #C
    e1.1[i+1,3] <- d1[i,1] - d1[i,3]
    e1.1[i+2,1] <- b1[i,2] #B
    e1.1[i+2,2] <- b1[i,3] #C
    e1.1[i+2,3] <- d1[i,2] - d1[i,3]
  } else {
    if (d1[i,1] > d1[i,2] & d1[i,1] > d1[i,3] & d1[i,3] > d1[i,2]){
      e1.1[i,1] <- b1[i,1] #A
      e1.1[i,2] <- b1[i,2] #B
      e1.1[i,3] <- d1[i,1] - d1[i,2]
      e1.1[i+1,1] <- b1[i,1] #A
      e1.1[i+1,2] <- b1[i,3] #C
      e1.1[i+1,3] <- d1[i,1] - d1[i,3]
      e1.1[i+2,1] <- b1[i,3] #C
      e1.1[i+2,2] <- b1[i,2] #B
      e1.1[i+2,3] <- d1[i,3] - d1[i,2]
    } else {
      if (d1[i,2] > d1[i,1] & d1[i,1] > d1[i,3] & d1[i,2] > d1[i,3]){
        e1.1[i,1] <- b1[i,2] #B
        e1.1[i,2] <- b1[i,1] #A
        e1.1[i,3] <- d1[i,2] - d1[i,1]
        e1.1[i+1,1] <- b1[i,2] #B
        e1.1[i+1,2] <- b1[i,3] #C
        e1.1[i+1,3] <- d1[i,2] - d1[i,3]
        e1.1[i+2,1] <- b1[i,1] #A
        e1.1[i+2,2] <- b1[i,3] #C
        e1.1[i+2,3] <- d1[i,1] - d1[i,3]
      } else {
        if (d1[i,2] > d1[i,1] & d1[i,1] > d1[i,3] & d1[i,3] > d1[i,2]){
          e1.1[i,1] <- b1[i,2] #B
          e1.1[i,2] <- b1[i,1] #A
          e1.1[i,3] <- d1[i,2] - d1[i,1]
          e1.1[i+1,1] <- b1[i,2] #B
          e1.1[i+1,2] <- b1[i,3] #C
```

```
          e1.1[i+1,3] <- d1[i,2] - d1[i,3]
          e1.1[i+2,1] <- b1[i,3] #C
          e1.1[i+2,2] <- b1[i,1] #A
          e1.1[i+2,3] <- d1[i,3] - d1[i,1]
        } else {
          if (d1[i,1] > d1[i,2] & d1[i,3] > d1[i,1] & d1[i,3] > d1[i,2]){
            e1.1[i,1] <- b1[i,3] #C
            e1.1[i,2] <- b1[i,1] #A
            e1.1[i,3] <- d1[i,3] - d1[i,1]
            e1.1[i+1,1] <- b1[i,3] #C
            e1.1[i+1,2] <- b1[i,2] #B
            e1.1[i+1,3] <- d1[i,3] - d1[i,2]
            e1.1[i+2,1] <- b1[i,1] #A
            e1.1[i+2,2] <- b1[i,2] #B
            e1.1[i+2,3] <- d1[i,1] - d1[i,2]
          } else {
            if (d1[i,2] > d1[i,1] & d1[i,3] > d1[i,1] & d1[i,3] > d1[i,2]){
              e1.1[i,1] <- b1[i,3] #C
              e1.1[i,2] <- b1[i,1] #A
              e1.1[i,3] <- d1[i,3] - d1[i,1]
              e1.1[i+1,1] <- b1[i,3] #C
              e1.1[i+1,2] <- b1[i,2] #B
              e1.1[i+1,3] <- d1[i,3] - d1[i,2]
              e1.1[i+2,1] <- b1[i,2] #B
              e1.1[i+2,2] <- b1[i,1] #A
              e1.1[i+2,3] <- d1[i,2] - d1[i,1]
            } else {
              if (d1[i,1] == d1[i,2] | d1[i,1] == d1[i,3] | d1[i,2] == d1[i,3]){
                e1.1[i,1] <- f #C
                e1.1[i,2] <- f #A
                e1.1[i,3] <- f
                e1.1[i+1,1] <- f #C
                e1.1[i+1,2] <- f #B
                e1.1[i+1,3] <- f
                e1.1[i+2,1] <- f #B
                e1.1[i+2,2] <- f #A
                e1.1[i+2,3] <- f
              }
            }
          }
        }
      }
    }
  }
}

for (i in seq(from = 2, to = 1111, by = 3)){
  if (d1[i,1] > d1[i,2] & d1[i,1] > d1[i,3] & d1[i,2] > d1[i,3]){
    e2.1[i,1] <- b1[i,1] #A
    e2.1[i,2] <- b1[i,2] #B
    e2.1[i,3] <- d1[i,1] - d1[i,2]
    e2.1[i+1,1] <- b1[i,1] #A
    e2.1[i+1,2] <- b1[i,3] #C
    e2.1[i+1,3] <- d1[i,1] - d1[i,3]
    e2.1[i+2,1] <- b1[i,2] #B
    e2.1[i+2,2] <- b1[i,3] #C
    e2.1[i+2,3] <- d1[i,2] - d1[i,3]
  } else {
    if (d1[i,1] > d1[i,2] & d1[i,1] > d1[i,3] & d1[i,3] > d1[i,2]){
      e2.1[i,1] <- b1[i,1] #A
      e2.1[i,2] <- b1[i,2] #B
      e2.1[i,3] <- d1[i,1] - d1[i,2]
      e2.1[i+1,1] <- b1[i,1] #A
      e2.1[i+1,2] <- b1[i,3] #C
      e2.1[i+1,3] <- d1[i,1] - d1[i,3]
      e2.1[i+2,1] <- b1[i,3] #C
      e2.1[i+2,2] <- b1[i,2] #B
      e2.1[i+2,3] <- d1[i,3] - d1[i,2]
    } else {
      if (d1[i,2] > d1[i,1] & d1[i,1] > d1[i,3] & d1[i,2] > d1[i,3]){
        e2.1[i,1] <- b1[i,2] #B
        e2.1[i,2] <- b1[i,1] #A
        e2.1[i,3] <- d1[i,2] - d1[i,1]
        e2.1[i+1,1] <- b1[i,2] #B
        e2.1[i+1,2] <- b1[i,3] #C
        e2.1[i+1,3] <- d1[i,2] - d1[i,3]
        e2.1[i+2,1] <- b1[i,1] #A
        e2.1[i+2,2] <- b1[i,3] #C
        e2.1[i+2,3] <- d1[i,1] - d1[i,3]
      } else {
        if (d1[i,2] > d1[i,1] & d1[i,1] > d1[i,3] & d1[i,3] > d1[i,2]){
          e2.1[i,1] <- b1[i,2] #B
          e2.1[i,2] <- b1[i,1] #A
          e2.1[i,3] <- d1[i,2] - d1[i,1]
          e2.1[i+1,1] <- b1[i,2] #B
          e2.1[i+1,2] <- b1[i,3] #C
          e2.1[i+1,3] <- d1[i,2] - d1[i,3]
          e2.1[i+2,1] <- b1[i,3] #C
          e2.1[i+2,2] <- b1[i,1] #A
          e2.1[i+2,3] <- d1[i,3] - d1[i,1]
        } else {
          if (d1[i,1] > d1[i,2] & d1[i,3] > d1[i,1] & d1[i,3] > d1[i,2]){
            e2.1[i,1] <- b1[i,3] #C
            e2.1[i,2] <- b1[i,1] #A
            e2.1[i,3] <- d1[i,3] - d1[i,1]
            e2.1[i+1,1] <- b1[i,3] #C
            e2.1[i+1,2] <- b1[i,2] #B
            e2.1[i+1,3] <- d1[i,3] - d1[i,2]
            e2.1[i+2,1] <- b1[i,1] #A
            e2.1[i+2,2] <- b1[i,2] #B
            e2.1[i+2,3] <- d1[i,1] - d1[i,2]
          } else {
            if (d1[i,2] > d1[i,1] & d1[i,3] > d1[i,1] & d1[i,3] > d1[i,2]){
              e2.1[i,1] <- b1[i,3] #C
              e2.1[i,2] <- b1[i,1] #A
              e2.1[i,3] <- d1[i,3] - d1[i,1]
```

```
            e2.1[i+1,1] <- b1[i,3] #C
            e2.1[i+1,2] <- b1[i,2] #B
            e2.1[i+1,3] <- d1[i,3] - d1[i,2]
            e2.1[i+2,1] <- b1[i,2] #B
            e2.1[i+2,2] <- b1[i,1] #A
            e2.1[i+2,3] <- d1[i,2] - d1[i,1]
          } else {
            if (d1[i,1] == d1[i,2] | d1[i,1] == d1[i,3] | d1[i,2] == d1[i,3]){
              e2.1[i,1] <- f #C
              e2.1[i,2] <- f #A
              e2.1[i,3] <- f
              e2.1[i+1,1] <- f #C
              e2.1[i+1,2] <- f #B
              e2.1[i+1,3] <- f
              e2.1[i+2,1] <- f #B
              e2.1[i+2,2] <- f #A
              e2.1[i+2,3] <- f
            }
          }
        }
      }
    }
  }
 }
}

for (i in seq(from = 3, to = 1111, by = 3)){
 if (d1[i,1] > d1[i,2] & d1[i,1] > d1[i,3] & d1[i,2] > d1[i,3]){
   e3.1[i,1] <- b1[i,1] #A
   e3.1[i,2] <- b1[i,2] #B
   e3.1[i,3] <- d1[i,1] - d1[i,2]
   e3.1[i+1,1] <- b1[i,1] #A
   e3.1[i+1,2] <- b1[i,3] #C
   e3.1[i+1,3] <- d1[i,1] - d1[i,3]
   e3.1[i+2,1] <- b1[i,2] #B
   e3.1[i+2,2] <- b1[i,3] #C
   e3.1[i+2,3] <- d1[i,2] - d1[i,3]
 } else {
   if (d1[i,1] > d1[i,2] & d1[i,1] > d1[i,3] & d1[i,3] > d1[i,2]){
     e3.1[i,1] <- b1[i,1] #A
     e3.1[i,2] <- b1[i,2] #B
     e3.1[i,3] <- d1[i,1] - d1[i,2]
     e3.1[i+1,1] <- b1[i,1] #A
     e3.1[i+1,2] <- b1[i,3] #C
     e3.1[i+1,3] <- d1[i,1] - d1[i,3]
     e3.1[i+2,1] <- b1[i,3] #C
     e3.1[i+2,2] <- b1[i,2] #B
     e3.1[i+2,3] <- d1[i,3] - d1[i,2]
   } else {
     if (d1[i,2] > d1[i,1] & d1[i,1] > d1[i,3] & d1[i,2] > d1[i,3]){
       e3.1[i,1] <- b1[i,2] #B
       e3.1[i,2] <- b1[i,1] #A
       e3.1[i,3] <- d1[i,2] - d1[i,1]
       e3.1[i+1,1] <- b1[i,2] #B
       e3.1[i+1,2] <- b1[i,3] #C
       e3.1[i+1,3] <- d1[i,2] - d1[i,3]
       e3.1[i+2,1] <- b1[i,1] #A
       e3.1[i+2,2] <- b1[i,3] #C
       e3.1[i+2,3] <- d1[i,1] - d1[i,3]
     } else {
       if (d1[i,2] > d1[i,1] & d1[i,1] > d1[i,3] & d1[i,3] > d1[i,2]){
         e3.1[i,1] <- b1[i,2] #B
         e3.1[i,2] <- b1[i,1] #A
         e3.1[i,3] <- d1[i,2] - d1[i,1]
         e3.1[i+1,1] <- b1[i,2] #B
         e3.1[i+1,2] <- b1[i,3] #C
         e3.1[i+1,3] <- d1[i,2] - d1[i,3]
         e3.1[i+2,1] <- b1[i,3] #C
         e3.1[i+2,2] <- b1[i,1] #A
         e3.1[i+2,3] <- d1[i,3] - d1[i,1]
       } else {
         if (d1[i,1] > d1[i,2] & d1[i,3] > d1[i,1] & d1[i,3] > d1[i,2]){
           e3.1[i,1] <- b1[i,3] #C
           e3.1[i,2] <- b1[i,1] #A
           e3.1[i,3] <- d1[i,3] - d1[i,1]
           e3.1[i+1,1] <- b1[i,3] #C
           e3.1[i+1,2] <- b1[i,2] #B
           e3.1[i+1,3] <- d1[i,3] - d1[i,2]
           e3.1[i+2,1] <- b1[i,1] #A
           e3.1[i+2,2] <- b1[i,2] #B
           e3.1[i+2,3] <- d1[i,1] - d1[i,2]
         } else {
           if (d1[i,2] > d1[i,1] & d1[i,3] > d1[i,1] & d1[i,3] > d1[i,2]){
             e3.1[i,1] <- b1[i,3] #C
             e3.1[i,2] <- b1[i,1] #A
             e3.1[i,3] <- d1[i,3] - d1[i,1]
             e3.1[i+1,1] <- b1[i,3] #C
             e3.1[i+1,2] <- b1[i,2] #B
             e3.1[i+1,3] <- d1[i,3] - d1[i,2]
             e3.1[i+2,1] <- b1[i,2] #B
             e3.1[i+2,2] <- b1[i,1] #A
             e3.1[i+2,3] <- d1[i,2] - d1[i,1]
           } else {
             if (d1[i,1] == d1[i,2] | d1[i,1] == d1[i,3] | d1[i,2] == d1[i,3]){
               e3.1[i,1] <- f #C
               e3.1[i,2] <- f #A
               e3.1[i,3] <- f
               e3.1[i+1,1] <- f #C
               e3.1[i+1,2] <- f #B
               e3.1[i+1,3] <- f
               e3.1[i+2,1] <- f #B
               e3.1[i+2,2] <- f #A
               e3.1[i+2,3] <- f
             }
           }
         }
```

```
        }
      }
    }
  }
}

### Games from 1994 to 2000 ###

e1.2 <- data.frame()
e2.2 <- data.frame()
e3.2 <- data.frame()

for (i in seq(from = 1, to = 199, by = 3)){
  if (d2[i,1] > d2[i,2] & d2[i,1] > d2[i,3] & d2[i,2] > d2[i,3]){
    e1.2[i,1] <- b2[i,1] #A
    e1.2[i,2] <- b2[i,2] #B
    e1.2[i,3] <- d2[i,1] - d2[i,2]
    e1.2[i+1,1] <- b2[i,1] #A
    e1.2[i+1,2] <- b2[i,3] #C
    e1.2[i+1,3] <- d2[i,1] - d2[i,3]
    e1.2[i+2,1] <- b2[i,2] #B
    e1.2[i+2,2] <- b2[i,3] #C
    e1.2[i+2,3] <- d2[i,2] - d2[i,3]
  } else {
    if (d2[i,1] > d2[i,2] & d2[i,1] > d2[i,3] & d2[i,3] > d2[i,2]){
      e1.2[i,1] <- b2[i,1] #A
      e1.2[i,2] <- b2[i,2] #B
      e1.2[i,3] <- d2[i,1] - d2[i,2]
      e1.2[i+1,1] <- b2[i,1] #A
      e1.2[i+1,2] <- b2[i,3] #C
      e1.2[i+1,3] <- d2[i,1] - d2[i,3]
      e1.2[i+2,1] <- b2[i,3] #C
      e1.2[i+2,2] <- b2[i,2] #B
      e1.2[i+2,3] <- d2[i,3] - d2[i,2]
    } else {
      if (d2[i,2] > d2[i,1] & d2[i,1] > d2[i,3] & d2[i,2] > d2[i,3]){
        e1.2[i,1] <- b2[i,2] #B
        e1.2[i,2] <- b2[i,1] #A
        e1.2[i,3] <- d2[i,2] - d2[i,1]
        e1.2[i+1,1] <- b2[i,2] #B
        e1.2[i+1,2] <- b2[i,3] #C
        e1.2[i+1,3] <- d2[i,2] - d2[i,3]
        e1.2[i+2,1] <- b2[i,1] #A
        e1.2[i+2,2] <- b2[i,3] #C
        e1.2[i+2,3] <- d2[i,1] - d2[i,3]
      } else {
        if (d2[i,2] > d2[i,1] & d2[i,1] > d2[i,3] & d2[i,3] > d2[i,2]){
          e1.2[i,1] <- b2[i,2] #B
          e1.2[i,2] <- b2[i,1] #A
          e1.2[i,3] <- d2[i,2] - d2[i,1]
          e1.2[i+1,1] <- b2[i,2] #B
          e1.2[i+1,2] <- b2[i,3] #C
          e1.2[i+1,3] <- d2[i,2] - d2[i,3]
          e1.2[i+2,1] <- b2[i,3] #C
          e1.2[i+2,2] <- b2[i,1] #A
          e1.2[i+2,3] <- d2[i,3] - d2[i,1]
        } else {
          if (d2[i,1] > d2[i,2] & d2[i,3] > d2[i,1] & d2[i,3] > d2[i,2]){
            e1.2[i,1] <- b2[i,3] #C
            e1.2[i,2] <- b2[i,1] #A
            e1.2[i,3] <- d2[i,3] - d2[i,1]
            e1.2[i+1,1] <- b2[i,3] #C
            e1.2[i+1,2] <- b2[i,2] #B
            e1.2[i+1,3] <- d2[i,3] - d2[i,2]
            e1.2[i+2,1] <- b2[i,1] #A
            e1.2[i+2,2] <- b2[i,2] #B
            e1.2[i+2,3] <- d2[i,1] - d2[i,2]
          } else {
            if (d2[i,2] > d2[i,1] & d2[i,3] > d2[i,1] & d2[i,3] > d2[i,2]){
              e1.2[i,1] <- b2[i,3] #C
              e1.2[i,2] <- b2[i,1] #A
              e1.2[i,3] <- d2[i,3] - d2[i,1]
              e1.2[i+1,1] <- b2[i,3] #C
              e1.2[i+1,2] <- b2[i,2] #B
              e1.2[i+1,3] <- d2[i,3] - d2[i,2]
              e1.2[i+2,1] <- b2[i,2] #B
              e1.2[i+2,2] <- b2[i,1] #A
              e1.2[i+2,3] <- d2[i,2] - d2[i,1]
            } else {
              if (d2[i,1] == d2[i,2] | d2[i,1] == d2[i,3] | d2[i,2] == d2[i,3]){
                e1.2[i,1] <- f #C
                e1.2[i,2] <- f #A
                e1.2[i,3] <- f
                e1.2[i+1,1] <- f #C
                e1.2[i+1,2] <- f #B
                e1.2[i+1,3] <- f
                e1.2[i+2,1] <- f #B
                e1.2[i+2,2] <- f #A
                e1.2[i+2,3] <- f
              }
            }
          }
        }
      }
    }
  }
}

for (i in seq(from = 2, to = 199, by = 3)){
  if (d2[i,1] > d2[i,2] & d2[i,1] > d2[i,3] & d2[i,2] > d2[i,3]){
    e2.2[i,1] <- b2[i,1] #A
    e2.2[i,2] <- b2[i,2] #B
    e2.2[i,3] <- d2[i,1] - d2[i,2]
    e2.2[i+1,1] <- b2[i,1] #A
    e2.2[i+1,2] <- b2[i,3] #C
    e2.2[i+1,3] <- d2[i,1] - d1[i,3]
```

```
    e2.2[i+2,1] <- b2[i,2] #B
    e2.2[i+2,2] <- b2[i,3] #C
    e2.2[i+2,3] <- d2[i,2] - d2[i,3]
  } else {
    if (d2[i,1] > d2[i,2] & d2[i,1] > d2[i,3] & d2[i,3] > d2[i,2]){
      e2.2[i,1] <- b2[i,1] #A
      e2.2[i,2] <- b2[i,2] #B
      e2.2[i,3] <- d2[i,1] - d1[i,2]
      e2.2[i+1,1] <- b2[i,1] #A
      e2.2[i+1,2] <- b2[i,3] #C
      e2.2[i+1,3] <- d2[i,1] - d2[i,3]
      e2.2[i+2,1] <- b2[i,3] #C
      e2.2[i+2,2] <- b2[i,2] #B
      e2.2[i+2,3] <- d2[i,3] - d2[i,2]
    } else {
      if (d2[i,2] > d2[i,1] & d2[i,1] > d2[i,3] & d2[i,2] > d2[i,3]){
        e2.2[i,1] <- b2[i,2] #B
        e2.2[i,2] <- b2[i,1] #A
        e2.2[i,3] <- d2[i,2] - d2[i,1]
        e2.2[i+1,1] <- b2[i,2] #B
        e2.2[i+1,2] <- b2[i,3] #C
        e2.2[i+1,3] <- d2[i,2] - d2[i,3]
        e2.2[i+2,1] <- b2[i,1] #A
        e2.2[i+2,2] <- b2[i,3] #C
        e2.2[i+2,3] <- d2[i,1] - d2[i,3]
      } else {
        if (d2[i,2] > d2[i,1] & d2[i,1] > d2[i,3] & d2[i,3] > d2[i,2]){
          e2.2[i,1] <- b2[i,2] #B
          e2.2[i,2] <- b2[i,1] #A
          e2.2[i,3] <- d2[i,2] - d2[i,1]
          e2.2[i+1,1] <- b2[i,2] #B
          e2.2[i+1,2] <- b2[i,3] #C
          e2.2[i+1,3] <- d2[i,2] - d2[i,3]
          e2.2[i+2,1] <- b2[i,3] #C
          e2.2[i+2,2] <- b2[i,1] #A
          e2.2[i+2,3] <- d2[i,3] - d2[i,1]
        } else {
          if (d2[i,1] > d2[i,2] & d2[i,3] > d2[i,1] & d2[i,3] > d2[i,2]){
            e2.2[i,1] <- b2[i,3] #C
            e2.2[i,2] <- b2[i,1] #A
            e2.2[i,3] <- d2[i,3] - d2[i,1]
            e2.2[i+1,1] <- b2[i,3] #C
            e2.2[i+1,2] <- b2[i,2] #B
            e2.2[i+1,3] <- d2[i,3] - d2[i,2]
            e2.2[i+2,1] <- b2[i,1] #A
            e2.2[i+2,2] <- b2[i,2] #B
            e2.2[i+2,3] <- d2[i,1] - d2[i,2]
          } else {
            if (d2[i,2] > d2[i,1] & d2[i,3] > d2[i,1] & d2[i,3] > d2[i,2]){
              e2.2[i,1] <- b2[i,3] #C
              e2.2[i,2] <- b2[i,1] #A
              e2.2[i,3] <- d2[i,3] - d2[i,1]
              e2.2[i+1,1] <- b2[i,3] #C
              e2.2[i+1,2] <- b2[i,2] #B
              e2.2[i+1,3] <- d2[i,3] - d2[i,2]
              e2.2[i+2,1] <- b2[i,2] #B
              e2.2[i+2,2] <- b2[i,1] #A
              e2.2[i+2,3] <- d2[i,2] - d2[i,1]
            } else {
              if (d2[i,1] == d2[i,2] | d2[i,1] == d2[i,3] | d2[i,2] == d2[i,3]){
                e2.2[i,1] <- f #C
                e2.2[i,2] <- f #A
                e2.2[i,3] <- f
                e2.2[i+1,1] <- f #C
                e2.2[i+1,2] <- f #B
                e2.2[i+1,3] <- f
                e2.2[i+2,1] <- f #B
                e2.2[i+2,2] <- f #A
                e2.2[i+2,3] <- f
              }
            }
          }
        }
      }
    }
  }
}

for (i in seq(from = 3, to = 199, by = 3)){
  if (d2[i,1] > d2[i,2] & d2[i,1] > d2[i,3] & d2[i,2] > d2[i,3]){
    e3.2[i,1] <- b2[i,1] #A
    e3.2[i,2] <- b2[i,2] #B
    e3.2[i,3] <- d2[i,1] - d2[i,2]
    e3.2[i+1,1] <- b2[i,1] #A
    e3.2[i+1,2] <- b2[i,3] #C
    e3.2[i+1,3] <- d2[i,1] - d2[i,3]
    e3.2[i+2,1] <- b2[i,2] #B
    e3.2[i+2,2] <- b2[i,3] #C
    e3.2[i+2,3] <- d2[i,2] - d2[i,3]
  } else {
    if (d2[i,1] > d2[i,2] & d2[i,1] > d2[i,3] & d2[i,3] > d2[i,2]){
      e3.2[i,1] <- b2[i,1] #A
      e3.2[i,2] <- b2[i,2] #B
      e3.2[i,3] <- d2[i,1] - d2[i,2]
      e3.2[i+1,1] <- b2[i,1] #A
      e3.2[i+1,2] <- b2[i,3] #C
      e3.2[i+1,3] <- d2[i,1] - d2[i,3]
      e3.2[i+2,1] <- b2[i,3] #C
      e3.2[i+2,2] <- b2[i,2] #B
      e3.2[i+2,3] <- d2[i,3] - d2[i,2]
    } else {
      if (d2[i,2] > d2[i,1] & d2[i,1] > d2[i,3] & d2[i,2] > d2[i,3]){
        e3.2[i,1] <- b2[i,2] #B
        e3.2[i,2] <- b2[i,1] #A
        e3.2[i,3] <- d2[i,2] - d2[i,1]
        e3.2[i+1,1] <- b2[i,2] #B
```

```
        e3.2[i+1,2] <- b2[i,3] #C
        e3.2[i+1,3] <- d2[i,2] - d2[i,3]
        e3.2[i+2,1] <- b2[i,1] #A
        e3.2[i+2,2] <- b2[i,3] #C
        e3.2[i+2,3] <- d2[i,1] - d2[i,3]
      } else {
        if (d2[i,2] > d2[i,1] & d2[i,1] > d2[i,3] & d2[i,3] > d2[i,2]){
          e3.2[i,1] <- b2[i,2] #B
          e3.2[i,2] <- b2[i,1] #A
          e3.2[i,3] <- d2[i,2] - d2[i,1]
          e3.2[i+1,1] <- b2[i,2] #B
          e3.2[i+1,2] <- b2[i,3] #C
          e3.2[i+1,3] <- d2[i,2] - d2[i,3]
          e3.2[i+2,1] <- b2[i,3] #C
          e3.2[i+2,2] <- b2[i,1] #A
          e3.2[i+2,3] <- d2[i,3] - d2[i,1]
        } else {
          if (d2[i,1] > d2[i,2] & d2[i,3] > d2[i,1] & d2[i,3] > d2[i,2]){
            e3.2[i,1] <- b2[i,3] #C
            e3.2[i,2] <- b2[i,1] #A
            e3.2[i,3] <- d2[i,3] - d2[i,1]
            e3.2[i+1,1] <- b2[i,3] #C
            e3.2[i+1,2] <- b2[i,2] #B
            e3.2[i+1,3] <- d2[i,3] - d2[i,2]
            e3.2[i+2,1] <- b2[i,1] #A
            e3.2[i+2,2] <- b2[i,2] #B
            e3.2[i+2,3] <- d2[i,1] - d2[i,2]
          } else {
            if (d2[i,2] > d2[i,1] & d2[i,3] > d2[i,1] & d2[i,3] > d2[i,2]){
              e3.2[i,1] <- b2[i,3] #C
              e3.2[i,2] <- b2[i,1] #A
              e3.2[i,3] <- d2[i,3] - d2[i,1]
              e3.2[i+1,1] <- b2[i,3] #C
              e3.2[i+1,2] <- b2[i,2] #B
              e3.2[i+1,3] <- d2[i,3] - d2[i,2]
              e3.2[i+2,1] <- b2[i,2] #B
              e3.2[i+2,2] <- b2[i,1] #A
              e3.2[i+2,3] <- d2[i,2] - d2[i,1]
            } else {
              if (d2[i,1] == d2[i,2] | d2[i,1] == d2[i,3] | d2[i,2] == d2[i,3]){
                e3.2[i,1] <- f #C
                e3.2[i,2] <- f #A
                e3.2[i,3] <- f
                e3.2[i+1,1] <- f #C
                e3.2[i+1,2] <- f #B
                e3.2[i+1,3] <- f
                e3.2[i+2,1] <- f #B
                e3.2[i+2,2] <- f #A
                e3.2[i+2,3] <- f
              }
            }
          }
        }
      }
    }
  }
}

r1.1 <- rbind(e1.1,e2.1,e3.1)
r2.1 <- na.omit(r1.1)
r3.1 <- data.frame(r2.1[,2],r2.1[,1])

r1.2 <- rbind(e1.2,e2.2,e3.2)
r2.2 <- na.omit(r1.2)
r3.2 <- data.frame(r2.2[,2],r2.2[,1])

g1 <- graph.data.frame(r3.1, directed = TRUE)
g1_ <- graph.data.frame(r3.1, directed = TRUE)
E(g1)$weight <- r2.1[,3]
E(g1_)$weight <- 1

g2 <- graph.data.frame(r3.2, directed = TRUE)
g2_ <- graph.data.frame(r3.2, directed = TRUE)
E(g2)$weight <- r2.2[,3]
E(g2_)$weight <- 1

g1.final <- simplify(g1)
g1_final <- simplify(g1_)
g2.final <- simplify(g2)
g2_final <- simplify(g2_)

#### Doing PageRank ####

pagerank.g1.final <- page.rank(g1.final, vids = V(g1.final), directed = TRUE) # with simplify() applied to g1; weights = difference in earnings for
each row and are set so by default,because -> line 302
pagerank.g1_final <- page.rank(g1_final, vids = V(g1_final), directed = TRUE) # with simplify() applied to g1_; weights = 1 for each row and are
set so by default,because -> line 303
pagerank.g2.final <- page.rank(g2.final, vids = V(g2.final), directed = TRUE) # with simplify() applied to g2; weights = difference in earnings for
each row and are set so by default,because -> line 307
pagerank.g2_final <- page.rank(g2_final, vids = V(g2_final), directed = TRUE) # with simplify() applied to g2_; weights = 1 for each row and are
set so by default,because -> line 308

head(sort(pagerank.g1.final$vector, decreasing=TRUE), 10)
head(sort(pagerank.g1_final$vector, decreasing=TRUE), 10)
head(sort(pagerank.g2.final$vector, decreasing=TRUE), 10)
head(sort(pagerank.g2_final$vector, decreasing=TRUE), 10)

#### Doing Motif Detection ####

q1 <- get.edgelist(g1.final,names = FALSE) # for period 2001-2015
q2 <- get.edgelist(g2.final,names = FALSE) # for period 1994-2000

w5 <- q1 - 1 # for subgraph size of 3; for period 2001-2015
w6 <- q2 - 1 # for subgraph size of 3; for period 1994-2000
w7 <- q1 - 1 # for subgraph size of 4; for period 2001-2015
w8 <- q2 - 1 # for subgraph size of 4; for period 1994-2000
```

```
write.table(w5, file = "w5.txt", sep = " ", row.names = FALSE, col.names = FALSE, append = FALSE)
write.table(w6, file = "w6.txt", sep = " ", row.names = FALSE, col.names = FALSE, append = FALSE)
write.table(w7, file = "w7.txt", sep = " ", row.names = FALSE, col.names = FALSE, append = FALSE)
write.table(w8, file = "w8.txt", sep = " ", row.names = FALSE, col.names = FALSE, append = FALSE)

### The rest is done using the "Fanmod" program ###
degree(g2.final)[order(degree(g2.final), decreasing=TRUE)[1:10]]
```