由赛事官方提供的三组数据fullevents.csv、passingevents.csv、matches.csv由于文件格式的限制，在一定程度上不能很好地兼容多个处理软件、多个编程语言，需要对这三组数据进行数据预处理以发挥各种不同的软件、编程语言对数据不同方面处理的优势。

对这三组数据的预处理首先采用Excel中的函数和VBA编程进行粗略地预处理，处理过程满足以下公式：

* matches.csv:

OpponentID列处理过程：

其中满足:

Outcome列处理过程:

Side列处理过程同Outcome列，CoachID列处理过程同OpponentID列。

图片一 图片二

（两图片是处理结果的对比）

* passingevents.csv：

OriginPlayerID、DestinationPlayerID的处理过程：

将原来的不等长字符串转化为固定4位长度的字符和数字的组合，其中第一位表示球员的ID号，由于有ID大于10，故采用十六进制存储第一位。第二位为球员的种类，F为1，D为2，M为3，G为4。最后两位为比赛队伍的编号，00位哈士奇队，其余为各自的队伍ID。

EventSubType列处理规则：(最好做个好看的表格)

|  |  |
| --- | --- |
| Head pass | 1 |
| Simple pass | 2 |
| Launch | 3 |
| High pass | 4 |
| Hand pass | 5 |
| Smart pass | 6 |
| Cross | 7 |

图片三 图片四

（处理前和处理后的对比）

* fullevents.csv

TeamID、OriginPlayerID、DestinationPlayerID列处理同上，DestinationPlayerID空位补-1

EventType处理规则：

|  |  |
| --- | --- |
| Free Kick | 1 |
| Duel | 2 |
| Pass | 3 |
| Others on the ball | 4 |
| Foul | 5 |
| Goalkeeper leaving line | 6 |
| Offside | 7 |
| Save attempt | 8 |
| Shot | 9 |
| Substitution | 10 |
| Interruption | 11 |

EventSubType列处理规则：

|  |  |
| --- | --- |
| Goal kick | 1 |
| Air duel | 2 |
| Throw in | 3 |
| Head pass | 4 |
| Ground loose ball duel | 5 |
| Simple pass | 6 |
| Launch | 7 |
| High pass | 8 |
| Touch | 9 |
| Ground defending duel | 10 |
| Hand pass | 11 |
| Ground attacking duel | 12 |
| Foul | 13 |
| Free kick cross | 14 |
| Goalkeeper leaving line | 15 |
| ε | 16 |
| Free Kick | 17 |
| Smart pass | 18 |
| Cross | 19 |
| Save attempt | 20 |
| Corner | 21 |
| Clearance | 22 |
| Shot | 23 |
| Acceleration | 24 |
| Reflexes | 25 |
| Substitution | 26 |
| Late card foul | 27 |
| Simulation | 28 |
| Free kick shot | 29 |
| Protest | 30 |
| Hand foul | 31 |
| Penalty | 32 |
| Violent Foul | 33 |
| Whistle | 34 |
| Out of game foul | 35 |
| Ball out of the field | 36 |
| Time lost foul | 37 |