Squash

Ajit Singh

24 June 2017

```
#Loading required libraries
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(kernlab)
```

```
##
## Attaching package: 'kernlab'
```

```
## The following object is masked from 'package:ggplot2':
##
##     alpha
```

```
library(ggplot2)
library(cvTools)
```

```
## Loading required package: robustbase
```

```
library(caTools)

#import data
sqAct1 <- read.table("Squash1PlayerActivity.txt")
sqAct2 <- read.table("Squash2PlayerActivity.txt")
sqPos1 <- read.table("squash1positions.txt")
sqPos2 <- read.table("squash2positions.txt")

nameHeadAct <- c("Frames","Seconds","StrokePlayer","StrokeType","StrokeOutcome","ForehandBack
hand","StrokeX","StrokeY")

#Change header names
sqAct1 <- sqAct1[2:132,]
head(sqAct1)
```

```
##    V1    V2 V3  V4 V5 V6  V7  V8
## 2  63  2.48  1   S  n  b 275 118
## 3 224  8.92  2   S  i  f 245  32
## 4 269 10.72  1 LOL  i  b 348 166
## 5 301    12  2 VLL  i  b 290 249
## 6 336  13.4  1   C  i  b 220 168
## 7 364 14.52  2   C  i  f 298  48
```

```
colnames(sqAct1) <- nameHeadAct

#Manipulate data by changing char to num
sqAct1$StrokeType <- factor(sqAct1$StrokeType, levels =
c("S","LL","C","CS","DL","DC","LOL","LOC","KL","KC","VLL","VC","VCS","VDL","VDC","VBN","VBO","V
BR","VKL","VKC","BN","BO","BR","BS","COS"),labels = c("1","2","3","4","5","6","7","8","9","1
0","11","12","13","14","15","16","17","18","19","20","21","22","23","24","25"))

sqAct1$StrokeOutcome <- factor(sqAct1$StrokeOutcome, levels = c("i","l","s","t","n"), labels
= c("1","2","3","4","5"))

sqAct1$ForehandBackhand <- factor(sqAct1$ForehandBackhand, levels = c("f","b"), labels =
c("1","2"))

#shifting 4th column to the last column
sqAct1 <- sqAct1[, c(1,2,3,5,6,7,8,4)]

#randomizing the data
sqAct1 <- sqAct1[sample(nrow(sqAct1)),]
str(sqAct1)
```

```
## 'data.frame':    131 obs. of  8 variables:
##  $ Frames          : Factor w/ 132 levels "1000","1023",..: 56 32 93 48 95 128 13 10 74 78
...
##  $ Seconds         : Factor w/ 132 levels "10.72","105.48",..: 9 116 47 2 49 81 97 94 27 3
1 ...
##  $ StrokePlayer    : Factor w/ 3 levels "1","2","StrokePlayer": 1 2 2 2 2 2 1 1 2 1 ...
##  $ StrokeOutcome   : Factor w/ 5 levels "1","2","3","4",..: 1 5 1 1 4 1 1 1 5 1 ...
##  $ ForehandBackhand: Factor w/ 2 levels "1","2": 1 1 1 1 1 2 1 1 2 1 ...
##  $ StrokeX         : Factor w/ 91 levels "135","136","144",..: 77 20 22 13 49 81 69 8 79 6
4 ...
##  $ StrokeY         : Factor w/ 98 levels "101","103","104",..: 81 74 73 35 1 47 22 13 56 7
5 ...
##  $ StrokeType      : Factor w/ 25 levels "1","2","3","4",..: 2 1 1 1 19 22 1 3 17 2 ...
```

```
#transforming each column to numeric data type from factor type
sqAct1[nameHeadAct] <- lapply(sqAct1[nameHeadAct], as.numeric)

#visualizing transformed data
str(sqAct1)
```

```
## 'data.frame':    131 obs. of  8 variables:
##  $ Frames          : num  56 32 93 48 95 128 13 10 74 78 ...
##  $ Seconds         : num  9 116 47 2 49 81 97 94 27 31 ...
##  $ StrokePlayer    : num  1 2 2 2 2 2 1 1 2 1 ...
##  $ StrokeOutcome   : num  1 5 1 1 4 1 1 1 5 1 ...
##  $ ForehandBackhand: num  1 1 1 1 1 2 1 1 2 1 ...
##  $ StrokeX         : num  77 20 22 13 49 81 69 8 79 64 ...
##  $ StrokeY         : num  81 74 73 35 1 47 22 13 56 75 ...
##  $ StrokeType      : num  2 1 1 1 19 22 1 3 17 2 ...
```

```
head(sqAct1)
```

```
##     Frames Seconds StrokePlayer StrokeOutcome ForehandBackhand StrokeX
## 33      56       9            1             1                 1      77
## 128     32     116            2             5                 1      20
## 66      93      47            2             1                 1      22
## 27      48       2            2             1                 1      13
## 68      95      49            2             4                 1      49
## 98     128      81            2             1                 2      81
##     StrokeY StrokeType
## 33       81          2
## 128      74          1
## 66       73          1
## 27       35          1
## 68        1         19
## 98       47         22
```

```
#-------------------------------------------------------------------------#

#Data manipulation of player 2
sqAct2 <- sqAct2[2:132,]
head(sqAct2)
```

```
##    V1    V2 V3 V4 V5 V6  V7  V8
## 2  81  3.24  2  S  i  b 124 164
## 3 134  5.36  1 BO  i  b  30 103
## 4 170   6.8  2 DL  i  f 246 227
## 5 194  7.76  1 LL  n  f 231 246
## 6 357 14.28  2  S  i  f 109 118
## 7 407 16.28  1 LL  i  f  28 159
```

```
#changing header names for better understanding
colnames(sqAct2) <- nameHeadAct
sqAct2$StrokeType <- factor(sqAct2$StrokeType, levels =
c("S","LL","C","CS","DL","DC","LOL","LOC","KL","KC","VLL","VC","VCS","VDL","VDC","VBN","VBO","V
BR","VKL","VKC","BN","BO","BR","BS","COS"),labels = c("1","2","3","4","5","6","7","8","9","1
0","11","12","13","14","15","16","17","18","19","20","21","22","23","24","25"))

#Manipulate data by changing char to num
sqAct2$StrokeOutcome <- factor(sqAct2$StrokeOutcome, levels = c("i","l","s","t","n"), labels
= c("1","2","3","4","5"))

sqAct2$ForehandBackhand <- factor(sqAct2$ForehandBackhand, levels = c("f","b"), labels =
c("1","2"))

#moving 4th column to the last
sqAct2 <- sqAct2[, c(1,2,3,5,6,7,8,4)]

#randomizing the data
sqAct2 <- sqAct2[sample(nrow(sqAct2)),]
str(sqAct2)
```

```
## 'data.frame':    131 obs. of  8 variables:
##  $ Frames          : Factor w/ 299 levels "10014","1012",..: 150 169 192 134 164 218 207 1
44 146 199 ...
##  $ Seconds         : Factor w/ 299 levels "101.2","102.28",..: 18 38 60 2 32 86 75 12 14 6
7 ...
##  $ StrokePlayer    : Factor w/ 3 levels "1","2","EventPlayer": 2 1 2 1 2 2 1 2 2 1 ...
##  $ StrokeOutcome   : Factor w/ 5 levels "1","2","3","4",..: 1 4 1 1 1 1 1 1 1 1 ...
##  $ ForehandBackhand: Factor w/ 2 levels "1","2": 2 1 2 2 2 2 2 2 1 2 ...
##  $ StrokeX         : Factor w/ 120 levels "100","101","102",..: 55 116 88 3 85 73 69 114 9
1 79 ...
##  $ StrokeY         : Factor w/ 152 levels "10","101","103",..: 124 73 103 81 100 122 110 3
6 43 106 ...
##  $ StrokeType      : Factor w/ 25 levels "1","2","3","4",..: 3 9 2 2 2 22 2 3 3 2 ...
```

```
#making data numeric from factor
sqAct2[nameHeadAct] <- lapply(sqAct2[nameHeadAct], as.numeric)

#visualizing the data
str(sqAct2)
```

```
## 'data.frame':    131 obs. of  8 variables:
##  $ Frames          : num  150 169 192 134 164 218 207 144 146 199 ...
##  $ Seconds         : num  18 38 60 2 32 86 75 12 14 67 ...
##  $ StrokePlayer    : num  2 1 2 1 2 2 1 2 2 1 ...
##  $ StrokeOutcome   : num  1 4 1 1 1 1 1 1 1 1 ...
##  $ ForehandBackhand: num  2 1 2 2 2 2 2 2 1 2 ...
##  $ StrokeX         : num  55 116 88 3 85 73 69 114 91 79 ...
##  $ StrokeY         : num  124 73 103 81 100 122 110 36 43 106 ...
##  $ StrokeType      : num  3 9 2 2 2 22 2 3 3 2 ...
```

```
head(sqAct2)
```

```
##     Frames Seconds StrokePlayer StrokeOutcome ForehandBackhand StrokeX
## 68     150      18            2             1                2      55
## 85     169      38            1             4                1     116
## 104    192      60            2             1                2      88
## 52     134       2            1             1                2       3
## 80     164      32            2             1                2      85
## 130    218      86            2             1                2      73
##     StrokeY StrokeType
## 68      124          3
## 85       73          9
## 104     103          2
## 52       81          2
## 80      100          2
## 130     122         22
```

```
#-------------------------------------------------------------------------#

#player's position data manipulation
head(sqPos1)
```

```
##       V1      V2          V3          V4          V5          V6
## 1 Frames Seconds          X1          Y1       X1Cam       Y1Cam
## 2      1       0 6.101446152  0.54678216  269.551239 44.74259949
## 3      2    0.04 6.099555206 0.548891211 269.4902344 44.79514313
## 4      3    0.08 6.098546219 0.549808216 269.4571228 44.81723785
## 5      4    0.12 6.098580551 0.548982143 269.4560242 44.79377747
## 6      5    0.16 6.098722076 0.547036314 269.4556274 44.73877716
##            V7          V8          V9         V10   V11
## 1          X2          Y2       X2Cam       Y2Cam Phase
## 2 8.240261078 3.933374786 341.8302002 169.9903259     2
## 3 8.232774353 3.922166824 341.6695862 169.6081238     2
## 4 8.224371338 3.910958862 341.4848633 169.2265625     2
## 5 8.217028809 3.902433777  341.319519 168.9372101     2
## 6 8.212075806 3.898316574 341.2033386 168.7993317     2
```

```
sqPos1 <- sqPos1[2:13729,]

#changing header names
namePos <- c("Frames","Seconds","X1","Y1","X1Cam","Y1Cam","X2","Y2","X2Cam","Y2Cam","Phase")
colnames(sqPos1) <- namePos
head(sqPos1)
```

```
##   Frames Seconds          X1          Y1       X1Cam       Y1Cam
## 2      1       0 6.101446152  0.54678216  269.551239 44.74259949
## 3      2    0.04 6.099555206 0.548891211 269.4902344 44.79514313
## 4      3    0.08 6.098546219 0.549808216 269.4571228 44.81723785
## 5      4    0.12 6.098580551 0.548982143 269.4560242 44.79377747
## 6      5    0.16 6.098722076 0.547036314 269.4556274 44.73877716
## 7      6     0.2 6.097613907 0.545165682 269.4112244 44.68087769
##            X2          Y2       X2Cam       Y2Cam Phase
## 2 8.240261078 3.933374786 341.8302002 169.9903259     2
## 3 8.232774353 3.922166824 341.6695862 169.6081238     2
## 4 8.224371338 3.910958862 341.4848633 169.2265625     2
## 5 8.217028809 3.902433777  341.319519 168.9372101     2
## 6 8.212075806 3.898316574 341.2033386 168.7993317     2
## 7 8.209616852 3.899672699 341.1365051 168.8507233     2
```

```
#randomizing data
sqPos1 <- sqPos1[sample(nrow(sqPos1)),]
str(sqPos1)
```

```
## 'data.frame':    13728 obs. of  11 variables:
##  $ Frames : Factor w/ 13729 levels "1","10","100",..: 6577 4038 3439 6156 7219 3724 10621
 11305 11816 68 ...
##  $ Seconds: Factor w/ 13729 levels "0","0.04","0.08",..: 1236 12406 11817 807 1864 12099 5
250 5967 6477 8458 ...
##  $ X1     : Factor w/ 13646 levels "0.973207378",..: 3752 7402 13377 8578 6137 9900 6469 1
1468 8796 7414 ...
##  $ Y1     : Factor w/ 13659 levels "0.158134174",..: 10594 34 8255 5077 358 8655 7276 9118
8844 11781 ...
##  $ X1Cam  : Factor w/ 13635 levels "100.3951111",..: 4189 6669 13436 8646 5670 10057 6805
 11608 8821 7336 ...
##  $ Y1Cam  : Factor w/ 13653 levels "100.0035095",..: 7325 10091 4613 1553 10552 5024 3678
 5412 5235 8330 ...
##  $ X2     : Factor w/ 13696 levels "0.884523106",..: 7016 8334 7136 5635 11761 4706 8634 3
840 2182 30 ...
##  $ Y2     : Factor w/ 13708 levels "0.303972554",..: 7290 5016 7725 3184 10894 9593 11577
 1571 12117 13473 ...
##  $ X2Cam  : Factor w/ 13689 levels "100.1696396",..: 7054 8510 7177 5693 11619 4634 8464 3
751 2056 13652 ...
##  $ Y2Cam  : Factor w/ 13703 levels "100.0054398",..: 4798 2560 5258 571 8202 7493 8978 125
12 9959 10556 ...
##  $ Phase  : Factor w/ 3 levels "1","2","Phase": 1 2 2 1 2 1 2 2 1 2 ...
```

```
#making data to numeric data
sqPos1[namePos] <- lapply(sqPos1[namePos], as.numeric)

#visualize the data
str(sqPos1)
```

```
## 'data.frame':    13728 obs. of  11 variables:
##  $ Frames : num  6577 4038 3439 6156 7219 ...
##  $ Seconds: num  1236 12406 11817 807 1864 ...
##  $ X1     : num  3752 7402 13377 8578 6137 ...
##  $ Y1     : num  10594 34 8255 5077 358 ...
##  $ X1Cam  : num  4189 6669 13436 8646 5670 ...
##  $ Y1Cam  : num  7325 10091 4613 1553 10552 ...
##  $ X2     : num  7016 8334 7136 5635 11761 ...
##  $ Y2     : num  7290 5016 7725 3184 10894 ...
##  $ X2Cam  : num  7054 8510 7177 5693 11619 ...
##  $ Y2Cam  : num  4798 2560 5258 571 8202 ...
##  $ Phase  : num  1 2 2 1 2 1 2 2 1 2 ...
```

```
head(sqPos1)
```

```
##       Frames Seconds    X1    Y1 X1Cam Y1Cam    X2    Y2 X2Cam Y2Cam Phase
## 3562    6577    1236  3752 10594  4189  7325  7016  7290  7054  4798     1
## 13632   4038   12406  7402    34  6669 10091  8334  5016  8510  2560     2
## 13093   3439   11817 13377  8255 13436  4613  7136  7725  7177  5258     2
## 3183    6156     807  8578  5077  8646  1553  5635  3184  5693   571     1
## 4140    7219    1864  6137   358  5670 10552 11761 10894 11619  8202     2
## 13350   3724   12099  9900  8655 10057  5024  4706  9593  4634  7493     1
```

```
#-------------------------------------------------------------------------#

# 2nd player's Position
head(sqPos2)
```

```
##       V1      V2          V3          V4          V5          V6
## 1 Frames Seconds          X1          Y1       X1Cam       Y1Cam
## 2      1    0.04 7.887197876 4.141967773 49.86886597 88.49708557
## 3      2    0.08 7.887401581 4.142041779 49.86247253 88.49447632
## 4      3    0.12 7.887469482 4.142049027 49.86026001 88.49433136
## 5      4    0.16 7.887537384 4.142056274 49.85804749  88.4941864
## 6      5     0.2   7.8877388 4.142077637 49.85148621 88.49378204
##            V7          V8          V9         V10   V11
## 1          X2          Y2       X2Cam       Y2Cam Phase
## 2 5.591320419 0.736949158 142.1392212 245.5996399     2
## 3 5.601895523 0.673336077 141.7651672 247.9669189     2
## 4 5.613120651 0.616586828 141.3566132 250.0461578     2
## 5 5.619547272 0.579920292 141.1322632  251.380661     2
## 6 5.620265579 0.564735794 141.1262817 251.9388123     2
```

```
sqPos2 <- sqPos2[2:13729,]

#changing column names
namePos <- c("Frames","Seconds","X1","Y1","X1Cam","Y1Cam","X2","Y2","X2Cam","Y2Cam","Phase")
colnames(sqPos2) <- namePos
head(sqPos2)
```

```
##   Frames Seconds          X1          Y1       X1Cam       Y1Cam
## 2      1    0.04 7.887197876 4.141967773 49.86886597 88.49708557
## 3      2    0.08 7.887401581 4.142041779 49.86247253 88.49447632
## 4      3    0.12 7.887469482 4.142049027 49.86026001 88.49433136
## 5      4    0.16 7.887537384 4.142056274 49.85804749  88.4941864
## 6      5     0.2   7.8877388 4.142077637 49.85148621 88.49378204
## 7      6    0.24 7.888299561 4.142137528 49.83320618 88.49262238
##            X2          Y2       X2Cam       Y2Cam Phase
## 2 5.591320419 0.736949158 142.1392212 245.5996399     2
## 3 5.601895523 0.673336077 141.7651672 247.9669189     2
## 4 5.613120651 0.616586828 141.3566132 250.0461578     2
## 5 5.619547272 0.579920292 141.1322632  251.380661     2
## 6 5.620265579 0.564735794 141.1262817 251.9388123     2
## 7  5.61917305 0.557924795 141.1864624 252.1946869     2
```

```
#Randomizing data
sqPos2 <- sqPos2[sample(nrow(sqPos2)),]
str(sqPos2)
```

```
## 'data.frame':    13728 obs. of  11 variables:
##  $ Frames : Factor w/ 15509 levels "1","10","100",..: 12592 9996 1819 13300 3604 13939 904
1 10250 1526 13720 ...
##  $ Seconds: Factor w/ 15509 levels "0.04","0.08",..: 5448 2886 10184 6151 11965 6811 1902
 3140 9895 6589 ...
##  $ X1     : Factor w/ 15471 levels "2.745683289",..: 12251 11016 8100 11084 1388 4890 4578
2403 4999 12773 ...
##  $ Y1     : Factor w/ 15495 levels "0.144649911",..: 13274 15433 4903 736 535 3527 9622 13
510 4767 7892 ...
##  $ X1Cam  : Factor w/ 15495 levels "100.0288391",..: 8146 10622 11722 8893 3092 15023 25 2
445 14909 6670 ...
##  $ Y1Cam  : Factor w/ 15492 levels "100.0075607",..: 12752 10519 5514 9535 10016 7032 713
 12044 5731 2574 ...
##  $ X2     : Factor w/ 15467 levels "3.305606842",..: 10124 7717 4459 7559 10016 13800 1259
6 7260 12197 3807 ...
##  $ Y2     : Factor w/ 15499 levels "0.34586525","0.34597702",..: 8140 2980 3794 4964 11109
12148 10206 1688 13544 15308 ...
##  $ X2Cam  : Factor w/ 15497 levels "100.0242462",..: 9119 11407 15012 11649 10071 5729 703
2 12035 7947 817 ...
##  $ Y2Cam  : Factor w/ 15496 levels "100.0036392",..: 1297 6339 5657 4406 13790 13057 14789
7670 11402 9602 ...
##  $ Phase  : Factor w/ 3 levels "1","2","Phase": 1 2 1 1 2 1 1 1 1 2 ...
```

```
#transform data to numeric
sqPos2[namePos] <- lapply(sqPos2[namePos], as.numeric)

#visualizing the data
str(sqPos2)
```

```
## 'data.frame':    13728 obs. of  11 variables:
##  $ Frames : num  12592 9996 1819 13300 3604 ...
##  $ Seconds: num  5448 2886 10184 6151 11965 ...
##  $ X1     : num  12251 11016 8100 11084 1388 ...
##  $ Y1     : num  13274 15433 4903 736 535 ...
##  $ X1Cam  : num  8146 10622 11722 8893 3092 ...
##  $ Y1Cam  : num  12752 10519 5514 9535 10016 ...
##  $ X2     : num  10124 7717 4459 7559 10016 ...
##  $ Y2     : num  8140 2980 3794 4964 11109 ...
##  $ X2Cam  : num  9119 11407 15012 11649 10071 ...
##  $ Y2Cam  : num  1297 6339 5657 4406 13790 ...
##  $ Phase  : num  1 2 1 1 2 1 1 1 1 2 ...
```

```
head(sqPos2)
```

```
##       Frames Seconds    X1    Y1 X1Cam Y1Cam    X2    Y2 X2Cam Y2Cam Phase
## 7374   12592    5448 12251 13274  8146 12752 10124  8140  9119  1297     1
## 5037    9996    2886 11016 15433 10622 10519  7717  2980 11407  6339     2
## 11635   1819   10184  8100  4903 11722  5514  4459  3794 15012  5657     1
## 802    13300    6151 11084   736  8893  9535  7559  4964 11649  4406     1
## 13241   3604   11965  1388   535  3092 10016 10016 11109 10071 13790     2
## 8587   13939    6811  4890  3527 15023  7032 13800 12148  5729 13057     1
```

```
#-------------------------------------------------------------------------#

#loading data to some another variables to be implemented in algorithm
Act1 <- sqAct1
Act2 <- sqAct1

#only 1500 rows are taken, this is done to make processing faster if whole data is used at a
 time them it will take approximately 2 or 3 hours to process
Pos1 <- sqPos1[1:1500,]
Pos2 <- sqPos2[1:1500,]


#gausspr is a function in kernlab library this function is used implement gaussian process.
# Here data and the field on which process is implementing are provided to the function.
gp1 <- gausspr(StrokeType~.,data=Act1,var=0.1)
```

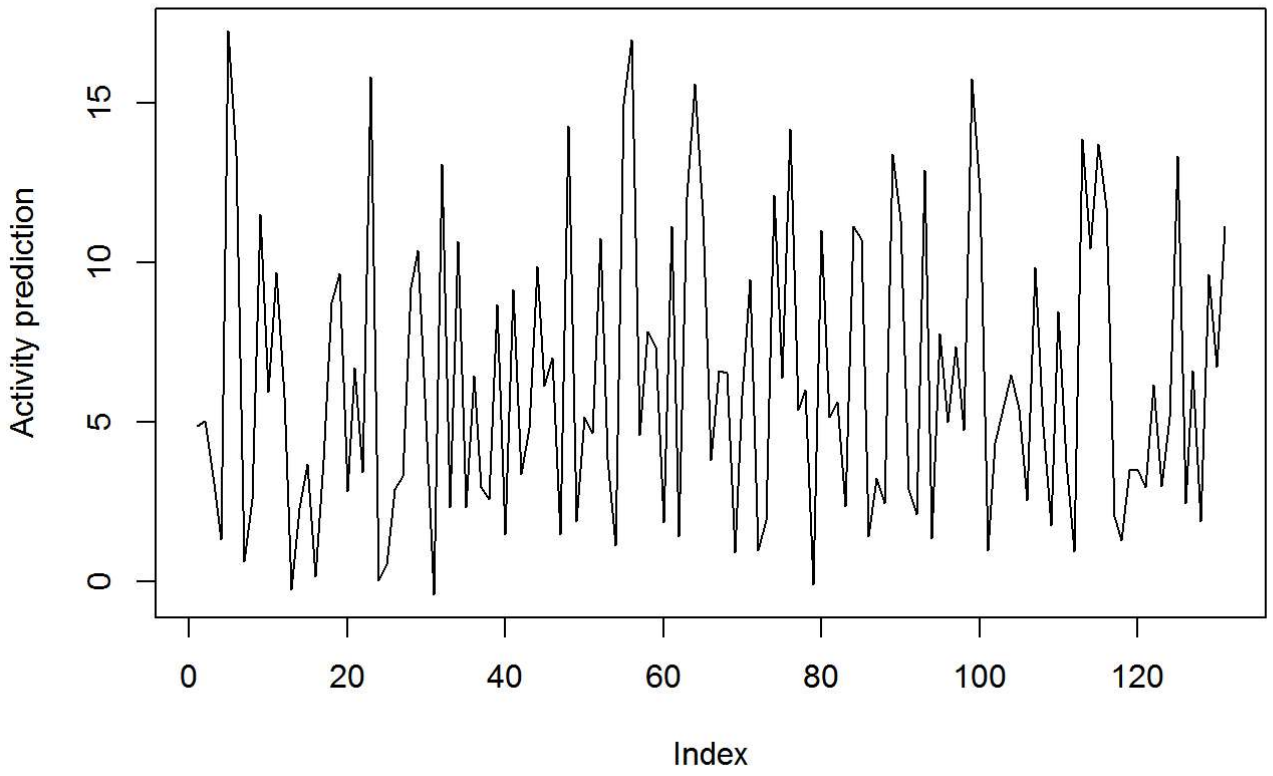
```
## Using automatic sigma estimation (sigest) for RBF or laplace kernel
```

```
# printing the model which is processed
gp1
```

```
## Gaussian Processes object of class "gausspr"
## Problem type: regression
##
## Gaussian Radial Basis kernel function.
##  Hyperparameter : sigma =  0.108704432441809
##
## Number of training instances learned : 131
## Train error : 0.227961878
```

```
# predicting the gp with the dataset
pred1 <- predict(gp1,Act1[,-8])

#plot the predicted result
plot(pred1, type="l", ylab="Activity prediction")
```




```
#Gaussian process is implemented on the 2nd  player activity data
gp2 <- gausspr(StrokeType~.,data=Act2,var=0.1)
```

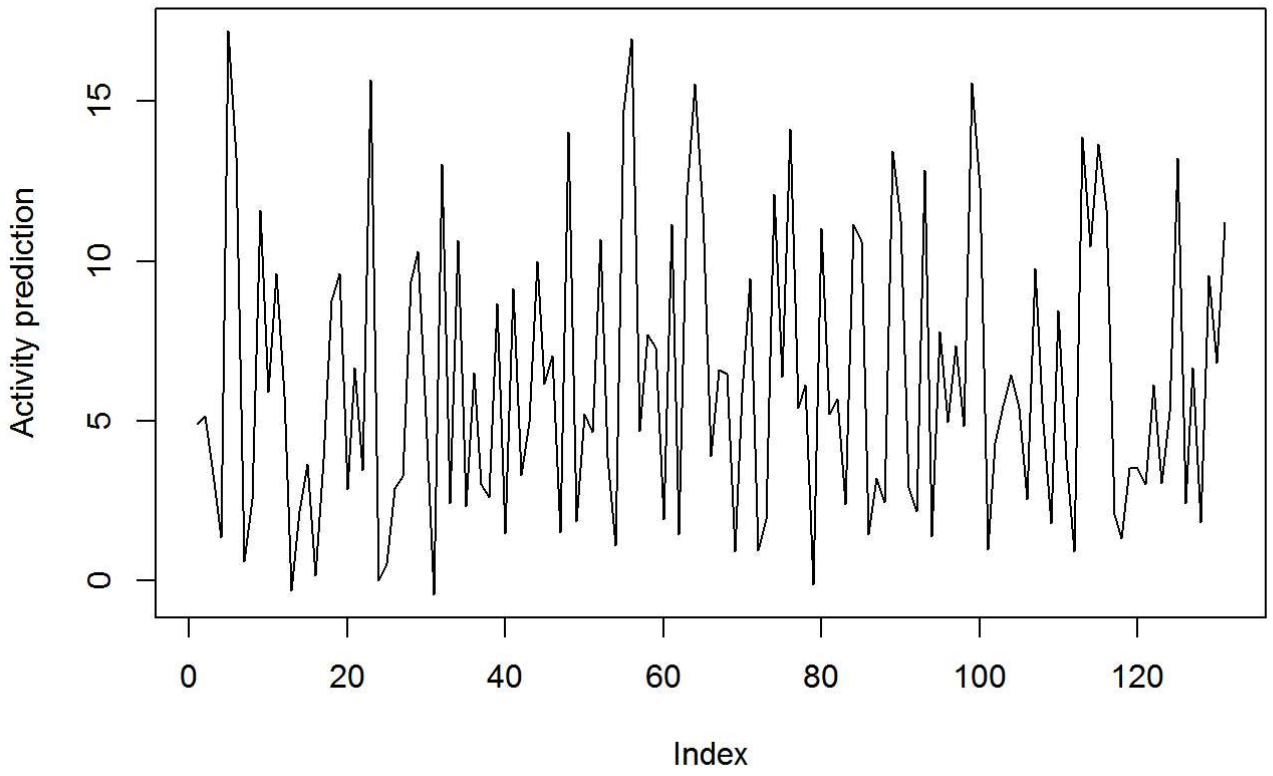
```
## Using automatic sigma estimation (sigest) for RBF or laplace kernel
```

```
gp2
```

```
## Gaussian Processes object of class "gausspr"
## Problem type: regression
##
## Gaussian Radial Basis kernel function.
##  Hyperparameter : sigma =  0.104514449210775
##
## Number of training instances learned : 131
## Train error : 0.235918451
```

```
# predicting the gp with data
pred2 <- predict(gp2,Act2[,-8])

#ploting the predicted result
plot(pred2, type="l", ylab="Activity prediction")
```




```
#-------------------------------------------------------------------------------#


#Gaussian process on the position data of player 1
gp3 <- gausspr(Phase~.,data=Pos1,var=0.1)
```

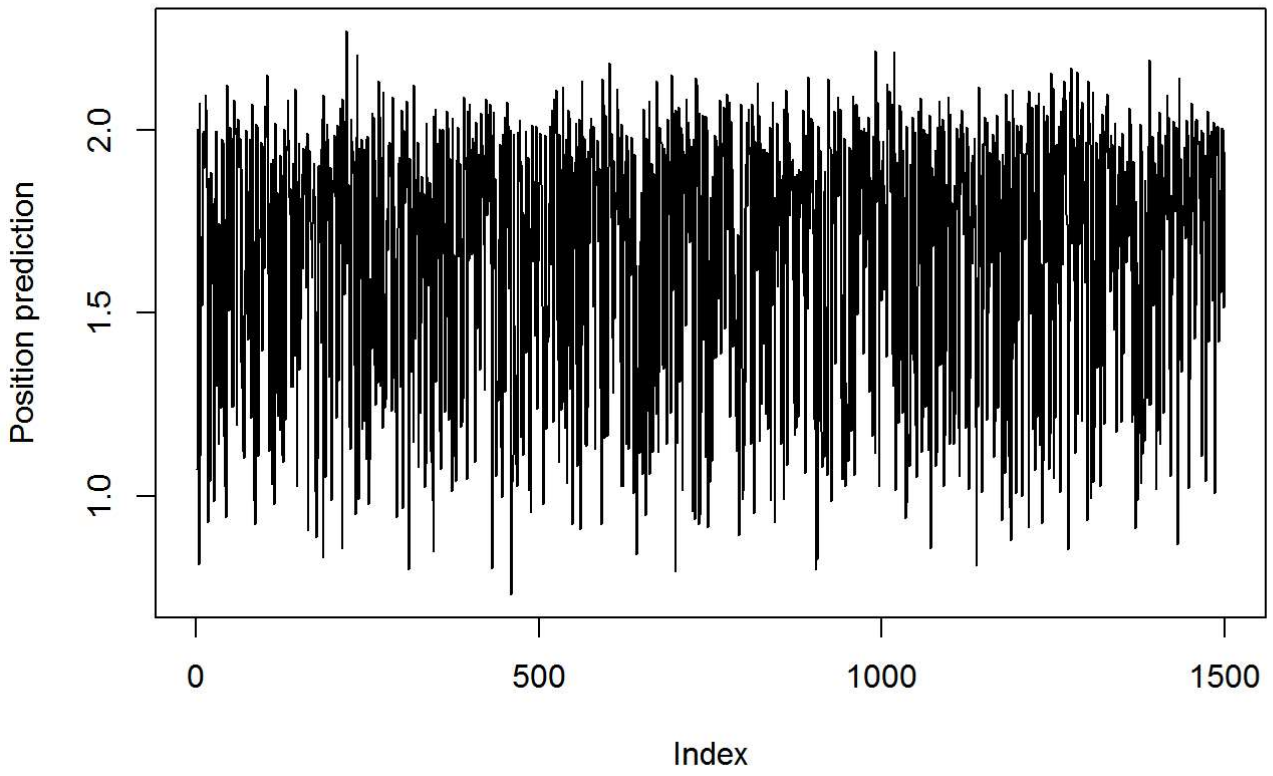
```
## Using automatic sigma estimation (sigest) for RBF or laplace kernel
```

```
gp3
```

```
## Gaussian Processes object of class "gausspr"
## Problem type: regression
##
## Gaussian Radial Basis kernel function.
##  Hyperparameter : sigma =  0.0725094420733168
##
## Number of training instances learned : 1500
## Train error : 0.299420752
```

```
# predicting the gp with position
pred3 <- predict(gp3,Pos1[,-11])

#ploting the result
plot(pred3, type="l", ylab="Position prediction")
```




```
#Gaussian process on 2nd player position data
gp4 <- gausspr(Phase~.,data=Pos2,var=0.1)
```

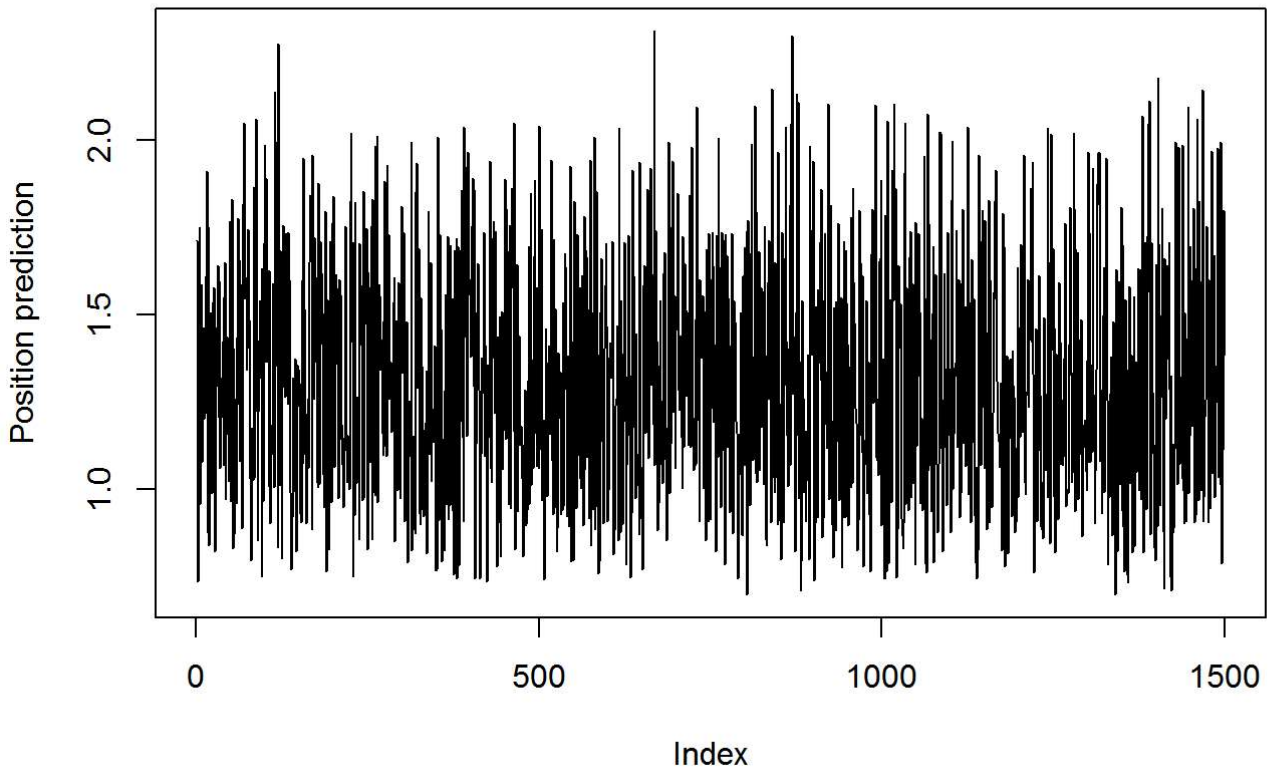
```
## Using automatic sigma estimation (sigest) for RBF or laplace kernel
```

```
gp4
```

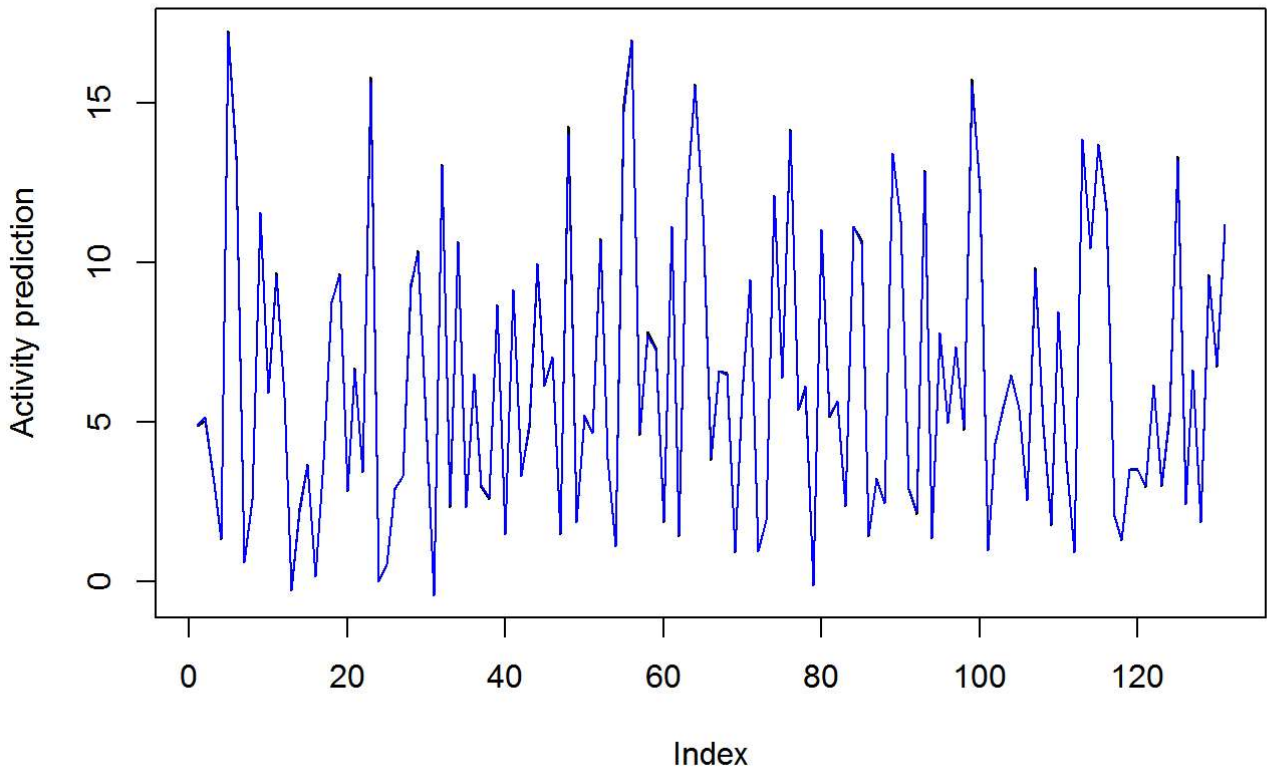
```
## Gaussian Processes object of class "gausspr"
## Problem type: regression
##
## Gaussian Radial Basis kernel function.
##  Hyperparameter : sigma =  0.0684175490528396
##
## Number of training instances learned : 1500
## Train error : 0.171100667
```

```
# predicting the gp
pred4 <- predict(gp4,Pos1[,-11])

#ploting the result
plot(pred4, type="l", ylab="Position prediction")
```



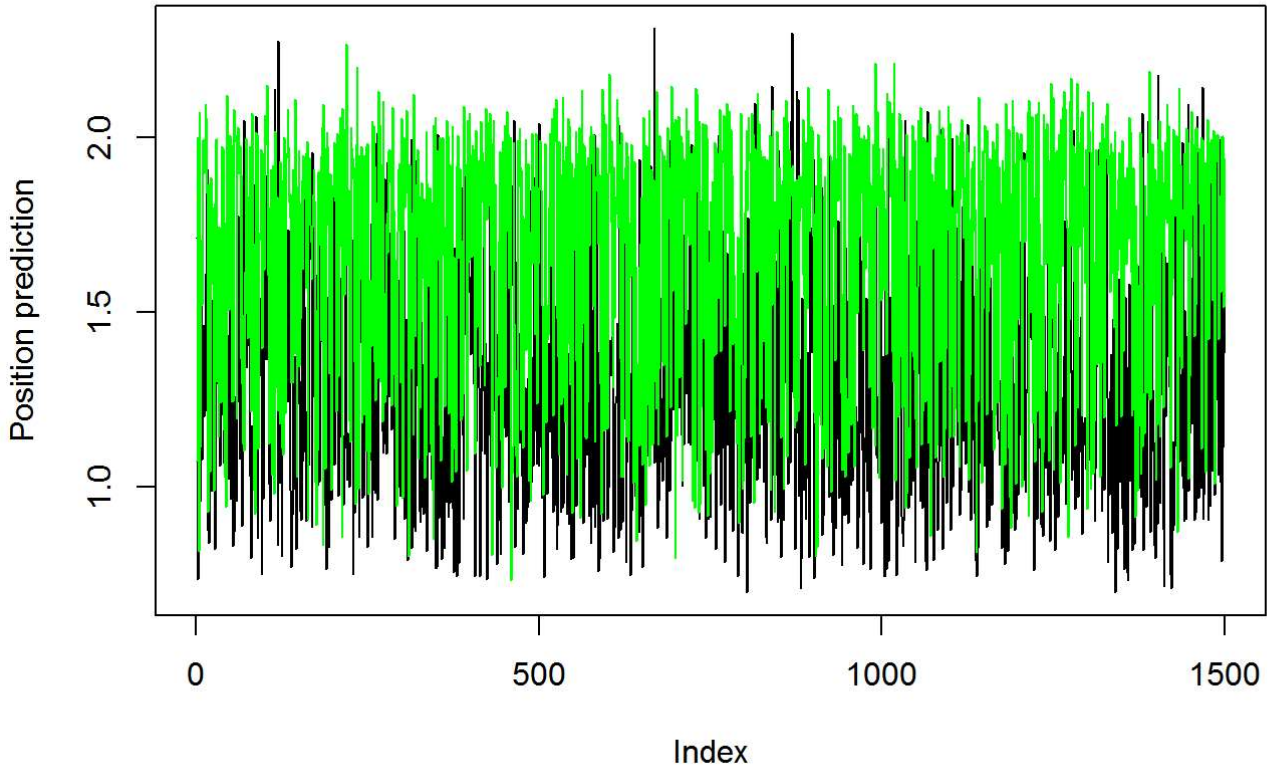

```
#=============================================================================#
#analyzing the activities of both players using plots
plot(pred1, type="l", ylab="Activity prediction")
lines(pred2, type="l", ylab="Activity prediction", col= "blue")
```




```
### Player 1 has few more types of shots during the game than te other player, as it clearly
 seen on from the plot that black lines are going higher.

#-------------------------------------------------------------------------------#

#Analyzing the positions changed by players during the game
plot(pred4, type="l", ylab="Position prediction")
lines(pred3, type = "l", col= "green")
```



Player 2 is occupying more area of the court by frequently moving in the court as compare d with the 1st player. This shows that 1st player is more active during the game.