KBO2021NewStat

Seung Hun Han

2021 12 1

Creating New Statistics for evaluating Hitter's Performance using Multivariate Statistical Methods

```
library(tidyverse)
library(rvest)
library(rJava)
library(XML)
library(dplyr)
library(plyr)
library(ggplot2)
library(httr)
library(RSelenium)
library(wdman)
library(binman) #list_versions()
library(stringr)
library(seleniumPipes)
library(wordcloud2)
library(wordcloud)
library(RColorBrewer)
library(knitr)
Batraw<-read.csv("2021bat.csv")</pre>
which(is.na(Batraw))
## [1] 7103 7104
names (Batraw)
   [1] "Name"
                                                       "WAR"
##
                               "Team"
## [4] "G"
                               "PA"
                                                       "AB"
## [7] "RUN"
                               "HIT"
                                                      "X2B"
                               "HR"
                                                      "TB"
## [10] "X3B"
## [13] "RBI"
                               "BB"
                                                       "HBP"
                                                       "DP"
## [16] "IB"
                               "SO"
## [19] "SAC"
                               "SAC2"
                                                      "BA"
## [22] "OBP"
                               "SLG"
                                                      "OPS"
## [25] "wOBA"
                               "wRCP"
                                                      "WPA"
```

```
## [28] "IsoP"
                                 "IsoD"
                                                          "wRC27"
   [31] "wRAA"
                                 "RAABat"
##
                                                          "RAAdef"
   [34] "RAADefPos"
                                 "RAAdeftotal"
                                                          "RAA.Adj"
                                 "Runper"
                                                          "Advancedpossibility"
  [37] "Advancedbase"
                                 "SB"
                                                          "CS"
   [40] "Outonbase"
## [43] "RAASB"
                                 "RAABR"
                                                          "Spd"
Batraw<-Batraw%>%filter(AB>=100)
Batraw<-as_tibble(Batraw)</pre>
Hittingraw \leftarrow Batraw [, c(1,4:32)]
Defenseraw <- Batraw [,33:35]
Runnerraw <- Batraw [, 37:44]
```

- Column 4~32 Batting
- Column 33~35 Def
- Column 37~44 Runner

```
Runnerraw[,"CS"] <--Runnerraw[,"CS"]
Hittingraw[,c("S0","DP")] <--Hittingraw[,c("S0","DP")]</pre>
```

• Take negative to column 40 and 41

```
Hittingraw<-Hittingraw%>%mutate(SOPer=-SO/AB,DPPer=-DP/AB)
Standardhit<-scale(Hittingraw[,-1],center=T)</pre>
```

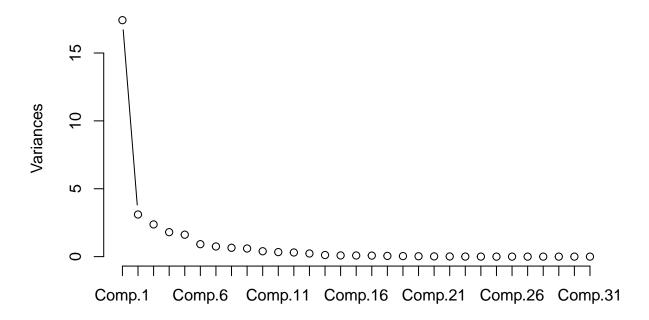
- Large number of SO and DP are assumed to have negative effect on player's value.
- However, SO and DP are accumulated statistics, so considering ratio value would compensate for the disadvantages for the sluggers.
- Though one might assert that only the ratio statistics such as AVG, SLG, OPS should be considered for the analysis, I have decided not to remove accumulative statistics such as Hits, RBIs and HRs.
- Being able to be in the starting lineup throughout whole season is also a valuable aspect. Therefore, accumulative statistics should be as highly valued as the ratio statistics.
- The reason mentioned above is the reason why I have decided to include G (Number of games played).
- Because variables included in the datasets do not have same unit, it would be wise to standardize each values relative to the mean value of an entire players.
- Because of the standardization, Newly derived statistics from an analysis which will be conducted later, would be a "relative" scales. To be precise, the statistics will show how much better or worse a player played in 2021 season. Therefore, work done here can not be replicated for other seasons.

```
Batcor<-cor(Standardhit)
kable(Batcor)</pre>
```

G 1.000000000000000000000000000000000000
PAO.900000000TTTD86G9980T0909646FTTT0000330065 0.00007568850T38028006T20088040FTT000040817338T0147111 ABO.8969900000000T03003669285044835428589141 0.07863522869862180303727040704126853985202120245 RIOS.6998236000000000000000000000000000000000000
ABD.839699000090076389500009054544825348383583141
ADD.896899000000001640936936448354108354161 0.65861521605018081031310794929041296295982201120245 0.66804678528453 0.0260515 0.0683477 0.0463477 0.0463477 0.6463477 0.6463477 0.6463477 0.6463477 0.6463477 0.6463477 0.6463477 0.6463477 0.6463477 0.6463477 0.6463477 0.6463477 0.64664902 0.5180930365 0.6466980238124 0.0664902 0.5180930365 0.6466980238124 0.0664902 0.5180930365 0.6466980238124 0.0664902 0.5180930365 0.64669802 0.6466980238124 0.0664902 0.5180930365 0.64669802 0.6466802 0.646680
Comparison
RUDS-869982300000000000000000000000000000000000
HTD.83599017039000000018360003181813053230224 0.095685418189206870986581870209832207 0.0956685093124 0.0064902 0.51860930365 0.518609305632779189387975 0.1980689063 0.1460806 0.23837397375 0.1980689063 0.5386093656188986986960000000000000000000000000000
HTD.8359%0945029000000183850500331818305922624
Comparison
X2B.74F3dD96E9BB3GD00000000000000000000000000000000000
Comparison
X3B.3 ID 39840#09826536906000000000000000000000000000000000
RRO.499\$1963583959001992800009280000928052743
RB0.49954963598348E99082990000006559645859698952743 -
TB0.82093B931P93T0G093T97EQG0QGDGB0B96B94838937255
TB0.8299393184170693175704090400408089454512588888889828393040907530318685368723325676644756280040021150 RB0.732868666828567231892898300000408098724-0.059837888366979 0.3028312 BB0.769877863D3328682029015669409447060000098724-0.0.6183789084689839483004093895341009080698724-0.0.6183708066680 0.3361271 HBP.4454606286488996838071846426937935250008000098724-0.3365383389300000000000000000000000000000000
RBQ.73286666821693266729982268330000400966533649
RBD.73D28G56G4321693267213982B263G0000000565739049 - 0.73D31D3B23T76UBU31D26J000001101E306540048(8)66479584
BB0.76087116108312868702901566940847040000000087294
BB0.76987851933286870290150590B47090800098724-0.61857066680 0.3510271 HBD:44546962868597688671260226137335250989070399-0.3363353\$393546393390924837528899200493560548536 0.481.3207338278 0.035325395346393390924837528899200493560548536 0.39766508712 0.0448895478336539369473445983395318527720955883 B 0.438538905238276092092042050386395089727909000-0.4488954763356732369473445983395318527720955883 B 0.43853890523827609204204503863950847270909000-0.0448895476333653932689403445983395318527720955883 B 0.438538905238276386906082388689276698 0.039766709386823123237136492186869404399906617 DP 0 0.226600900000-66113-0 0 0.4495861 0.580028686383979390668923886397638936030 0.41830339313945293130492186869404399906617 DP 0 0.926600900000-6767-0 - 0 0.0343913-0 - 0.4495861 0.580028686383979390689283839638960889283712 0.09960176000000-0 0 0.0343913-0 - 0.4495861 0.5800286863839709390689283836388698688283712 0.09960176000000-0 0 0.0343913-0 - 0.0495807 SAC2996886786786786786786785889898688283712 0.09960176000000-0 0 0.0343913-0 - 0.0495807 SAC299688678678678678678586980688283712 0.09960176000000-0 0 0.0343913-0 - 0.0495807 SAC29968867867867867867858698678588977- 0.099601786000000-0 0 0.0495407 0.3649679 BAO.5296886786786786986785867878390723- 0.64250000000887989283908888898989888889898988888989898888898998888
Color Colo
HBP:445469628485953830716612052537525008070399
0.443558890638831509264805088315098772090900
SO
SO 1.00000000066113 0.01236967 0.64066939035060665398690000032838383970698
0.640640330358605668338693050903288383977698
DP-
0.580618688838307330648918330648918340648336003 0.4113603369313467091183803869306283982055326 SAC.1750000745 - 0.0852506 - 0 0 0.096617167607000 - 0 0 0 0.0343913 - 0 0.01024805881246733865088898060688258712 0.01024863861246733865088898060688258712 0.4253011365685
\$SAC:1750\corrected{9000000000000000000000000000000000000
$\begin{array}{cccccccccccccccccccccccccccccccccccc$
\$\text{SAC29P68B6786726780376783677836778363783383777} \ 1.000000330090665768445701378393478420562180053435 \ 0.4255012365685 \ 0.0495407 \ 0.3649679\$ \$\text{BA0.52960862973767367367367367678366723} \ 0.20556502929901 \ 0.1696665 \ 0.6000036111\$ \$\text{OBP:50B58B23B09904B0884930000057353556350452367} \ 0.26139502674394 \ 0.430990000000888008599845938848860903840407752\$ \$\text{C.4456356048363696296300000888608599845538848860903840407752} \ 0.4308668966900000888008599845938848860903840407752\$ \$\text{OP6.51767869B669069096988596900000888008599845938848860903840407752} \ 0.4308966900000888000599865938848860903840407752\$ \$\text{OP6.51767869B66906909698859853685000900488839329372} \ 0.5667688128373600000088500000000000000000000000000000
0.42550H193608611043804T643369310638565120763906723
BA0.5206086120380816433653206338060723 - 0.5032000000082083183104581603806268150368793 0.20536502207901 0.1696665 0.6000056111 OBDP.50854B236097041804881B3000076737531563764512367 - 0.43090000000838538680730076309417630
$\begin{array}{cccccccccccccccccccccccccccccccccccc$
$\begin{array}{r} 0.45 \textbf{B39172709}275 & 0.3120377 \\ \text{wOB521763062767420509321702187085386306007904682} - & 0.548 \textbf{82849908659650090057146302365009905909204605}10963 \\ 0.4312 \textbf{92029858433} & 0.3485601 \\ \text{wRCF217630763063839420147C17210607900833209473} - & 0.547 \textbf{50483093099109900070050999072509097480875}32008 \\ 0.44 \textbf{924103879}118 & 0.3406279 \\ \text{WRM48060151765060563823060073072266090094764445} - & 0.6016485 60573070610742560900000000000000000000000000000000000$
$\frac{0.5488288998659650990057163923659996924466510963}{0.4312029858433} = \frac{0.3485601}{0.3485601} \\ \text{wRCF2N68K628639420147CF22086D9083329473} - 0.5478648398999999999999999999999999999999999$
$0.4312029358433 \\ 0.3485601 \\ \text{wRCF27685635637070002063942015471722066590089409} \\ 0.4424402899118 \\ 0.3406279 \\ \text{WRM4806015977653066668162370607394722660906047704445} - 0.60164876537066074256990090073969025098523997 \\ \end{array}$
$\frac{\text{wRCF276856387000636820420147123066790893209}473-0.547864881085099199900070097699912899099758087532008}{0.44224102899118} 0.3406279 \\ \text{WRM480601917653066668123060073072866926094704}445-0.6001648668723076607425699000002989025698823997}$
$0.44\textbf{2}4\textbf{1}\textbf{1}\textbf{2}\textbf{3}\textbf{5}\textbf{9}118 \\ \textbf{WRM} \textbf{3}0\textbf{6}\textbf{0}\textbf{5}\textbf{1}\textbf{7}\textbf{6}\textbf{5}\textbf{2}\textbf{6}\textbf{3}\textbf{5}\textbf{3}\textbf{2}\textbf{3}\textbf{0}\textbf{0}\textbf{7}\textbf{2}\textbf{3}\textbf{2}\textbf{2}\textbf{6}\textbf{0}\textbf{2}\textbf{0}\textbf{3}\textbf{4}\textbf{4}\textbf{4}\textbf{5} - 0.60\textbf{1}\textbf{6}\textbf{3}\textbf{5}\textbf{6}\textbf{5}\textbf{3}\textbf{5}\textbf{3}\textbf{7}\textbf{0}\textbf{0}\textbf{0}\textbf{0}\textbf{0}\textbf{0}\textbf{3}\textbf{2}\textbf{3}\textbf{0}\textbf{2}\textbf{0}\textbf{3}\textbf{5}\textbf{3}\textbf{3}\textbf{2}\textbf{3}\textbf{9}\textbf{7}$
WRA48080 LIST 17650 8656 8853 760 9730 7226 609469 4764 445 - 0.6016485 7853 78640 72569 7000 9000 9000 9000 9000 9500 8523997
0.3 7125626363 015 0.298 303 0477
IsoP.2003009789390480580018683456769878204836081 - 0.377308408589837899779977000000385795804917298956
0.5042088696341
Iso D .03007207996321547 - 0.07904001530509053920307 0.0343913 0.430142664230300127029660070004868593509072563
$0.0260 \$ 1666 490 946 980 6284981 \qquad 0.12 263 341 204 9549 666 6189515$

```
G PA ABRUNITX2BX3BHRTBRBBBHBB SO DPSAGAGAGAOBBLQPSODRAMPIAoBSOD/RG27RASGBBPPPer
wRC2746696496395086792719787073163999938492178 - 0.54782997538978999936409793244009907928782059366
                                       0.40536028828780
                                                                                    0.3550175
wR0A52760767782709898218645768540645607952617 - 0.5478598392699999999999999999997777527300
                                       0.4414698893976
                                                                                    0.3418433
RAAB0168B0907076083007387907406768215621720 - 0.566726370328909014068023904867219012070900 \\
                                       0.4299496928460
                                                                                    0.2400036274
             - - 0.0064220 -
                                          0.4495861 -
                                                                  - 0.19B2398102 -
                                                                                    1.0000000
   0.4293\mathbf{607653655851533894539650P9310252569290617512656070558483172385606959030\ 0.3550485462082014900
0.0940649365749727
                                          0.7056326750016111
                                                                   0.1170477
                                                                                 0.0120274900
summary(princomp(covmat=Batcor,cor=T))
## Importance of components:
##
                            Comp.1
                                      Comp.2
                                                 Comp.3
                                                            Comp.4
## Standard deviation
                         4.1735420 1.7613303 1.53991357 1.34217665 1.2714742
## Proportion of Variance 0.5618856 0.1000737 0.07649464 0.05811091 0.0521499
## Cumulative Proportion 0.5618856 0.6619593 0.73845391 0.79656482 0.8487147
##
                             Comp.6
                                        Comp.7
                                                   Comp.8
                                                              Comp.9
## Standard deviation
                         0.95690204 0.86541193 0.80695756 0.77336516 0.62963314
## Proportion of Variance 0.02953747 0.02415928 0.02100582 0.01929334 0.01278832
## Cumulative Proportion 0.87825219 0.90241147 0.92341729 0.94271064 0.95549896
                            Comp.11
                                        Comp.12
                                                    Comp.13
                                                                Comp.14
## Standard deviation
                         0.57765799 0.552547874 0.479226857 0.343250339
## Proportion of Variance 0.01076415 0.009848682 0.007408335 0.003800671
## Cumulative Proportion 0.96626311 0.976111792 0.983520127 0.987320798
##
                             Comp.15
                                         Comp.16
                                                     Comp.17
                                                                Comp.18
## Standard deviation
                         0.294401737 0.285845664 0.274621711 0.229127197
## Proportion of Variance 0.002795883 0.002635734 0.002432809 0.001693525
## Cumulative Proportion 0.990116681 0.992752415 0.995185224 0.996878749
                                          Comp.20
                             Comp.19
                                                       Comp.21
## Standard deviation
                         0.194270097 0.1659686756 0.1317992021 0.1011618754
## Proportion of Variance 0.001217447 0.0008885678 0.0005603558 0.0003301202
## Cumulative Proportion 0.998096196 0.9989847639 0.9995451197 0.9998752399
##
                                           Comp.24
                              Comp.23
                                                        Comp.25
## Standard deviation
                         5.421907e-02 2.700284e-02 9.842147e-03 6.748528e-03
## Proportion of Variance 9.482926e-05 2.352107e-05 3.124769e-06 1.469117e-06
## Cumulative Proportion 9.999701e-01 9.999936e-01 9.999967e-01 9.999982e-01
##
                              Comp.27
                                           Comp.28
                                                        Comp.29
                                                                     Comp.30
## Standard deviation
                         4.790836e-03 4.402058e-03 3.720404e-03 3.490557e-04
## Proportion of Variance 7.403906e-07 6.251004e-07 4.464971e-07 3.930318e-09
## Cumulative Proportion 9.999989e-01 9.999995e-01 1.000000e+00 1.000000e+00
##
                              Comp.31
## Standard deviation
                         1.230455e-08
## Proportion of Variance 4.883937e-18
## Cumulative Proportion 1.000000e+00
screeplot(princomp(covmat=Batcor,cor=T),npcs=31,type='1')
```

princomp(covmat = Batcor, cor = T)



- Variance explained by PC1,2 is 66.2%.
- Scree plot also has a distinctive breaking point at PC=2.
- Therefore, conserve up to PC2.

```
PCBat<-princomp(covmat=Batcor,cor=T)

PCBatload<-PCBat$loadings[,1:2]

PCBat1weight<-PCBat$sdev[1]^2/length(PCBat$sdev)

PCBat2weight<-PCBat$sdev[2]^2/length(PCBat$sdev)

PCBat1<-data.frame(PCB1=apply(Standardhit%*%PCBatload[,1],1,sum))

PCBat2<-PCBat2weight/PCBat1weight*data.frame(PCB2=apply(Standardhit%*%PCBatload[,2],1,sum))</pre>
```

- Retain PC scores through multiplication of standardized Batting data and each PC loadings.
- Use relative proportion of variance explained by each PC as a weight (Put PC2 proportion as default).

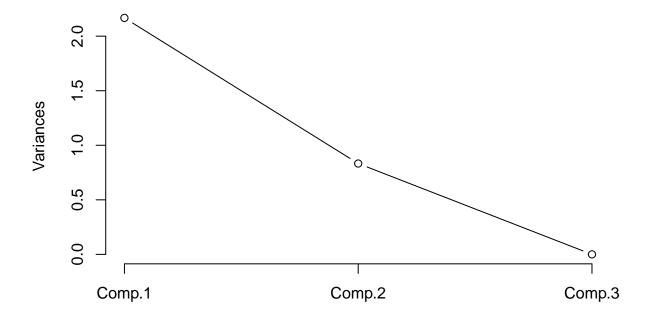
```
StandardDef<-scale(Defenseraw,center=T)
Defensecor<-cor(StandardDef)
summary(princomp(covmat=Defensecor,cor=T))

## Importance of components:
## Comp.1 Comp.2 Comp.3</pre>
```

```
## Standard deviation    1.4721604 0.9125326 5.304924e-03
## Proportion of Variance 0.7224187 0.2775719 9.380738e-06
## Cumulative Proportion 0.7224187 0.9999906 1.000000e+00

screeplot(princomp(covmat=Defensecor,cor=T),npcs=3,type='l')
```

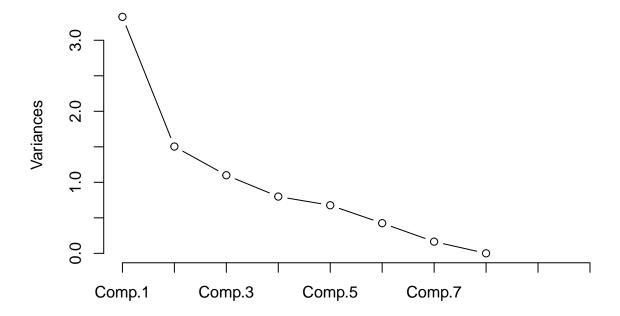
princomp(covmat = Defensecor, cor = T)



```
PCDef<-princomp(covmat=Defensecor,cor=T)</pre>
PCDefload<-PCDef$loadings[,1]</pre>
PCDef1<-data.frame(PCD1=apply(StandardDef%*%PCDefload,1,sum))</pre>
Runnerraw[which(is.na(Runnerraw)),1]<-0</pre>
StandardRun<-scale(Runnerraw,center=T)</pre>
Runningcor<-cor(StandardRun)</pre>
summary(princomp(covmat=Runningcor,cor=T))
## Importance of components:
##
                                         Comp.2
                                                    Comp.3
                              Comp.1
                                                               Comp.4
                                                                           Comp.5
## Standard deviation
                           1.8246801 1.2262488 1.0487739 0.89412179 0.82279488
## Proportion of Variance 0.4161822 0.1879608 0.1374908 0.09993172 0.08462393
## Cumulative Proportion 0.4161822 0.6041429 0.7416338 0.84156549 0.92618942
##
                               Comp.6
                                           Comp.7
                                                         Comp.8
## Standard deviation
                           0.65198113 0.40606529 2.272100e-02
## Proportion of Variance 0.05313492 0.02061113 6.453047e-05
```

```
screeplot(princomp(covmat=Runningcor,cor=T),npcs=10,type='1')
```

princomp(covmat = Runningcor, cor = T)



```
PCRun<-princomp(covmat=Runningcor,cor=T)

PCRunload<-PCRun$loadings[,1:3]

PCRun1weight<-PCRun$sdev[1]^2/length(PCRun$sdev)

PCRun2weight<-PCRun$sdev[2]^2/length(PCRun$sdev)

PCRun3weight<-PCRun$sdev[3]^2/length(PCRun$sdev)

PCRun3weight<-PCRun$sdev[3]^2/length(PCRun$sdev)

PCRun1<-data.frame(PCR1=apply(StandardRun%*%PCRunload[,1],1,sum))

PCRun2<-PCRun2weight/PCRun1weight*data.frame(PCR2=apply(StandardRun%*%PCRunload[,2],1,sum))

PCRun3<-PCRun3weight/PCRun1weight*data.frame(PCR3=apply(StandardRun%*%PCRunload[,3],1,sum))

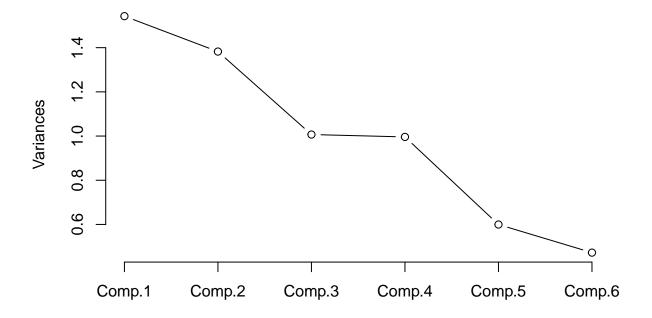
PCcombined<-data.frame(cbind(Batraw[,c("Name", "Team")],PCBat1,PCBat2,PCDef1,PCRun1,PCRun2,PCRun3))

PCcombined<-PCcombined%>%mutate(Total=apply(PCcombined[,3:7],1,sum))
```

• Repeat same process for Running and Defense dataset.

```
library(psych)
##
            : 'psych'
##
  The following objects are masked from 'package:ggplot2':
##
##
       %+%, alpha
##
library(GPArotation)
combinedcor<-cor(PCcombined[,3:8])</pre>
summary(princomp(covmat=combinedcor,cor=T))
## Importance of components:
##
                              Comp.1
                                        Comp.2
                                                  Comp.3
                                                            Comp.4
## Standard deviation
                          1.2419886 1.1757473 1.0034156 0.9980461 0.77438001
## Proportion of Variance 0.2570893 0.2303969 0.1678072 0.1660160 0.09994407
## Cumulative Proportion 0.2570893 0.4874862 0.6552934 0.8213094 0.92125346
##
                               Comp.6
## Standard deviation
                           0.68737126
## Proportion of Variance 0.07874654
## Cumulative Proportion 1.00000000
screeplot(princomp(covmat=combinedcor,cor=T),npcs=6,type='1')
```

princomp(covmat = combinedcor, cor = T)



```
## Factor Analysis using method = ml
## Call: fa(r = combinedcor, nfactors = 3, rotate = "oblimin", scores = "regression",
## Standardized loadings (pattern matrix) based upon correlation matrix
         ML2
               ML1
                     ML3
                              h2
                                   u2 com
## PCB1 0.99 -0.01 -0.01 0.9950 0.005 1.0
## PCB2 -0.05 -0.01 0.86 0.7316 0.268 1.0
## PCD1 -0.01 1.00 0.00 0.9950 0.005 1.0
## PCR1 0.22 0.09 0.50 0.3311 0.669 1.4
## PCR2 -0.07 -0.02 0.01 0.0043 0.996 1.2
## PCR3 -0.11 -0.07 0.06 0.0123 0.988 2.4
##
##
                         ML2 ML1 ML3
## SS loadings
                        1.06 1.01 1.01
## Proportion Var
                        0.18 0.17 0.17
## Cumulative Var
                        0.18 0.34 0.51
## Proportion Explained 0.34 0.33 0.33
## Cumulative Proportion 0.34 0.67 1.00
##
##
   With factor correlations of
##
        ML2
              ML1 ML3
## ML2 1.00 -0.34 0.06
## ML1 -0.34 1.00 0.23
## ML3 0.06 0.23 1.00
##
## Mean item complexity = 1.3
## Test of the hypothesis that 3 factors are sufficient.
## The degrees of freedom for the null model are 15 and the objective function was 0.5
## The degrees of freedom for the model are 0 and the objective function was 0
## The root mean square of the residuals (RMSR) is 0
## The df corrected root mean square of the residuals is \, NA
##
## Fit based upon off diagonal values = 1
## Measures of factor score adequacy
                                                     ML2 ML1 ML3
## Correlation of (regression) scores with factors
                                                     1.00 1.00 0.87
## Multiple R square of scores with factors
                                                     0.99 0.99 0.76
## Minimum correlation of possible factor scores
                                                     0.99 0.99 0.53
```

fa(combinedcor, nfactors=3, fm="ml", rotate="oblimin",scores="regression")

- Combine PC scores from batting, running and defense score into one matrix.
- Conduct MLE based factor analysis using the combined data.

```
pcweights<-matrix(c(1.06,0.99*0.85,1*1.01,0.99*0.51,0,0),nrow=6,ncol=1)

PCcombined<-PCcombined%>%mutate(Total=as.matrix(PCcombined[,3:8])%*%pcweights)
PCcombined<-PCcombined%>%mutate(Unweightedtotal=apply(PCcombined[,3:8],1,sum))
```

- Divide each variables into different factors.
- PCs with communaltities lower than 0.3 have been removed.
- Use the factor loadings and portion of variance as weights.
- Perform linear combination using the calculated weights and PC scores and get the weighted sum, which will be named as PAINS (Perfomance Analysis INduced Statistics)

Aggregated Evaluation Statistics

* Calculate unweighted sum of each PCs (Only include PCs that are also included in factor analysis).

```
PCcombined$WAR<-Batraw[,"WAR"]

PCmatrix<-as.matrix(PCcombined[,c(11,9)])

rownames(PCmatrix)<-PCcombined$Name

K2<-kmeans(PCmatrix,centers=5,nstart=25)

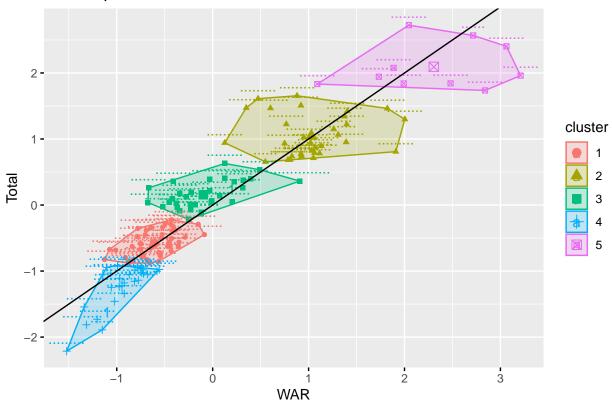
library(factoextra)
```

Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

- Use WAR as X-axis and PAINS as the y-axis to perform K-mean Clustering.
- Create 5 groups.
- Set nstart as 25 (Repeat the procedure using different starting group 25 times).

```
fviz_cluster(K2,data=PCmatrix)+geom_abline()
```

Cluster plot



• Players with poor defensive or running skills are less valued than WAR.

```
K2list<-as.matrix(K2$cluster)

K2lists<-as.data.frame(K2list)

K2lists<-K2lists%>%mutate(Name=rownames(K2list))

K2lists<-K2lists[,c(2,1)]

PCcombined1231<-merge(PCcombined,K2lists,by="Name")</pre>
```

• Create new table including the allocated cluster for each players.

1 1

0.516

```
PCcombined1231$V1<-as.factor(PCcombined1231$V1)

PCcombined1231%>%group_by(V1)%>%dplyr::summarise(Average_WAR=sum(WAR)/n())

## # A tibble: 5 x 2

## V1 Average_WAR

## <fct> <dbl>
```

```
## 2 2 3.27
## 3 3 1.27
## 4 4 -0.0332
## 5 5 5.44
```

```
Top<-PCcombined1231%>%filter(V1==3)%>%select(Name,WAR,Total)
Second<-PCcombined1231%>%filter(V1==5)%>%select(Name,WAR,Total)%>%arrange(desc(Total))
Third<-PCcombined1231%>%filter(V1==4)%>%select(Name,WAR,Total)%>%arrange(desc(Total))
Fourth<-PCcombined1231%>%filter(V1==2)%>%select(Name,WAR,Total)%>%arrange(desc(Total))
Last<-PCcombined1231%>%filter(V1==1)%>%select(Name,WAR,Total)%>%arrange(desc(Total))
head(Top)
```

```
##
                          WAR
                                   Total
       Name
## 1
       # A tibble: 6 x 1 0.7922603
## 2
                     WAR -0.5882020
## 3
                   <dbl>
                          0.1313977
## 4
                 1 0.94
                          1.6298761
## 5
                 2 1.32 0.1706952
## 6
                 3 0.88
                          0.6267546
```

head(Second)

```
##
       Name
                          WAR
                                   Total
## 1
       # A tibble: 6 x 1 12.339200
## 2
                       WAR 11.634834
## 3
                    <dbl> 10.886778
## 4
                  1 5
                           9.405617
## 5
                  2 6.14 8.879501
## 6
                  3 6.73 8.800335
```

head(Third)

```
##
       Name
                           WAR
                                     Total
## 1
       # A tibble: 6 x 1 0.03356933
## 2
                      WAR -4.13393758
## 3
                    <dbl> -4.27968824
                  1 1.22 -4.28746282
## 4
## 5
                   2 -0.16 -4.35555709
## 6
                  3 -0.35 -4.41969671
```

head(Fourth)

```
## Name WAR Total
## 1  # A tibble: 6 x 1 7.496698
## 2  WAR 7.298292
## 3  <dbl> 6.827819
## 4  1 3.02 6.667017
## 5  2 2.33 6.644821
## 6  3 3.09 6.626546
```

head(Last)

```
##
       Name
                          WAR
                                   Total
       # A tibble: 6 x 1 0.2034219
## 1
## 2
                     WAR -1.0124178
## 3
                    <dbl> -1.2626038
                 1 1.5 -1.2855334
## 4
## 5
                 2 0.8 -1.3213037
## 6
                 3 0.69 -1.4870436
```

• Divide players into each clusters.

```
##PCcombined1231<-PCcombined1231%>%mutate(Team=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3),Position=substr(PCcombined1231$Team,3,3)
```

- The second column originally contained year played, team, and position played.
- Divide those attached information into three pieces using substr
- Calculate median of PAINS and WAR for each position.

```
 \begin{tabular}{ll} \#PCcombined 1231 [which (PCcombined 1231 \$Position\%in\%c("1B","C","2B","3B","CF","DH","SS","LF","RF") == F), "Note that the property of the property of
```

- Some players who have moved team during the season had different format. (Contains two letters for team information)
- For those players, edit the dataset appropriately.