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")
which(is.na(Batraw))
```

```
## [1] 7103 7104
```

```
names(Batraw)
```

```
##  [1] "Name"                "Team"                "WAR"
##  [4] "G"                   "PA"                  "AB"
##  [7] "RUN"                 "HIT"                 "X2B"
## [10] "X3B"                 "HR"                  "TB"
## [13] "RBI"                 "BB"                  "HBP"
## [16] "IB"                  "SO"                  "DP"
## [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"
## [37] "Advancedbase"        "Runper"              "Advancedpossibility"
## [40] "Outonbase"           "SB"                  "CS"
## [43] "RAASB"               "RAABR"               "Spd"
```

```
Batraw<-Batraw%>%filter(AB>=100)
Batraw<-as_tibble(Batraw)
```

```
Hittingraw<-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

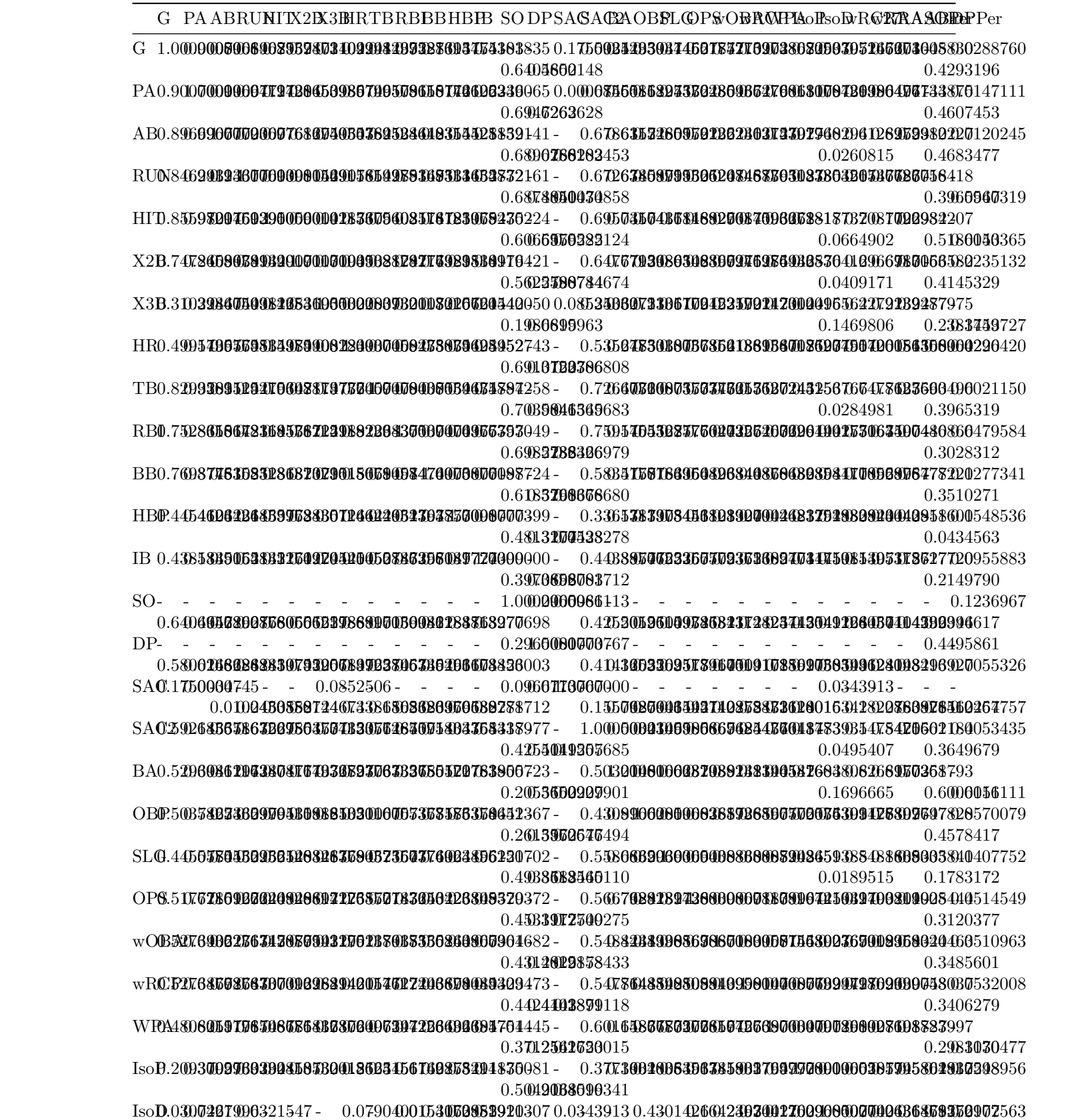
```
Runnerraw[,"CS"]<--Runnerraw[,"CS"]
Hittingraw[,c("SO","DP")]<--Hittingraw[,c("SO","DP")]
```

- Take negative to column 40 and 41

```
Hittingraw<-Hittingraw%>%mutate(SOPer=-SO/AB,DPPer=-DP/AB)

Standardhit<-scale(Hittingraw[,-1],center=T)
```

- 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)
```



G PA AB R UN H IT X2B X3B HR TB RBI BB HBP IB SO DP SAC SAC2 A OBP SLG OPS wOBA wRC wRAA Iso P Iso D wRC27 RAA SOPer DPPer

wRC [illegible] - 0.547 [illegible] 0.4057[illegible]28780 [illegible] 0.3550175

wRAA [illegible] - 0.547 [illegible] 0.441[illegible]603976 [illegible] 0.3418433

RAA [illegible] - 0.566 [illegible] 0.4299[illegible]928460 [illegible] 0.24[illegible]36274

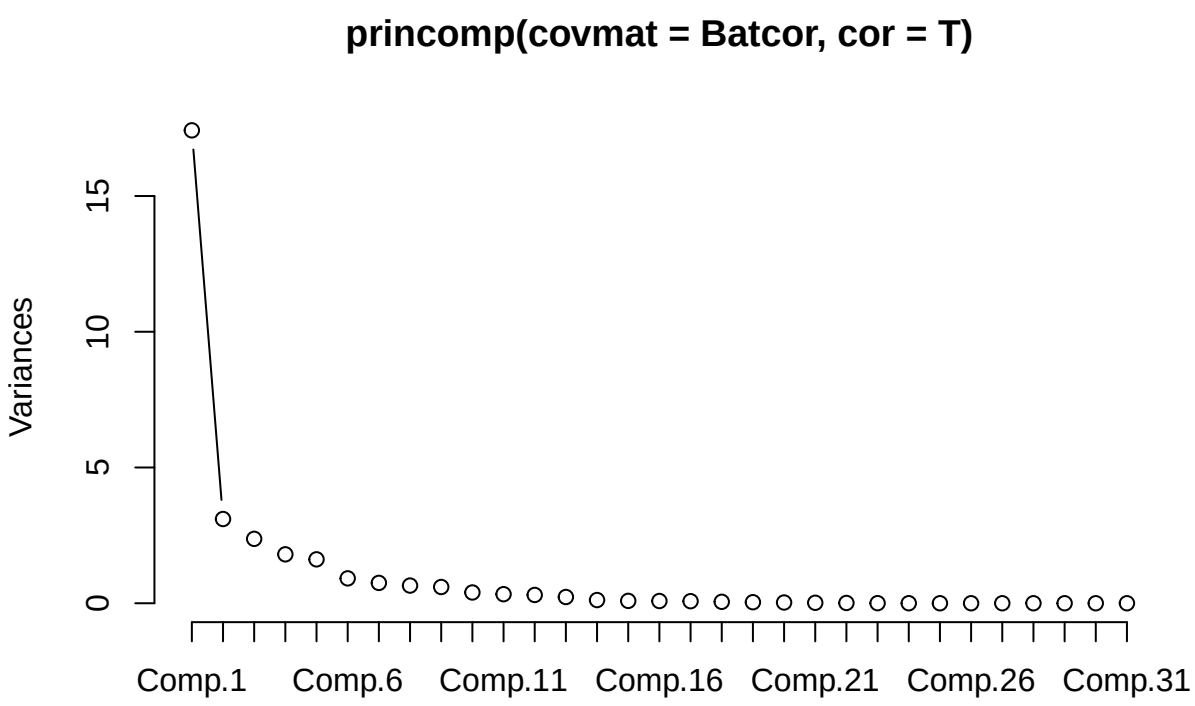
SOPer - - - - - - - 0.00[illegible]4220 - - - - - 0.4495861 - - - - - - - 0.19[illegible]02 - 1.0000000
0.42[illegible] 0.35[illegible]14900

DPPer [illegible]245 0.02[illegible]967 0.00[illegible] 1.0000000
0.09[illegible]749727 0.70[illegible]016111 0.1170477 0.01[illegible]4900

```r
summary(princomp(covmat=Batcor,cor=T))
```

```
## Importance of components:
##                           Comp.1    Comp.2     Comp.3     Comp.4    Comp.5
## 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    Comp.10
## 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.19      Comp.20      Comp.21      Comp.22
## 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.23      Comp.24      Comp.25      Comp.26
## 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
```

```r
screeplot(princomp(covmat=Batcor,cor=T),npcs=31,type='l')
```



- 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))
```

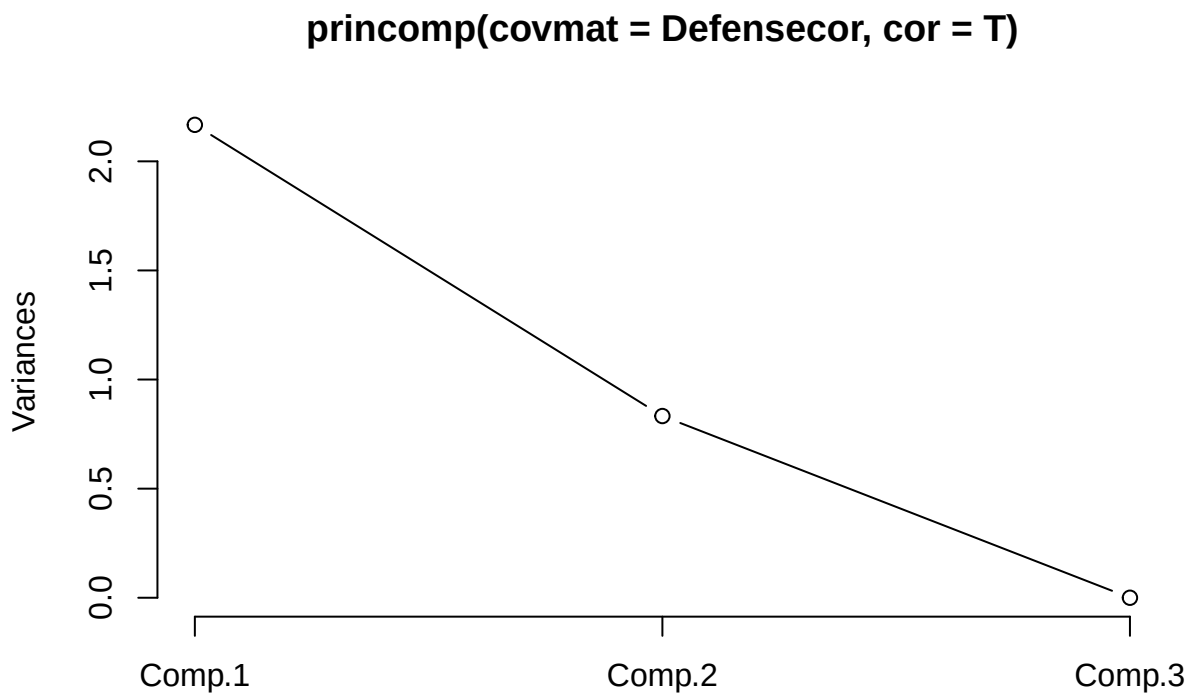
- Retain PC scores through multiplication of standardized Batting data and each PC loadings.
- Use relative proportion of variance explained by each PC as a weight (Put PC2 proportion as default).

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

```
## Importance of components:
##                           Comp.1    Comp.2       Comp.3
```

```
## 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')
```

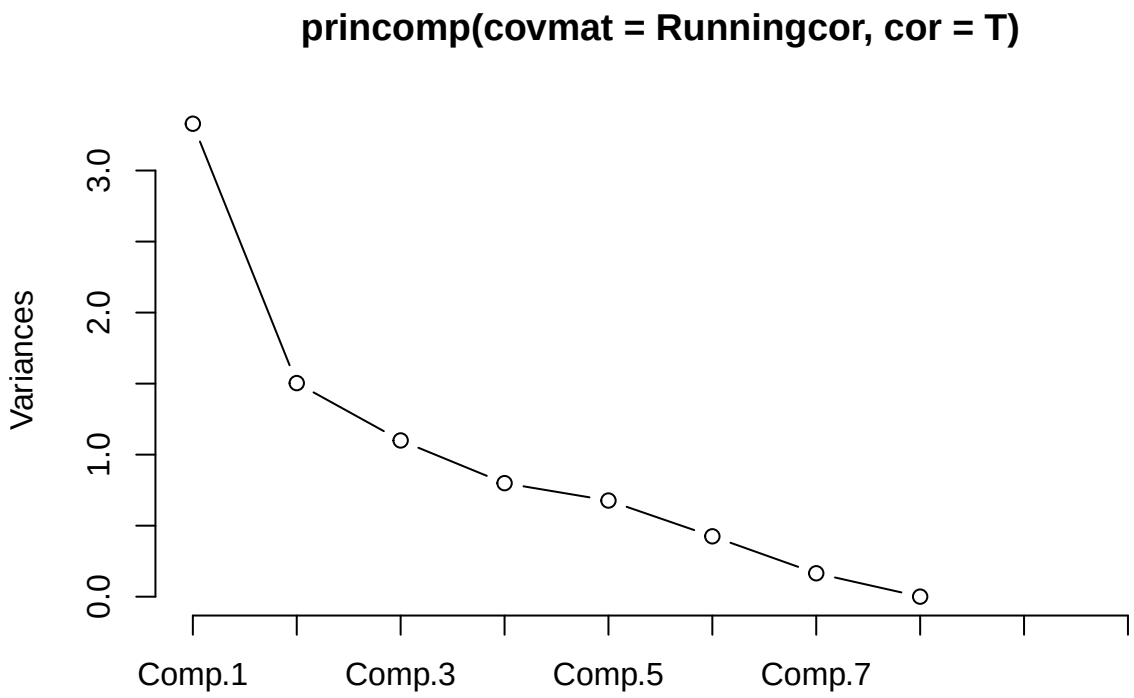



```
PCDef<-princomp(covmat=Defensecor,cor=T)
PCDefload<-PCDef$loadings[,1]
PCDef1<-data.frame(PCD1=apply(StandardDef%*%PCDefload,1,sum))


Runnerraw[which(is.na(Runnerraw)),1]<-0

StandardRun<-scale(Runnerraw,center=T)
Runningcor<-cor(StandardRun)
summary(princomp(covmat=Runningcor,cor=T))
```

```
## Importance of components:
##                           Comp.1    Comp.2    Comp.3     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
```

```
## Cumulative Proportion  0.97932434 0.99993547 1.000000e+00
```

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




```
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)


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.

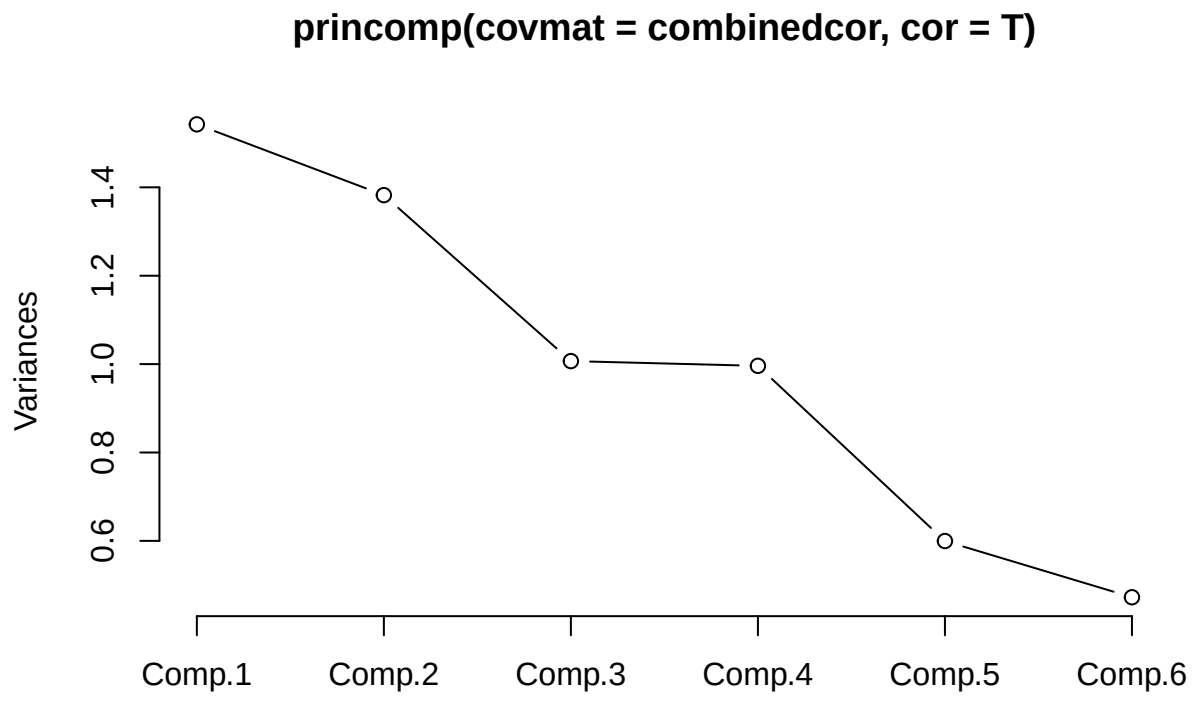
```
library(psych)
```

```
##
##          : 'psych'

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(GPArotation)
combinedcor<-cor(PCcombined[,3:8])

summary(princomp(covmat=combinedcor,cor=T))
```

```
## Importance of components:
##                            Comp.1    Comp.2    Comp.3    Comp.4     Comp.5
## 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='l')
```



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

```
## Factor Analysis using method =  ml
## Call: fa(r = combinedcor, nfactors = 3, rotate = "oblimin", scores = "regression",
##     fm = "ml")
## 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
```

- 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)

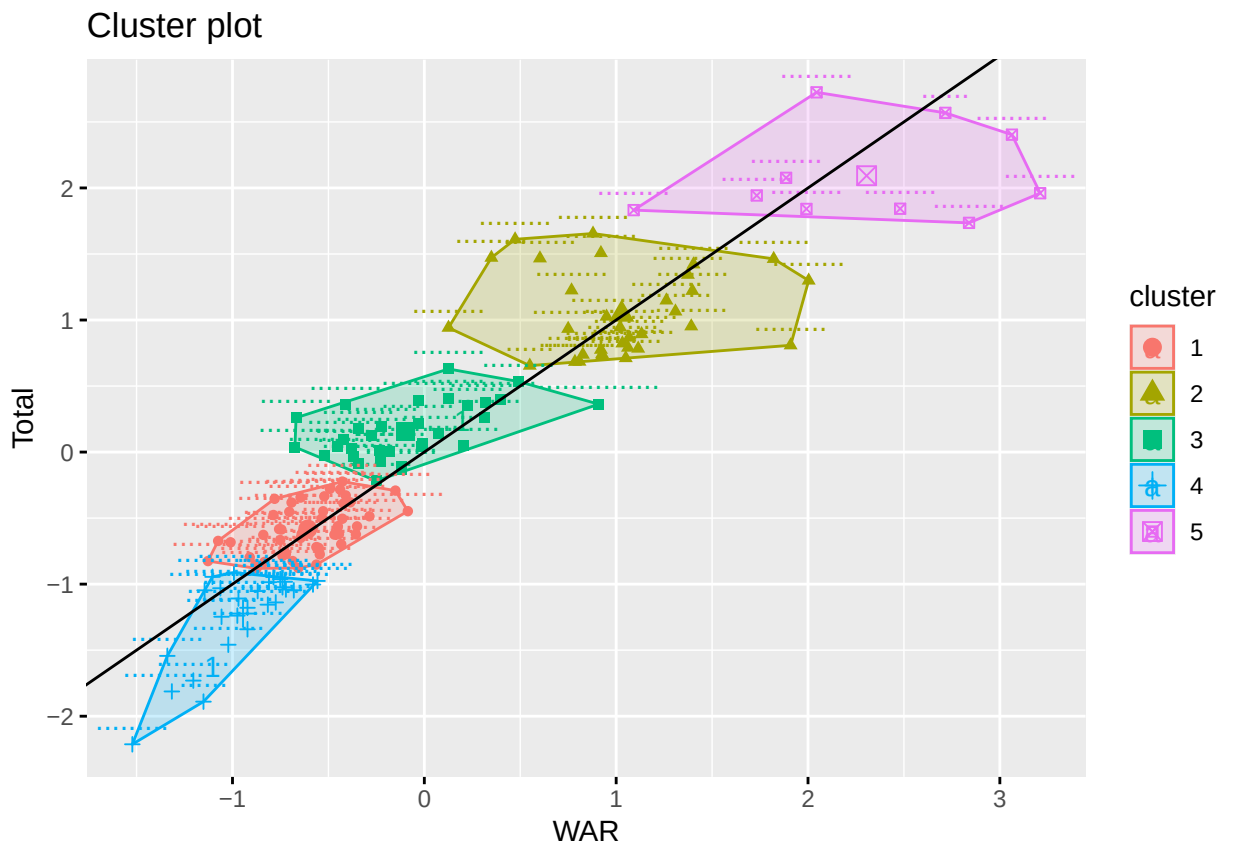
Aggregated Evaluation Statistics
* Calculate unweighted sum of each PCs (Only include PCs that are also included in factor analysis).

```
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()
```



- 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")
```

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

```
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>
## 1 1           0.516
```

```
## 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)
```

```
##     Name               WAR      Total
## 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
## 4              1  1.22 -4.28746282
## 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
## 1    # A tibble: 6 x 1  0.2034219
## 2                  WAR -1.0124178
## 3                <dbl> -1.2626038
## 4              1  1.5  -1.2855334
## 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(PCcombin

##PCcombined1231%>%group_by(Position)%>%dplyr::summarise(PAINSMedian=median(Total),WARMedian=median(WAR
```

- 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.

```
##PCcombined1231[which(PCcombined1231$Position%in%c("1B","C","2B","3B","CF","DH","SS","LF","RF")==F),"N
```

- 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.