410 Project PCA Analysis

Madison Nafarrete

2024-04-28

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Including

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Loading Libraries

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.3.3
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.0     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(tinytex)
```

```
## Warning: package 'tinytex' was built under R version 4.3.3
```

```
library(dplyr)
library(ggmosaic)
library(ggrepel)
library(tidyr)
library(janitor)
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.3.3
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
##
## Attaching package: 'GGally'
##
## The following object is masked from 'package:ggmosaic':
##
##     happy
```

```
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.3.3
```

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

```
library(FactoMineR)
```

```
## Warning: package 'FactoMineR' was built under R version 4.3.3
```

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.3.3
```

```
## corrplot 0.92 loaded
```

For this project:
1. A Data Science topic: Principle Component Analysis (PCA)
2. A Data Set: MLB Pitching Data (source: https://www.kaggle.com/datasets/open-source-sports/baseball-databank)
3. PCA reduces the number of variables or features in a data set while still preserving the most important information like major trends or patterns

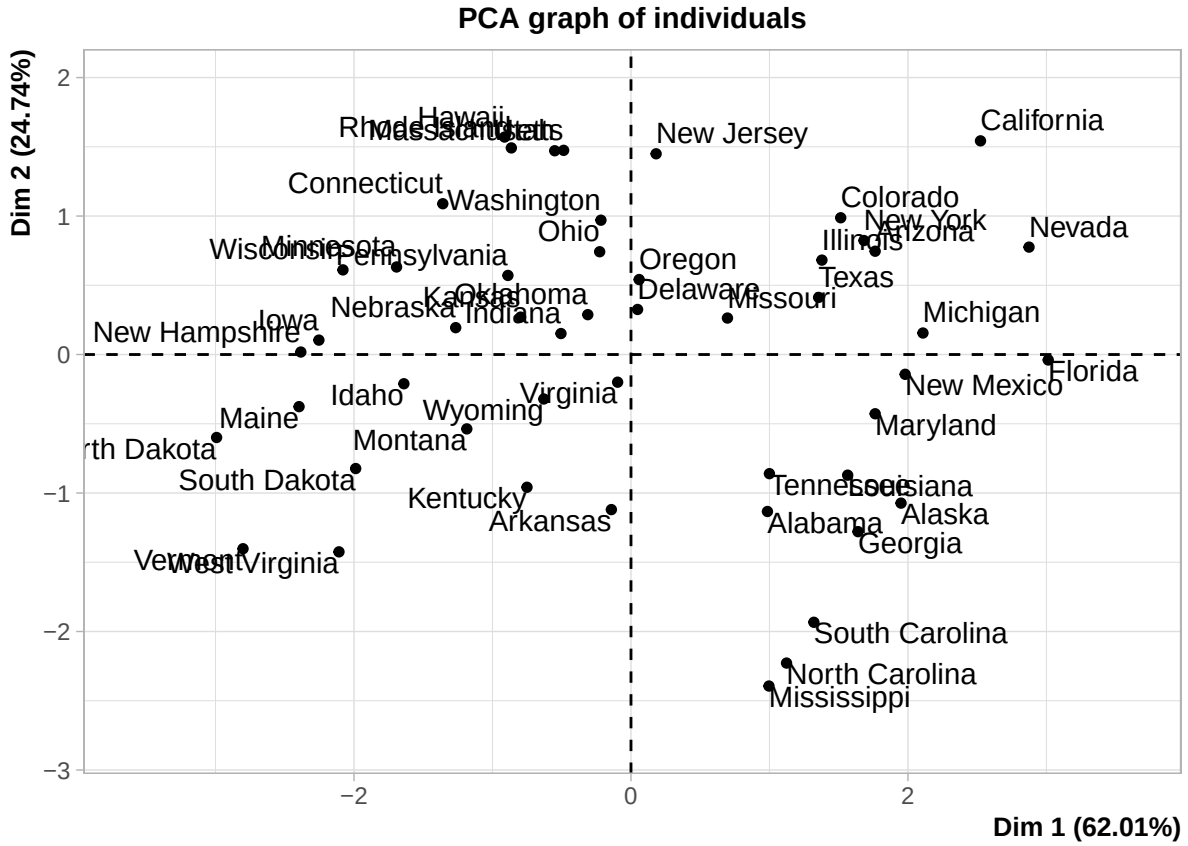
What is Principal Component Analysis?
Principal Component Analysis (PCA) is a method to use on a data set to figure out which numeric variables covary. (Bruce et al. Practical statistics for data scientists pgs 284-285) The goal of PCA is to shrink your data set into a smaller set of variables, which are called "principal components". These principal components explain the majority variability in your original data set. The code chunk below is an example of a PCA from the 'help' menu when looking up the function for PCA analysis ('PCA'). PCA uses variance as a form

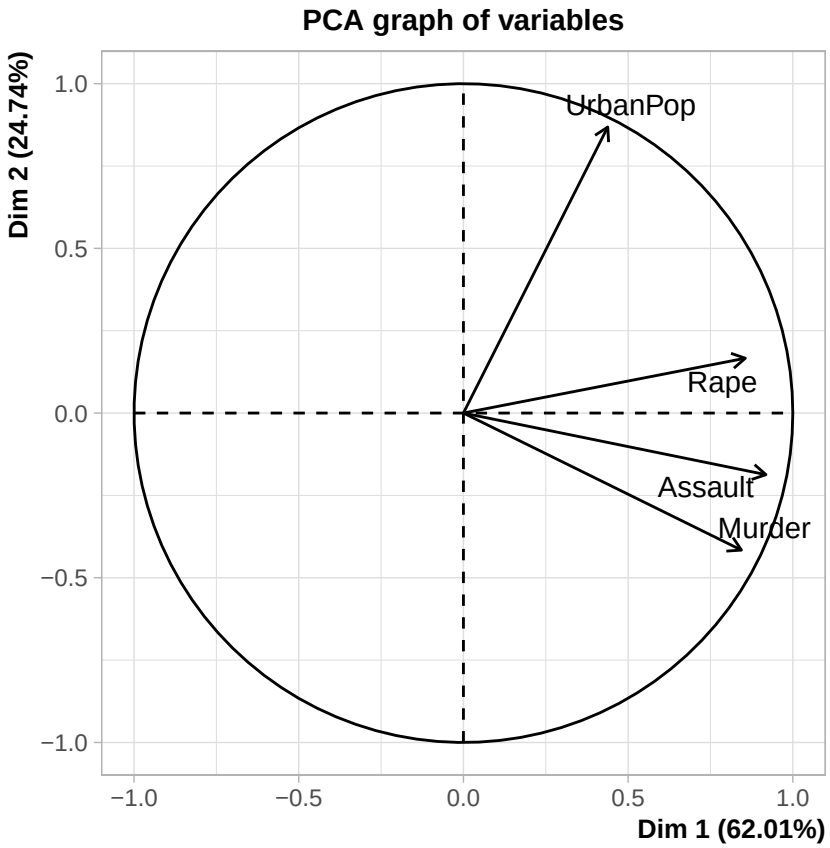
of measurement for each principal component using eigenvalues. Eigenvalues in PCA explain the variance of each principal component.

Data Cleaning

When prepping the data set for a principal components analysis make sure your data set is cleaned and optimized for your analysis. The data set in this example is already cleaned since it is a data set built into R. However, when loading your own data set make sure that there is only data you want to analyze and avoid data character(chr) data, so if there is chr data make sure to remove or not include them in your analysis.

```
# function in R to conduct a PCA
murder_res = PCA(USArrests, scale.unit = TRUE, graph = TRUE)
```

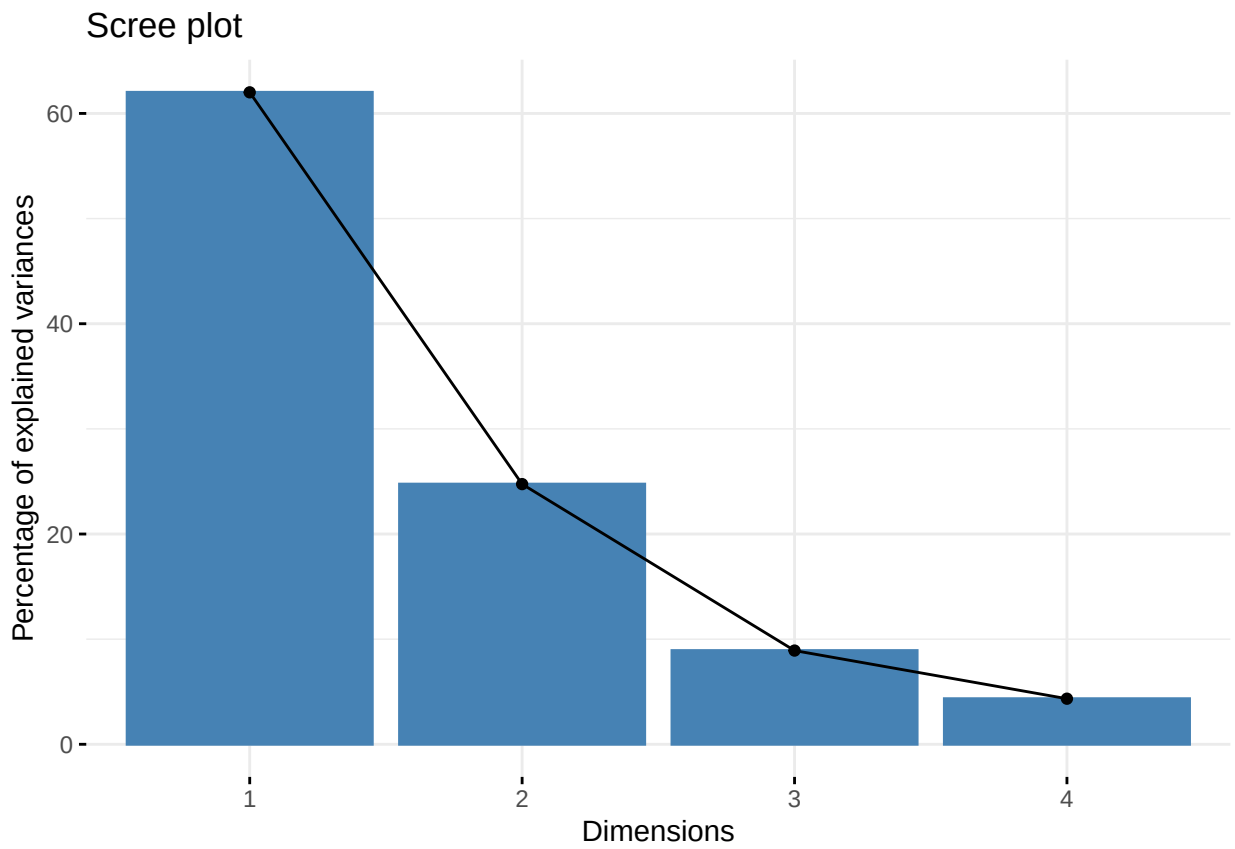





```
#visualizing eigenvalues
get_eigenvalue(murder_res)
```

```
##       eigenvalue variance.percent cumulative.variance.percent
## Dim.1  2.4802416        62.006039                      62.00604
## Dim.2  0.9897652        24.744129                      86.75017
## Dim.3  0.3565632         8.914080                      95.66425
## Dim.4  0.1734301         4.335752                     100.00000
```

```
fviz_eig(murder_res)
```



Parts of PCA
In using PCA(), the data set and variables in the data set are already standardized. Also, with graph = TRUE you can see that the function already provides visualizations and there is no need to do them separately. The standard deviations for each component represent the covariance matrix of the data by taking the square roots of its eigenvalues. These standard deviations show the amount of variability within the data explained by each principal component. A principal component "explains most of the variability of the full set of variables" (Bruce et al. Practical pg. 285) The 'Rotation Matrix' is the loadings of the variables in the original data set, loadings are the "weights that transform the predictors into components"(Bruce et al. Practical pg. 285).The points in the biplot above represent each state and how they are projected onto the first two components, and the vectors represent the variables in which the direction and magnitude show the variables contribution to each component. The 'biplot()' function is a graphical representation of observations and variables within the PCA. Recall, that PCA is used to "reduce the dimensionality of a data set consisting of a large number of interrelated variables, while retaining as much as possible of the variation present in the data set" (Mishra et al. Principal component analysis) Shrinking dimensions is reminiscent of matrices and matrixes, so the goal, generally, is to choose variables that have strong correlation with what what you are analyzing and visualize those patterns to see how much each variable affects the variance within a data set.

Interpretation of PCA Analysis Example The summary results shows that the first component has the strong negative correlation to all the original variables that were analyzed. Which suggest that 'Rape', 'Assault', and 'Murder' have a possibility to covary, so The second component has a strong positive correlation with the 'Rape' variable. The third component has a strong negative correlation with 'Assault'.

Interpretation Biplots: By examining the position of data points in relation to each other and the direction and length of arrows, you can interpret the relationship between the observations and variables in the dataset. Data points that align with the direction of an arrow have high values for the corresponding variable. On the other hand, data points that point in the opposite direction have low values for the variable. Variables that point in the same direction as each other have similar patterns of variation and are similar in

contribution to the principal components.

Interpretation of Correlation Circle and Correlation Plots
The correlation circle does exactly what the names implies. It visualizes the correlation between all variables. If variables are negatively correlated with each other they would be facing opposite of each other. If there is strong correlation will usually group and align together. Correlation plots also visualize correlation between variables within a data set. This can be useful in PCA because the analysis depends on variables that have strong correlation with one another.

Interpretation of Scree Plots
A scree plot is a plot of the variances of the principal components NOT the original data set. It shows the importance of each component explained as a proportion of explained variance. The scree plot helps you decide how many principal components to retain in your analysis. Typically, you retain the principal components before the elbow point on the scree plot, as they capture the majority of the variance in the data. Principal components beyond the elbow point (where the data flattens out) may capture noise or irrelevant variability in the data and can be discarded without significant loss of information.

Question: Which principal components of pitching contribute to being the best fantasy baseball pitcher?

Loading CSV File & Data Cleaning
First, we load the csv file 'Pitching.csv', which includes the years in which baseball first became a national sport. Then, filter out the years in which fantasy baseball is not relevant. For this project, I have specifically chosen the 2015 season because it is the most recent season within this data set relevant to how fantasy baseball points are scored. In order to have a good PCA analysis, you have to filter out the data set to include only the variables that are relevant to what you are analyzing.

Table 1: Table for Fantasy MLB Pitchers (first 6 pitchers shown)
(2015 Season)

playerID	yearID	stint	teamID	lgID	W	L	G	GS	CG	SHO	SV	IPouts	H	ER	HR	BB	SO	BAOpp	ERA	IBB	WP	HBP	BK	BFP	GF	R	SH	SF	GIDP
aardsda01	2015	1	ATL	NL	1	1	33	0	0	0	0	92	25	16	6	14	35	0.224	4.70	3	1	1	0	129	9	17	0	1	NA
abadfe01	2015	1	OAK	AL	2	2	62	0	0	0	0	143	45	22	11	19	45	0.254	4.15	3	4	1	0	205	17	23	3	3	NA
achteje01	2015	1	MIN	AL	0	1	11	0	0	0	0	40	12	10	4	6	14	0.236	6.75	1	0	0	0	58	4	10	0	0	NA
adamsau01	2015	1	CLE	AL	2	0	28	0	0	0	1	100	37	14	2	13	23	0.276	3.78	0	1	0	0	149	9	15	2	0	NA
adcocma01	2015	1	CIN	NL	1	2	13	0	0	0	0	54	15	12	3	12	13	0.226	6.00	0	1	1	0	83	4	12	1	1	NA
affelje01	2015	1	SFN	NL	2	2	52	0	0	0	0	106	43	23	6	14	21	0.293	3.86	2	1	2	0	163	7	24	0	0	NA

Table 2: Table for Pitchers for Fantasy Baseball(2015 Season)

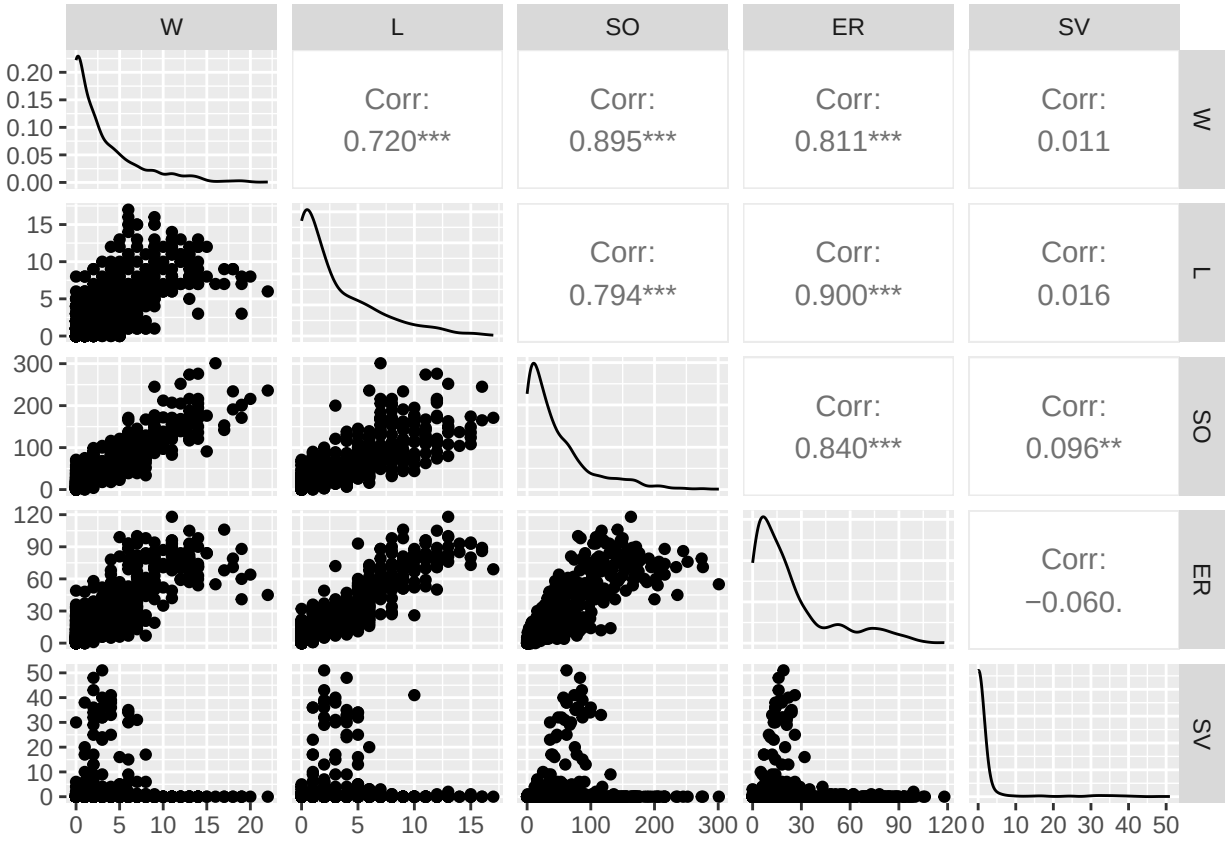
W	L	SO	ER	SV
1	1	35	16	0
2	2	45	22	0
0	1	14	10	0
2	0	23	14	1
1	2	13	12	0
2	2	21	23	0

Exploratory Data Analysis: Visualizations
Before doing a PCA it is important to do some exploratory data analysis to check and visualize your data. Below is a correlation plot to check the correlation of each variables to each other. In PCA, correlation is important because it will affect the outcome of variances for each variable and how they play into fantasy points.

Correlation Plot

The correlation plots shows that the selected variables for this PCA have weak to strong correlation with each other. There is an exception as saves and earned runs are weakly negatively correlated with each other. Overall, the correlation plot shows mid to strong positive correlation between all variables with saves (SV) being the least correlated most variables. There is also not many negatively correlated variables which shows that most of the impact of these variables is positive in terms of accruing fantasy points during the season. As expected, wins(W) has a strong positive correlation to losses (L), strikeouts(SO) and Earned Runs (ER), since wins has such a strong correlation with almost all the other variables within the analysis the component weighted with wins will have highest proportion of variance to the explanation of fantasy points.

```
# Correlation Plots of Variables
ggpairs(pitchers_fantasy_2015)
```



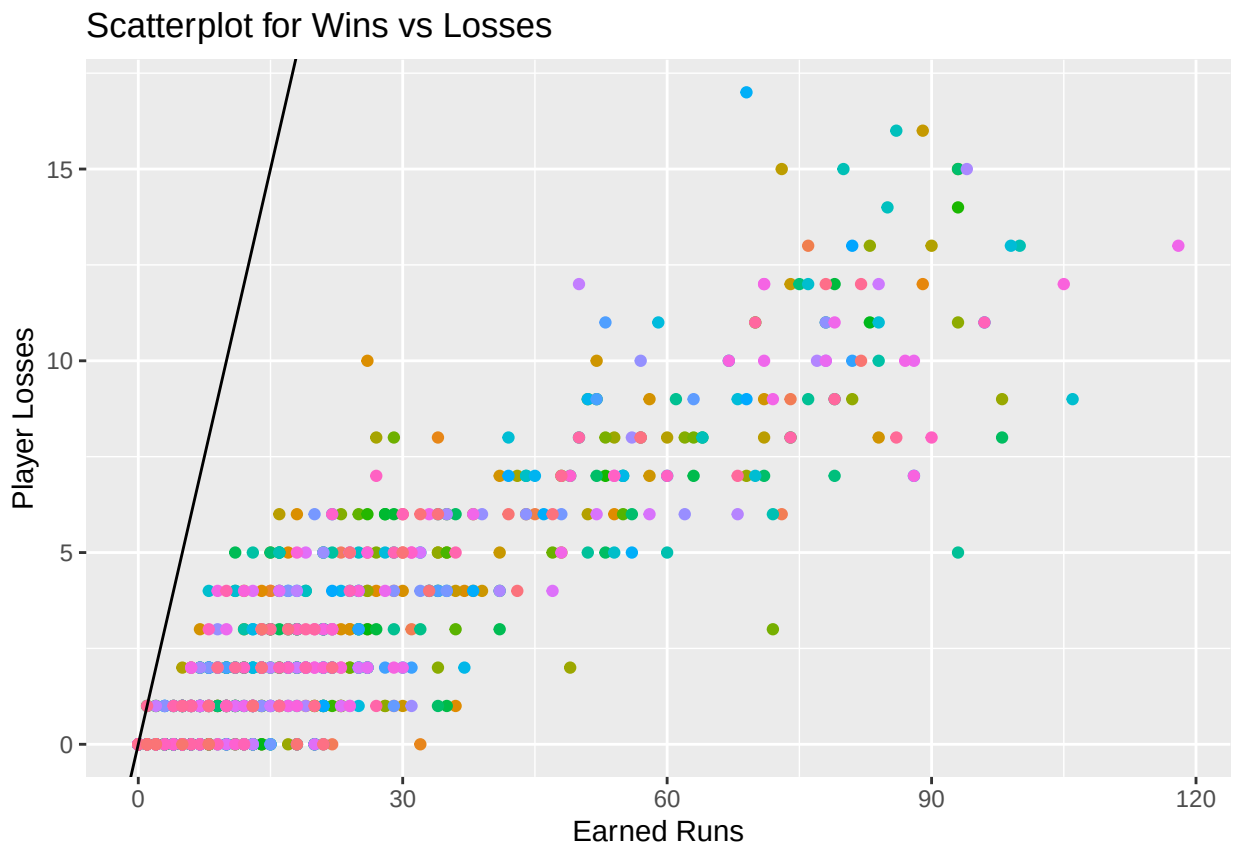

```
p_cor = cor(pitchers_fantasy_2015)
corrplot(p_cor, method = 'number')
```

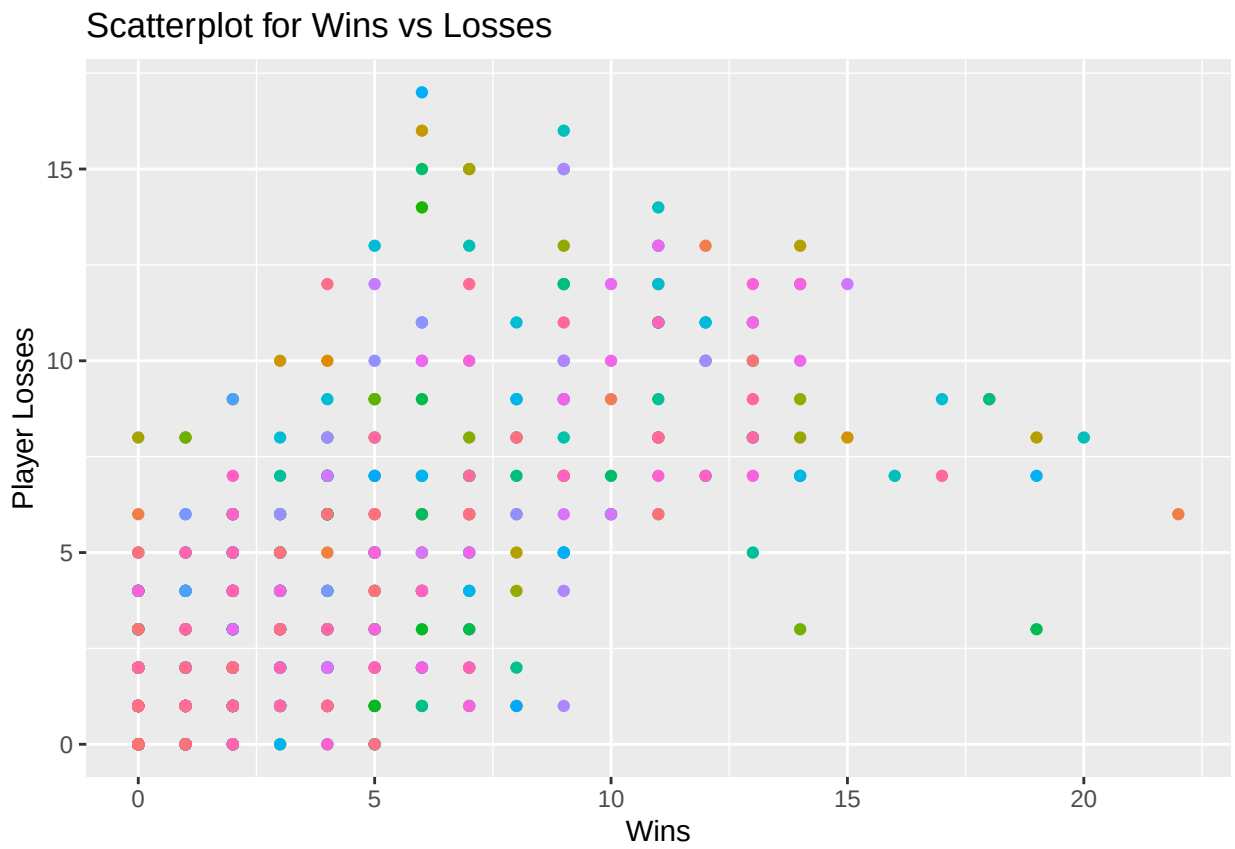
Scatterplots

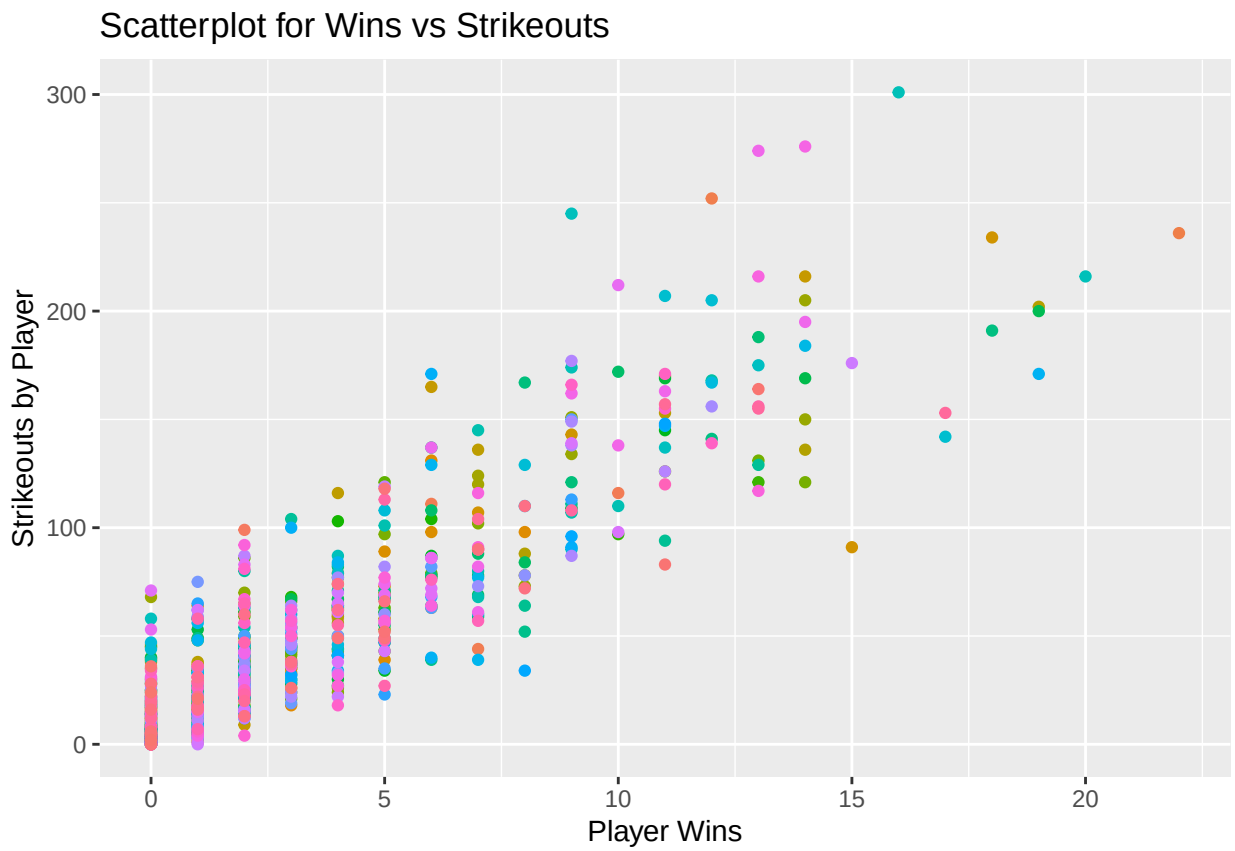
I used scattplots to compare the relationship of each variable. More often than not the plots showed patterns which is a sign of correlation between each other. The highest correlation, as shown in the correlation plot above, is between Losses and Earned Runs with .900. The closer the correlation is to 1, the stronger the correlation is between the two variables. With the ER and Losses plot you can see that several of the points are aligned, clustered and not randomly scattered. Wins and Strikeouts also have a strong positive correlation with each other and showed the points clustered and patterned in straight lines. The colored points represent players, however there is over 800 players for this season, some points may be overlapping. Additionally, I had to turn the legend off because it would block the whole graph.

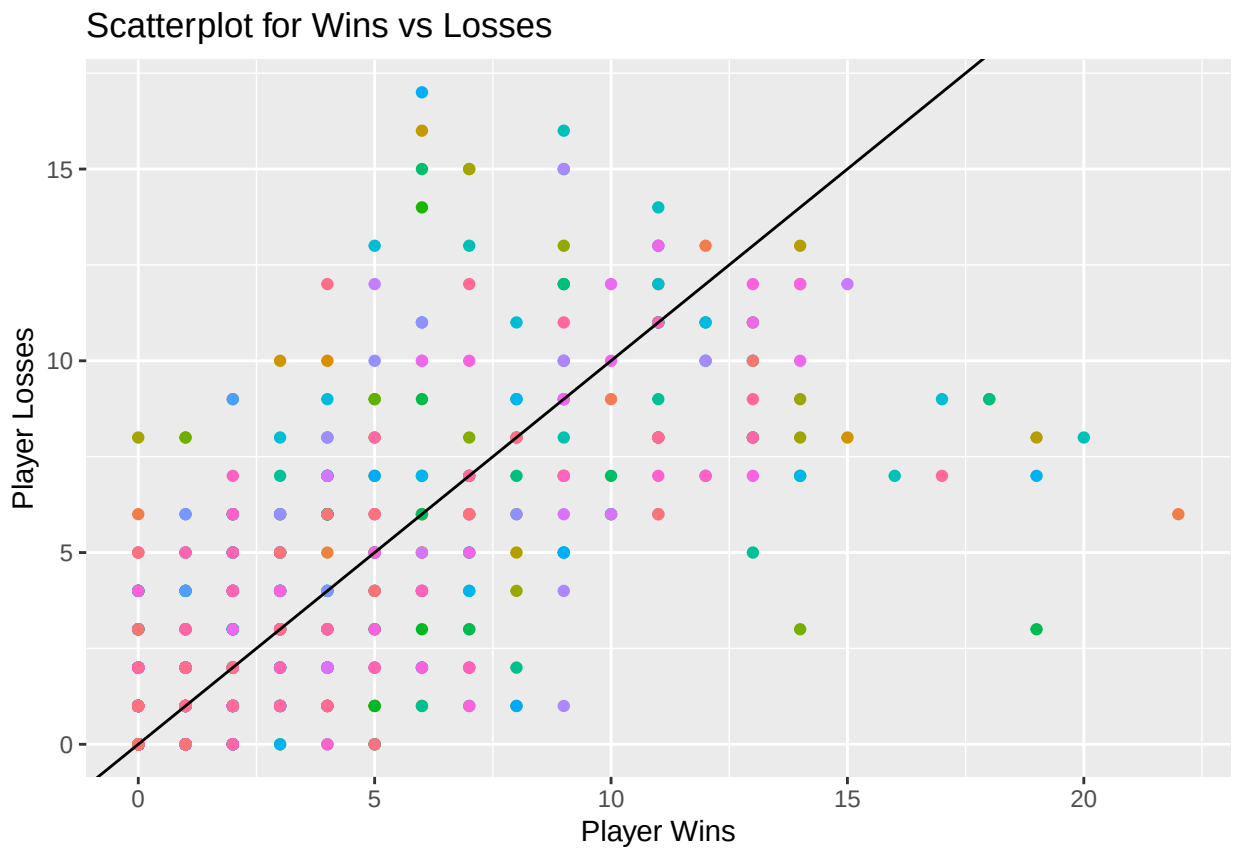
```
# Scatterplot
ggplot(pitch_fant_2015, aes(x = ER, y = L, color = playerID)) + xlab("Earned Runs") + ylab("Player Loss
```

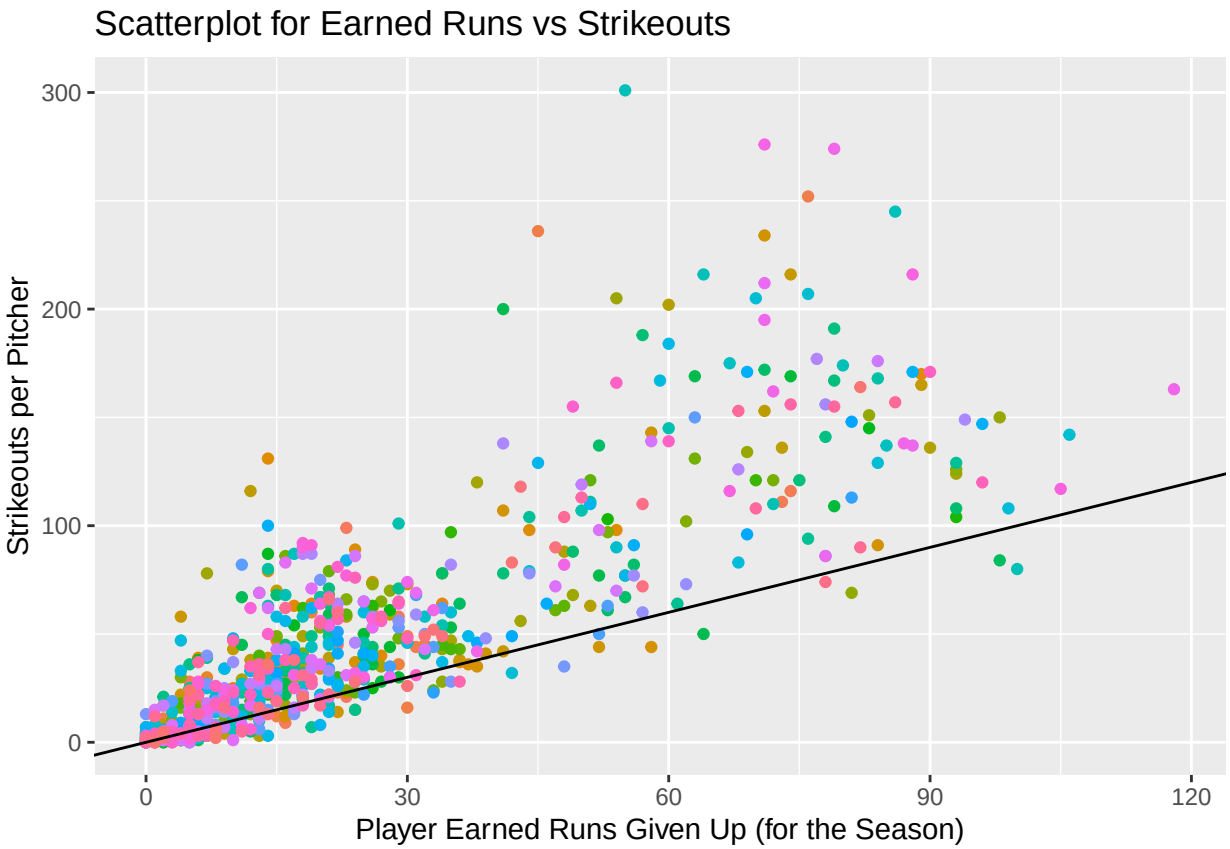
```
## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as
## of ggplot2 3.3.4.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```


Scatterplot for Wins vs Losses
Player Losses
Earned Runs
0
5
10
15
0
30
60
90
120


Scatterplot for Wins vs Losses
Player Losses
Wins
0
5
10
15
20


Scatterplot for Wins vs Strikeouts
Strikeouts by Player
300
200
100
0
0
5
10
15
20
Player Wins

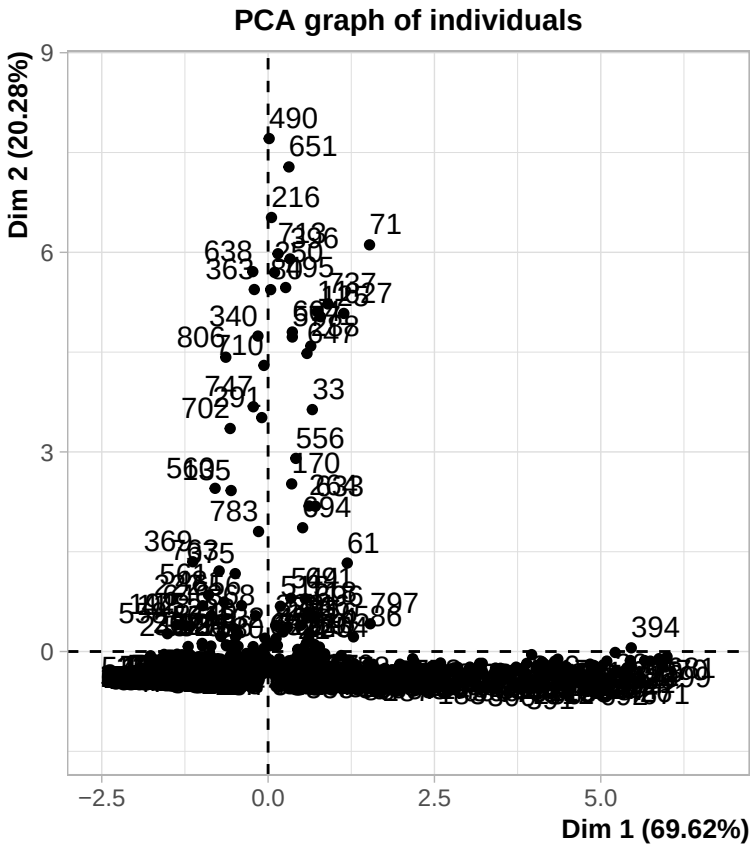

Scatterplot for Wins vs Losses
Player Losses
Player Wins
0
5
10
15
20

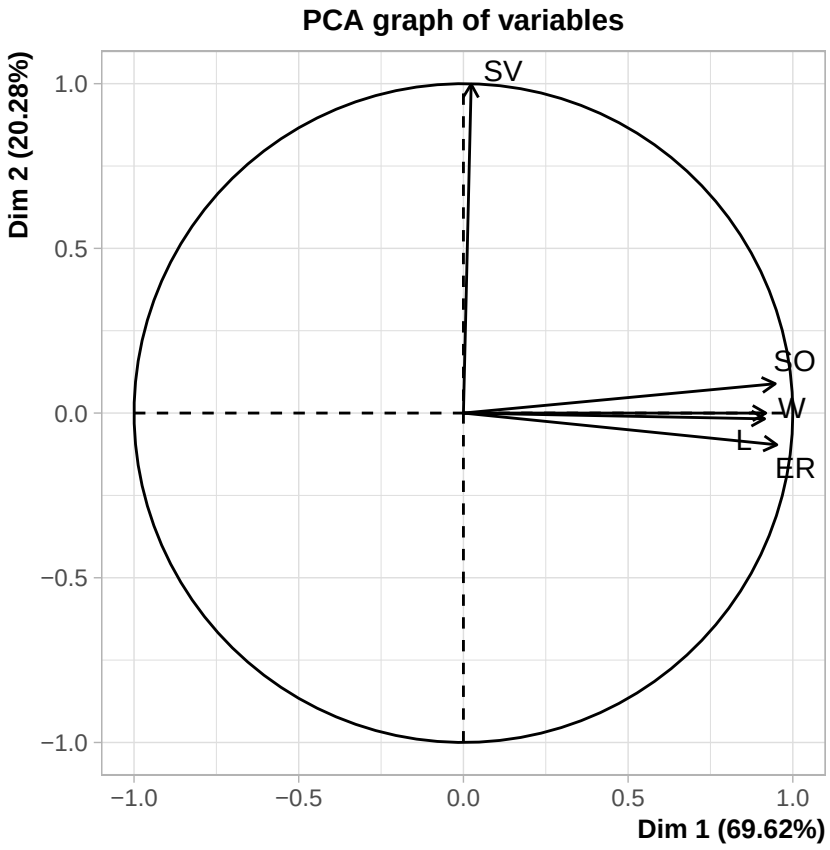


Principal Component Analysis on 2015 Fantasy MLB Season for Pitchers

I am using PCA() from the FactoMineR package which is now in base R. This specific PCA() function already runs most of the visualizations for PCA within the function so plotting the scree plot and biplot separately is not necessary. Diving into this principal component analysis will further help understand how these specific pitching statisitcs contribute to fantasy points.

```
# In-depth results of PCA
pca_res = PCA(pitchers_fantasy_2015, scale.unit = TRUE, ncp = 5, graph = TRUE)
```


PCA graph of individuals
Dim 2 (20.28%)
Dim 1 (69.62%)
9
6
3
0
−2.5
0.0
2.5
5.0
490
651
216
71
638
340
806
710
747
33
702
556
170
783
61
797
394

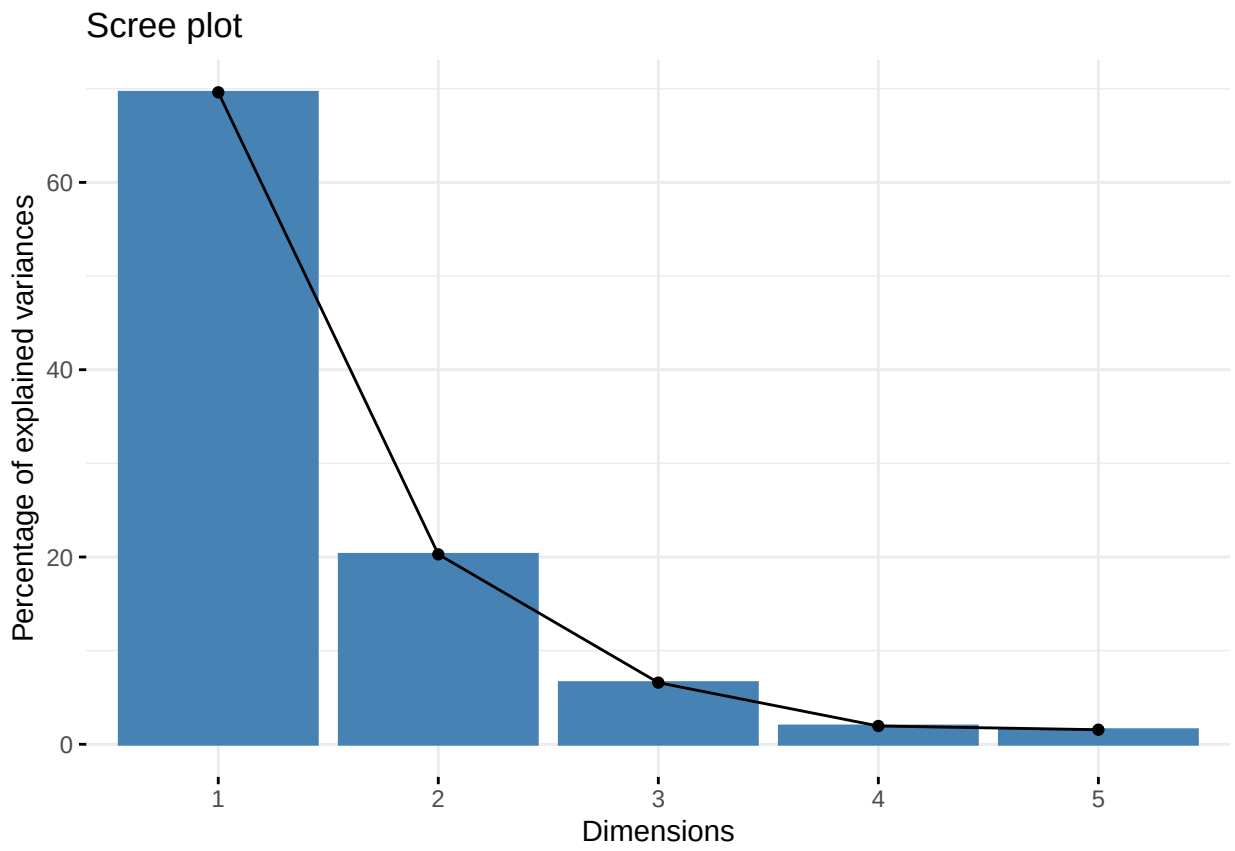



```
# shows functions to further analyze PCA
#pca_res

#visualizing eigenvalues
#displays the eigenvalues, variance percentage, and cumulative variance
get_eigenvalue(pca_res)
```

```
##       eigenvalue variance.percent cumulative.variance.percent
## Dim.1 3.48090332        69.618066                    69.61807
## Dim.2 1.01414569        20.282914                    89.90098
## Dim.3 0.32929569         6.585914                    96.48689
## Dim.4 0.09779008         1.955802                    98.44270
## Dim.5 0.07786521         1.557304                   100.00000
```

```
# scree plot
fviz_eig(pca_res)
```




```
# Extracting results for variables
pca_pitch = get_pca_var(pca_res)
pca_pitch
```

```
## Principal Component Analysis Results for variables
##  ===================================================
##   Name       Description
## 1 "$coord"   "Coordinates for the variables"
## 2 "$cor"     "Correlations between variables and dimensions"
## 3 "$cos2"    "Cos2 for the variables"
## 4 "$contrib" "contributions of the variables"
```

```
pca_pitch$cor # correlations btwn vars and dimensions
```

```
##         Dim.1         Dim.2       Dim.3       Dim.4        Dim.5
## W  0.91794064 -9.602122e-05 -0.35196612  0.13622318  0.122262294
## L  0.91439723 -1.748902e-02  0.36833887 -0.06426004  0.154171854
## SO 0.94671185  8.906519e-02 -0.19253317 -0.22476042 -0.090652021
## ER 0.95149997 -9.633387e-02  0.17613978  0.15313369 -0.175762451
## SV 0.02346226  9.983121e-01  0.04059284  0.03371644 -0.006160282
```

```
pca_pitch$cos2 # quality of representation
```

```
##           Dim.1        Dim.2       Dim.3       Dim.4        Dim.5
```

```
## W  0.8426150173 9.220075e-09 0.123880150 0.018556755 1.494807e-02
## L  0.8361222966 3.058657e-04 0.135673524 0.004129353 2.376896e-02
## SO 0.8962633353 7.932608e-03 0.037069020 0.050517248 8.217789e-03
## ER 0.9053521978 9.280215e-03 0.031025221 0.023449927 3.089244e-02
## SV 0.0005504776 9.966270e-01 0.001647778 0.001136798 3.794908e-05
```

```
pca_pitch$contrib # contributions of the variables to the principal components
```

```
##          Dim.1        Dim.2      Dim.3     Dim.4       Dim.5
## W  24.20679171 9.091470e-07 37.6197297 18.976112 19.19736580
## L  24.02026769 3.015994e-02 41.2011232  4.222670 30.52577872
## SO 25.74800998 7.821961e-01 11.2570619 51.658867 10.55386521
## ER 26.00911641 9.150771e-01  9.4216904 23.979863 39.67425338
## SV  0.01581422 9.827257e+01  0.5003947  1.162488  0.04873689
```

Results
The biplot above shows that the first component is responsible for 69.6% of the variation, while the second component has the second highest explanation in variation with 20.3%. After the second component there is a steep drop off for components 3, 4, and 5, the last two eigenvalue variances are less than 2% which means they have little impact on fantasy points. The circle correlation plot shows that most of the variables are positively correlated with each other, the exception is saves (SV) where it accounts for most of the variability in component 2. The pca_pitch$cos2 function represents how well the principal components explain the variability of the variables in the original data set. So, you can see principal component 1 represents four of the five variables, meaning they mostly contribute to principal component one, while the rest of the components have very little impact on the variables from the original data set. In relation to fantasy points for these visualizations, there is a strong correlation between the variables and their contributions to fantasy points. The scree plot shows that component 1 is responsible for most of the variables, as it contributes to over 60% variation, as mentioned before. In addition, the first four variables have the most affect on fantasy points, while saves is difficult to interpret because not all pitchers get saves and saves don't contribute many points compared to getting a win.

Conclusion In conclusion, using principal component analysis to explore the variables put into fantasy pitching was very interesting and insightful. The use of variance and correlation to figure out which variables had the most impact on each principal component, and being able to visualize that using different plots such as the scree plot made it easier to interpret the effect the variables and components had on each other. It shed light on why fantasy points are distributed the way they are. For example, wins being the most points and stats like saves do not get as many points.

Future In the future, I think it would be cool to try and test hitters with this method too. It would be interesting to see if someone can find a set algorithm to optimize fantasy points for their team by using PCA. Hopefully, it can be useful in sports science so that players can figure out how to optimize their pitching arsenal and see how they can improve during the season or future seasons. The use of PCA can be expanded besides being used for census, financial and agricultural data.

Bibliography Bruce, Peter C., et al. Practical Statistics for Data Scientists: 50+ Essential Concepts Using r and Python. O'Reilly Media, 2020.

Kassambara, Alboukadel. Practical Guide to Principal Component Methods in R: PCA, (M)Ca, FAMD, MFA, HCPC, Factoextra. STHDA, 2017.

Mishra, Sidharth, et al. "Principal component analysis." International Journal of Livestock Research, 2017, p. 1, https://doi.org/10.5455/ijlr.20170415115235.