Volleyball Scores

Rose Hedderman EID: rrh2298

3/22/2021

Introduction

This report is an explanatory variable analysis over volleyball statistics for a mens and a womens team. The datasets in this report are called ‘mens’ and ‘womens’ and they were found on Kaggle as CSV files. Both datasets have the same variables which include points scored from blocking (Blocks), attacking (Attacks), serving (Serves), and the total points for each player (Total). There are also statistics on the Rank, ShirtNumber, and Team for each player. These datasets were combined and a Gender variable was created to compare. The topic is of interest to me because I have played volleyball for about eight years and was intrigued to compare mens stats against womens. The results of this report determine that there is not a clear cluster that shows gender when scores are analyzed. Implications of this study include that mens play does not differ from womens very much. This could encourage co-ed play from either side: encourage men to play with female friends and women to team up with male friends. There does not appear to be an athletic difference. However, these are statistics from some of the best players.

Tidy

Both datasets, mens and womens, are imported and read in as excel files. They have mostly the same variables, so a new one was added as an identifier between the two before joining in the next step.

```
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 3.6.3
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.6.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.6.3
```

```
## -- Attaching packages ---------------------------------------------------------------------------
---------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.3
## v tibble  3.0.5     v stringr 1.4.0
## v tidyr   1.1.2     v forcats 0.5.0
## v readr   1.3.1
```

```
## Warning: package 'ggplot2' was built under R version 3.6.3
```

```
## Warning: package 'tibble' was built under R version 3.6.3
```

```
## Warning: package 'tidyr' was built under R version 3.6.3
```

```
## Warning: package 'readr' was built under R version 3.6.3
```

```
## Warning: package 'purrr' was built under R version 3.6.3
```

```
## Warning: package 'stringr' was built under R version 3.6.3
```

```
## Warning: package 'forcats' was built under R version 3.6.3
```

```
## -- Conflicts ------------------------------------------------------------------------------------
----------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(cluster)
mens <- read_excel("C:\\Users\\roseh\\OneDrive\\Desktop\\Spring 2021\\SDS 348\\mens.xlsx")
womens <- read_excel("C:\\Users\\roseh\\OneDrive\\Desktop\\Spring 2021\\SDS 348\\womens.xlsx")

# Make a gender column in each dataset and label it with M or W respectively so merge goes successfully
mens <- mens %>%
  mutate(Gender = 'M')

womens <- womens %>%
  mutate(Gender = 'W')
mens
```

```
## # A tibble: 255 x 9
##     Rank ShirtNumber Name               Team  Attacks Blocks Serves Total Gender
##    <dbl>       <dbl> <chr>              <chr>   <dbl>  <dbl>  <dbl> <dbl> <chr>
##  1     1          10 Amir Ghafour       IRI       214     21     14   249 M
##  2     2          11 Yuji Nishida       JPN       183     13     23   219 M
##  3     3          18 Ricardo Lucarelli~ BRA       173     14     23   210 M
##  4     4          13 Simon Hirsch       GER       170     21     13   204 M
##  5     5           9 Yoandy Leal Hidal~ BRA       161     22     19   202 M
##  6     6          12 Bruno Lima         ARG       177     16      6   199 M
##  7     7          15 Victor Poletaev    RUS       158     20     16   194 M
##  8     8          14 Yuki Ishikawa      JPN       170     12     11   193 M
##  9     9          18 Lincoln Alexander~ AUS       156     10     12   178 M
## 10    10           2 Chuan Jiang        CHN       152     19      6   177 M
## # ... with 245 more rows
```

Join/Merge

The two datasets were merged using a full join. Both data sets have all of the same variable names so each column name had to be specified to make sure the resulting dataset was tidy. No cases were dropped, but there were issues figuring out how to join by multiple columns for that was not shown in class.

```
# merge the datasets vertically because they have the same variables
scores <- mens %>%
  full_join(womens, by = c("Rank","Name", "ShirtNumber", "Team","Attacks", "Blocks", "Serves", "Total", "Ge
nder"))
scores
```

```
## # A tibble: 500 x 9
##     Rank ShirtNumber Name               Team  Attacks Blocks Serves Total Gender
##    <dbl>       <dbl> <chr>              <chr>   <dbl>  <dbl>  <dbl> <dbl> <chr>
##  1     1          10 Amir Ghafour       IRI       214     21     14   249 M
##  2     2          11 Yuji Nishida       JPN       183     13     23   219 M
##  3     3          18 Ricardo Lucarelli~ BRA       173     14     23   210 M
##  4     4          13 Simon Hirsch       GER       170     21     13   204 M
##  5     5           9 Yoandy Leal Hidal~ BRA       161     22     19   202 M
##  6     6          12 Bruno Lima         ARG       177     16      6   199 M
##  7     7          15 Victor Poletaev    RUS       158     20     16   194 M
##  8     8          14 Yuki Ishikawa      JPN       170     12     11   193 M
##  9     9          18 Lincoln Alexander~ AUS       156     10     12   178 M
## 10    10           2 Chuan Jiang        CHN       152     19      6   177 M
## # ... with 490 more rows
```

Summary Statistics

Dpylr Functions are used first in exploring and modifying the dataset, then additional summary statistics follow with discussion at the end of the section.

```
#install.packages("kableExtra")
library(kableExtra)
```

```
## Warning: package 'kableExtra' was built under R version 3.6.3
```

```
##
## Attaching package: 'kableExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
# Use all 6 dpylr functions while exploring and modifying dataset

# Filter for Team USA because we love our country and count the USA players
scores %>%
  filter(Team == 'USA') %>%
  summarize(count = n())
```

```
## # A tibble: 1 x 1
##   count
##   <int>
## 1    40
```

```
# Group by Team to see distribution of international representation and count how many players on each team there are
scores %>%
  group_by(Team) %>%
  summarize(numPlayers = n()) %>%
  arrange(numPlayers)
```

```
## # A tibble: 22 x 2
##    Team  numPlayers
##    <chr>      <int>
##  1 DOM           11
##  2 POR           12
##  3 AUS           13
##  4 BEL           13
##  5 IRI           13
##  6 KOR           13
##  7 ARG           14
##  8 THA           14
##  9 TUR           14
## 10 CAN           16
## # ... with 12 more rows
```

```
# Use arrange and top_n to find the highest score
scores %>%
  select(Total) %>%
  top_n(1, Total)
```

```
## # A tibble: 1 x 1
##   Total
##   <dbl>
## 1   421
```

```
# Use arrange to find the lowest rank possible
scores %>%
  arrange(desc(Rank))
```

```
## # A tibble: 500 x 9
##     Rank ShirtNumber Name               Team  Attacks Blocks Serves Total Gender
##    <dbl>       <dbl> <chr>              <chr>   <dbl>  <dbl>  <dbl> <dbl> <chr>
##  1   255          11 Aleksa Batak       SRB         0      0      1     1 M
##  2   254           2 Hideomi Fukatsu    JPN         0      0      1     1 M
##  3   253          28 Francesco Recine   ITA         1      0      0     1 M
##  4   252          21 Morteza Sharifi    IRI         1      0      0     1 M
##  5   251           1 Tyler Sanders      CAN         0      0      1     1 M
##  6   250          30 Nikolay Kolev      BUL         1      0      0     1 M
##  7   249          22 Andrija Vilimanov~ SRB         2      0      0     2 M
##  8   248           3 Luca Spirito       ITA         2      0      0     2 M
##  9   247           6 Benjamin Toniutti  FRA         1      1      0     2 M
## 10   246           5 Raphaël Corre      FRA         0      2      0     2 M
## # ... with 490 more rows
```

```
# Use summarize to find the standard deviation of total scores
scores %>%
  summarize(stdDTotal = sd(Total))
```

```
## # A tibble: 1 x 1
##   stdDTotal
##       <dbl>
## 1      59.8
```

```
# Using mutate, create a new variable that determines if a players score is at, above, or below the mean
scores <- scores %>%
  mutate(Avg = case_when(Total < 64.032 ~ 'Below',
                         Total > 64.032 ~ 'Above',
                         Total == 64.032 ~ 'Average'))

# Summary Statistics

# Summary stats for Attacks
scores %>%
  summarize(meanAttacks = mean(Attacks),
            medianAttacks = median(Attacks),
            minAttacks = min(Attacks),
            maxAttacks = max(Attacks),
            stdvAttacks = sd(Attacks))
```

```
## # A tibble: 1 x 5
##   meanAttacks medianAttacks minAttacks maxAttacks stdvAttacks
##         <dbl>         <dbl>      <dbl>      <dbl>       <dbl>
## 1        49.9            34          0        376        51.6
```

```
# Summary stats for Blocks
scores %>%
  summarize(meanBlocks = mean(Blocks),
            medianBlocks = median(Blocks),
            minBlocks = min(Blocks),
            maxBlocks = max(Blocks),
            stdvBlocks = sd(Blocks))
```

```
## # A tibble: 1 x 5
##   meanBlocks medianBlocks minBlocks maxBlocks stdvBlocks
##        <dbl>        <dbl>     <dbl>     <dbl>      <dbl>
## 1       8.83            6         0        51       8.66
```

```
# Summary stats for Serves
scores %>%
  summarize(meanServes = mean(Serves),
            medianServes = median(Serves),
            minServes = min(Serves),
            maxServes = max(Serves),
            stdvServes = sd(Serves))
```

```
## # A tibble: 1 x 5
##   meanServes medianServes minServes maxServes stdvServes
##        <dbl>        <dbl>     <dbl>     <dbl>      <dbl>
## 1       5.30            4         0        28       5.05
```

```
# Summary stats for Totals
scores %>%
  summarize(meanTotals = mean(Total),
            medianTotals = median(Total),
            minTotals = min(Total),
            maxTotals = max(Total),
            stdvTotals = sd(Total))
```

```
## # A tibble: 1 x 5
##   meanTotals medianTotals minTotals maxTotals stdvTotals
##        <dbl>        <dbl>     <dbl>     <dbl>      <dbl>
## 1       64.0         47.5         1       421       59.8
```

```
# Summary stats by gender
scores %>%
  group_by(Gender) %>%
  summarize(meanTotals = mean(Total),
            medianTotals = median(Total),
            minTotals = min(Total),
            maxTotals = max(Total),
            stdvTotals = sd(Total))
```

```
## # A tibble: 2 x 6
##   Gender meanTotals medianTotals minTotals maxTotals stdvTotals
## * <chr>       <dbl>        <dbl>     <dbl>     <dbl>      <dbl>
## 1 M            62.8           48         1       249       53.9
## 2 W            65.3           47         1       421       65.4
```

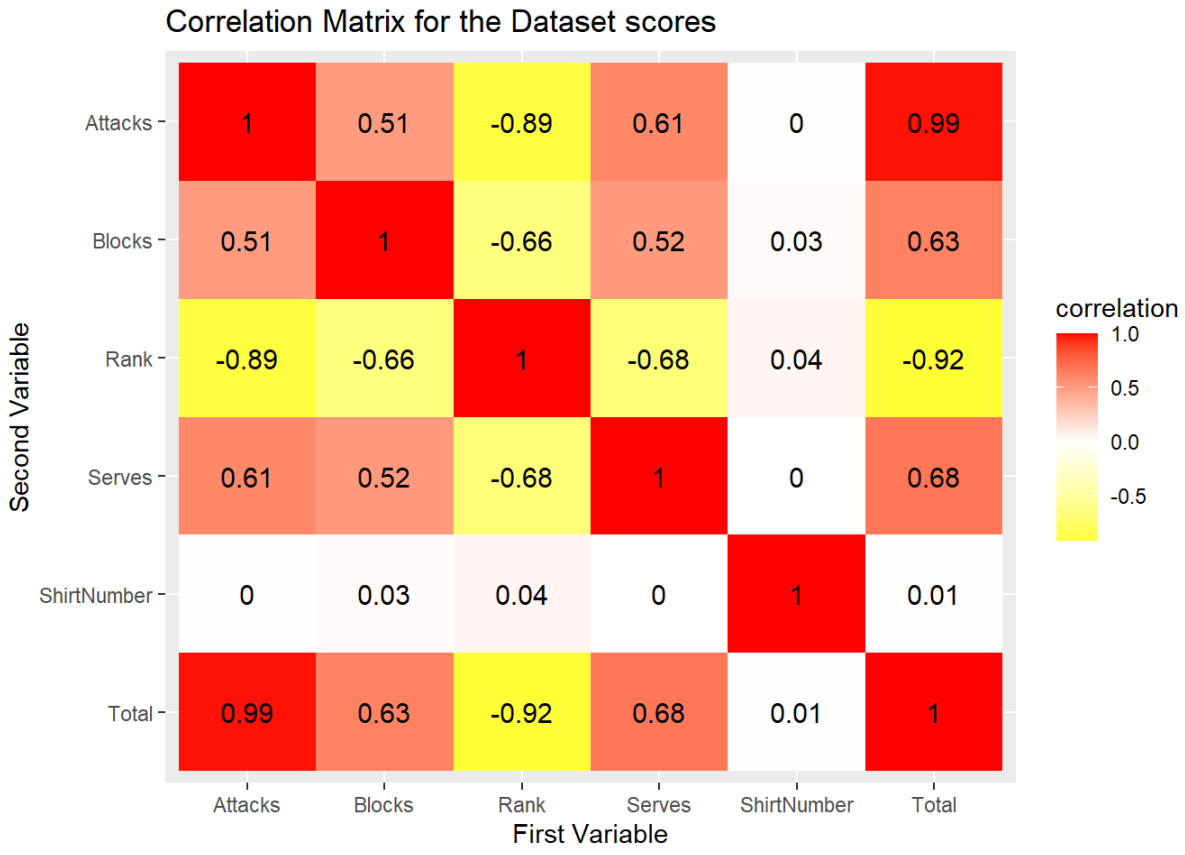
```
# Pretty Table using Kable
sumTable <- matrix(c(mean(scores$Attacks), median(scores$Attacks),min(scores$Attacks),
                     max(scores$Attacks),sd(scores$Attacks),mean(scores$Blocks),
                     median(scores$Blocks),min(scores$Blocks),max(scores$Blocks),
                     sd(scores$Blocks),mean(scores$Serves),
                   median(scores$Serves),min(scores$Serves),max(scores$Serves),
                   sd(scores$Serves),mean(scores$Total),median(scores$Total),
                   min(scores$Total),max(scores$Total),sd(scores$Total)), ncol = 5, byrow = T)

# Name columns according to summary statistic
colnames(sumTable) <- c("Mean", "Median", "Min", "Max","StdDev")
# Name row according to variable name
rownames(sumTable) <- c("Attacks", "Blocks", "Serves", "Total")
# amke table using kable package
sumTable %>%
  kbl() %>%
  kable_styling()
```

	Mean	Median	Min	Max	StdDev
Attacks	49.910	34.0	0	376	51.550332
Blocks	8.826	6.0	0	51	8.663362
Serves	5.296	4.0	0	28	5.048898
Total	64.032	47.5	1	421	59.795200

First, each of the scoring variables were averaged. The mean number of attacks was 49.91 attacks. The mean number of blocks was 8.826 blocks. The mean number of serves was 5.296 serves. The mean total score was 64.032 points. The USA has 40 players in this dataset. The DOM team had the least number of players with 11 players. The highest score is 421 total points. The lowest rank possible is number 255. The standard deviation of total scores is 59.7952. No player has exactly the mean total points foudn from the last table.

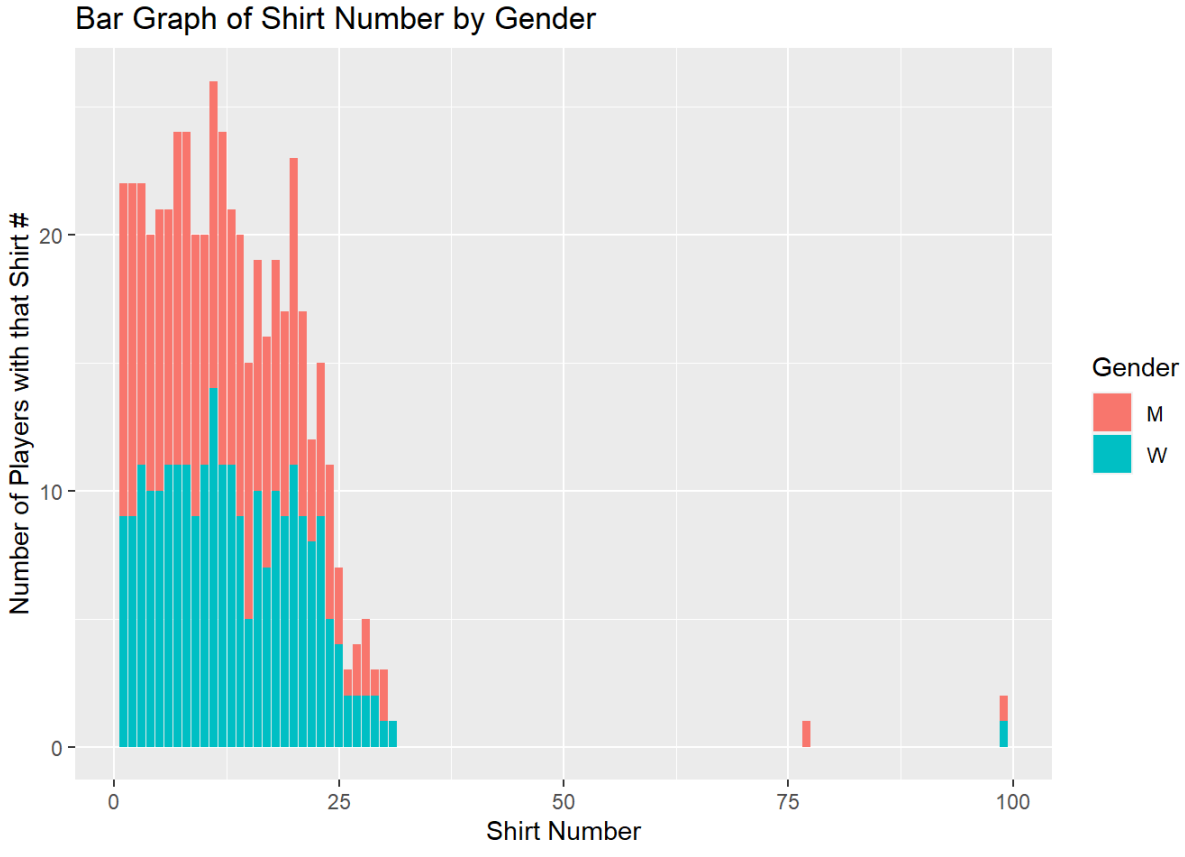
Visualizations

Visualization 1: Correlation Matrix



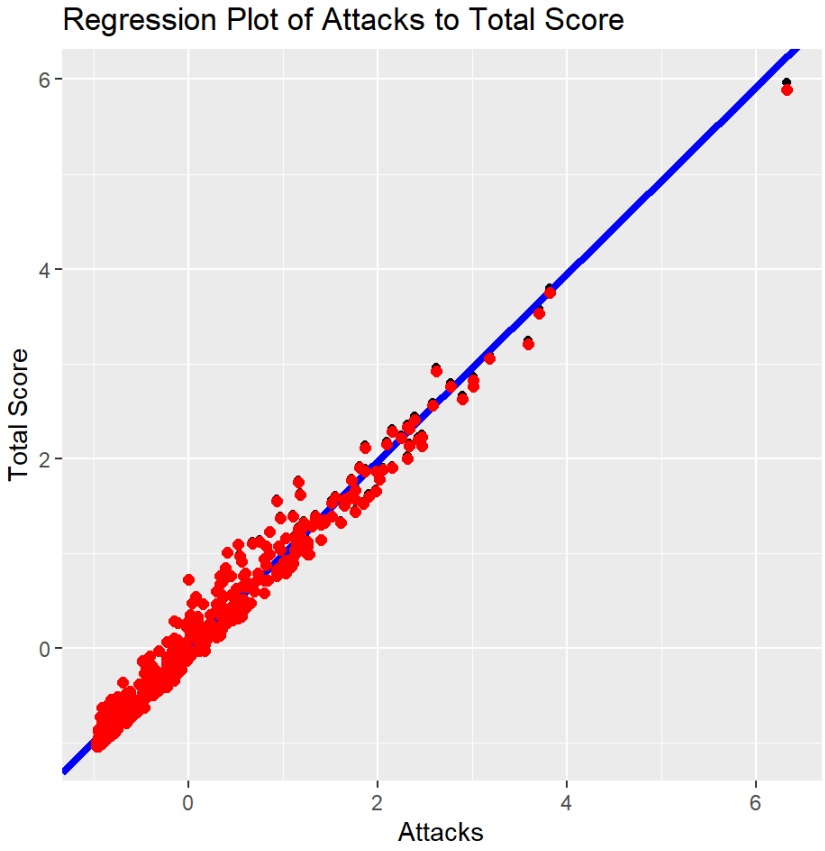
There is a clear 1 to 1 correlation between each variable and itself. The Attacks and Total variables are very closely correlated with a value of 0.99. Other honorable mentions include the 0.68 correlation value between Serves and Total points. Shirt Number has next to no correlation with any other numeric value, but this is expected.

Visualization 2: Bar Plot



This bar graph shows the display of Shirt Number by Gender. This display shows that men and women have a pretty even mix of picking the same shirt numbers. Even number 99 has one girl and one boy.

Visualization 3: Regression Line



This regression line shows the great correlation between Attacks and Total Score. These two variables were shown to be closely correlated in the correlation matrix and this regression line is an addition display how significant the correlation is.

Dimensionality Reduction

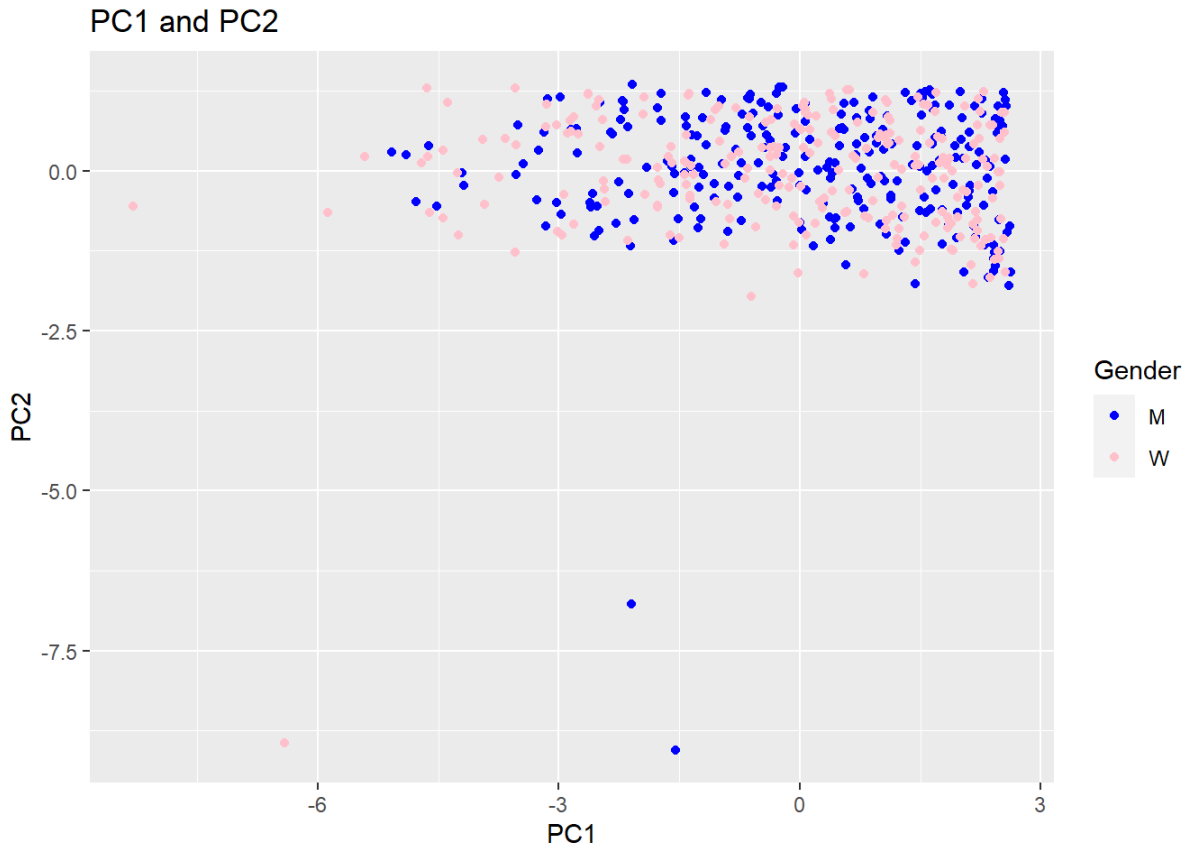
PCA

```
## [1] "sdev"     "rotation" "center"   "scale"    "x"
```

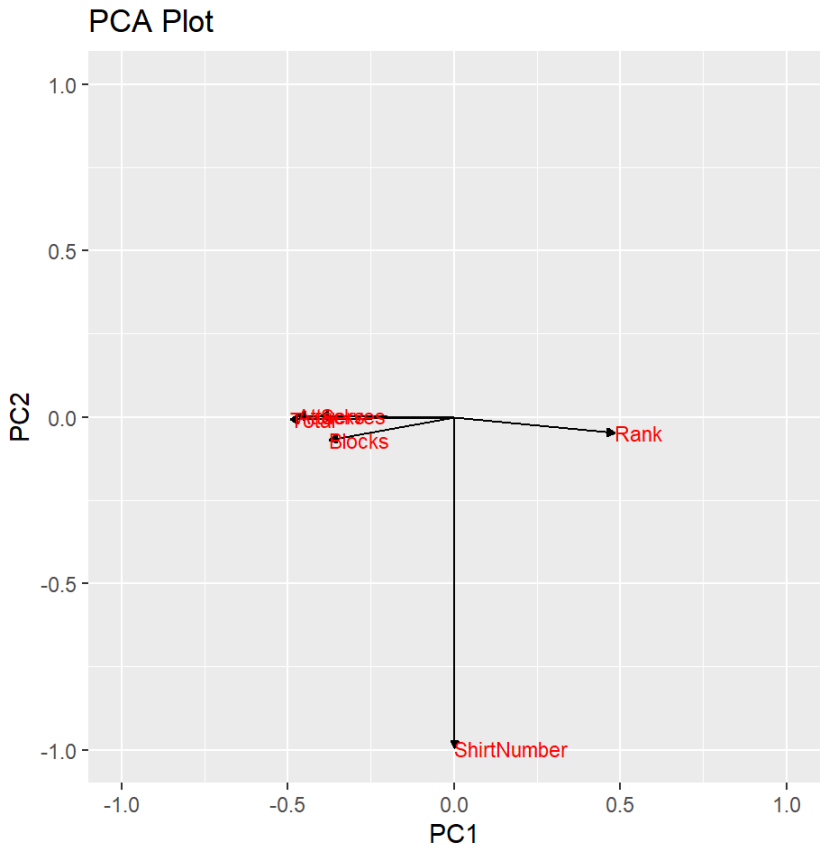
```
## Standard deviations (1, .., p=6):
## [1] 1.967322e+00 1.002205e+00 7.514636e-01 6.732528e-01 3.275122e-01
## [6] 9.753954e-16
##
## Rotation (n x k) = (6 x 6):
##                      PC1          PC2         PC3         PC4         PC5
## Rank         0.483792367 -0.048327475  0.10304148  0.12037062 -0.85936184
## ShirtNumber  0.001936035 -0.996469368 -0.06153794  0.02171028  0.05279011
## Attacks     -0.470462572  0.003895711 -0.44587888 -0.20031484 -0.34659499
## Blocks      -0.375055756 -0.068160370  0.83326704 -0.35205775 -0.15671098
## Serves      -0.400218214  0.003708914  0.17460870  0.89386669 -0.07937813
## Total       -0.493725357 -0.006203622 -0.24892850 -0.14822693 -0.32821202
##                      PC6
## Rank         5.987038e-16
## ShirtNumber  9.153110e-17
## Attacks     -6.477559e-01
## Blocks      -1.088595e-01
## Serves      -6.344194e-02
## Total        7.513568e-01
```

```
##            PC1         PC2         PC3        PC4        PC5           PC6
## [1,] -5.075485  0.28804513 -0.87621077 -0.2637308 -1.0116754 -1.776357e-15
## [2,] -4.905050  0.25390627 -0.94638560  1.8535002 -0.6417494 -1.998401e-15
## [3,] -4.774667 -0.48100303 -0.76968591  1.8915248 -0.5165946 -1.776357e-15
## [4,] -4.202426 -0.02472969 -0.35784091 -0.1464692 -0.4722694 -1.776357e-15
## [5,] -4.616797  0.38584170  0.05907771  0.9076939 -0.5470771 -1.776357e-15
## [6,] -3.440510  0.11297086 -1.11131148 -1.1963142 -0.3206528 -1.998401e-15
```

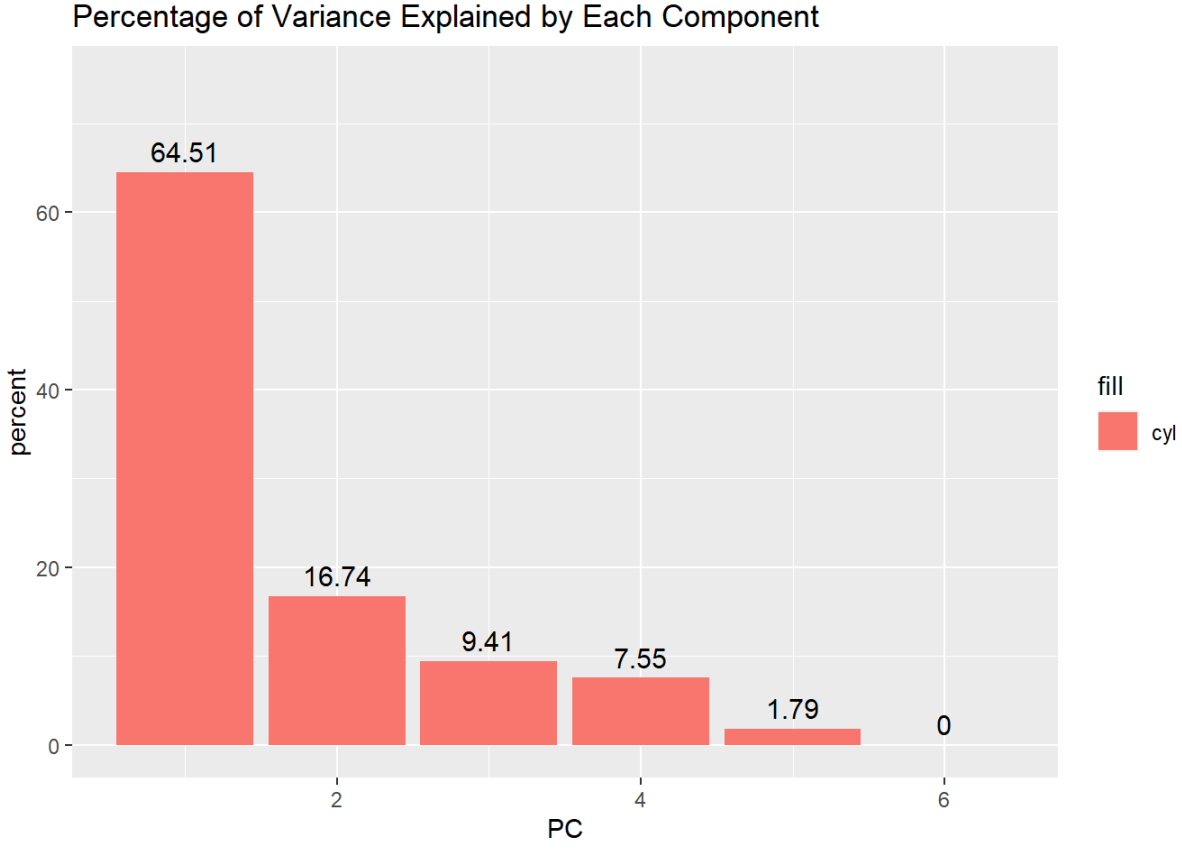
```
##         PC1         PC2         PC3        PC4        PC5           PC6 Gender
## 1 -5.075485  0.28804513 -0.87621077 -0.2637308 -1.0116754 -1.776357e-15      M
## 2 -4.905050  0.25390627 -0.94638560  1.8535002 -0.6417494 -1.998401e-15      M
## 3 -4.774667 -0.48100303 -0.76968591  1.8915248 -0.5165946 -1.776357e-15      M
## 4 -4.202426 -0.02472969 -0.35784091 -0.1464692 -0.4722694 -1.776357e-15      M
## 5 -4.616797  0.38584170  0.05907771  0.9076939 -0.5470771 -1.776357e-15      M
## 6 -3.440510  0.11297086 -1.11131148 -1.1963142 -0.3206528 -1.998401e-15      M
```




```
##                    PC1          PC2         PC3         PC4         PC5
## Rank         0.483792367 -0.048327475  0.10304148  0.12037062 -0.85936184
## ShirtNumber  0.001936035 -0.996469368 -0.06153794  0.02171028  0.05279011
## Attacks     -0.470462572  0.003895711 -0.44587888 -0.20031484 -0.34659499
## Blocks      -0.375055756 -0.068160370  0.83326704 -0.35205775 -0.15671098
## Serves      -0.400218214  0.003708914  0.17460870  0.89386669 -0.07937813
## Total       -0.493725357 -0.006203622 -0.24892850 -0.14822693 -0.32821202
##                     PC6
## Rank         5.987038e-16
## ShirtNumber  9.153110e-17
## Attacks     -6.477559e-01
## Blocks      -1.088595e-01
## Serves      -6.344194e-02
## Total        7.513568e-01
```




```
## [1] 6.450591e+01 1.674024e+01 9.411625e+00 7.554489e+00 1.787738e+00
## [6] 1.585660e-29
```



The standard deviations on each PC were: 1.967, 1.002, 0.751, 0.673, 0.327, 0.0000 eith respect to PC1, PC2, PC3, PC4, PC5, and PC6. The rotation matrix (orthogonal transformation) was then made to uncorrelate values in an effort to maximize variance. This rotated data was then plotted in the scatterplot above. While the variance was maximized, there was still not a lot of clustering between genders. As shown in the PCA Plot, Blocks, Attacks, and Total contributed greatly to the result while Rank did not as much and ShirtNumber did not really matter. Finally, PCA1 proved to explain the 64.51% of the variance while PC5 and PCA6 did not contribute at all.

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