R Cheatsheet

Kate's personal log of interesting and useful R snippets

Last updated: 23Feb16

Data Wrangling

Data Input Basics

Creating a simple dataframe for example purposes

```
Location <- c("New York", "London", "new york", "London", "London")
Product <- c("Tea", "Coffee", "Coffee", NA, "Coffee")
Revenue <- c("USD 10000", "5000", "6,500", "$7850", "300")
YearFounded <- c(1985, 2001, NA, NA, 1996)
SampleSet <- data.frame(Location, Product, Revenue, YearFounded, stringsAsFactors=FALSE)
SampleSet
```

```
##   Location Product   Revenue YearFounded
## 1 New York     Tea USD 10000        1985
## 2   London  Coffee      5000        2001
## 3 new york  Coffee     6,500          NA
## 4   London    <NA>     $7850          NA
## 5   London  Coffee       300        1996
```

Saving a copy of the dataframe as .csv (comma-separated) data

```
write.csv(SampleSet, "SampleSet.csv")
```

Reading in .csv (comma-separated) data

```
NewSampleSet <- read.csv("SampleSet.csv")

NewSampleSet
```

```
##   X Location Product   Revenue YearFounded
## 1 1 New York     Tea USD 10000        1985
## 2 2   London  Coffee      5000        2001
## 3 3 new york  Coffee     6,500          NA
## 4 4   London    <NA>     $7850          NA
## 5 5   London  Coffee       300        1996
```

Reading in .tsv (tab-separated) data

```
Football <- read.table("Football.tsv", header=TRUE, sep = "\t")
```

Looking at Basic Structure

Show the dimensions of the dataframe

```
dim(SampleSet)
```

```
## [1] 5 4
```

Show the top few rows of content in a dataframe

```
head(Football)
```

```
##                   X  Y    X5    X7  X11 X15  X18 X23 Tiers
## 1 ManchesterUnited 87 14.63 17.50 6.21 930 0.58  51 Tier1
## 2          Chelsea 85 14.29 18.46 7.01 937 0.68  63 Tier1
## 3          Arsenal 83 15.64 10.81 8.68 883 0.82  55 Tier1
## 4        Liverpool 76 12.52 19.40 4.93 922 0.74  45 Tier1
## 5          Everton 65 15.24  9.09 4.79 868 0.87  40 Tier2
## 6       AstonVilla 60 17.53 16.90 4.77 926 1.34  54 Tier2
```

Show a simple summary of each field in a dataframe

```
summary(Football)
```

```
##               X            Y               X5              X7
##  Arsenal       : 1   Min.   :11.00   Min.   : 6.94   Min.   : 9.09
##  AstonVilla    : 1   1st Qu.:38.50   1st Qu.:11.41   1st Qu.:11.70
##  BirminghamCity: 1   Median :47.50   Median :12.54   Median :16.34
##  Blackburn     : 1   Mean   :52.00   Mean   :13.10   Mean   :15.51
##  Bolton        : 1   3rd Qu.:61.25   3rd Qu.:14.80   3rd Qu.:18.70
##  Chelsea       : 1   Max.   :87.00   Max.   :19.94   Max.   :21.74
##  (Other)       :14
##       X11              X15              X18             X23
##  Min.   :3.740   Min.   : 654.0   Min.   :0.580   Min.   :40.00
##  1st Qu.:4.435   1st Qu.: 831.0   1st Qu.:1.005   1st Qu.:53.25
##  Median :4.765   Median : 892.0   Median :1.365   Median :59.00
##  Mean   :5.037   Mean   : 875.0   Mean   :1.318   Mean   :59.75
##  3rd Qu.:5.077   3rd Qu.: 931.8   3rd Qu.:1.587   3rd Qu.:63.50
##  Max.   :8.680   Max.   :1072.0   Max.   :2.340   Max.   :86.00
##
##    Tiers
##  Tier1: 4
##  Tier2:12
##  Tier3: 4
##
##
##
##
```

See the distributions of values for a certain field

```
table(SampleSet$Location)
```

```
##
##   London new york New York
##        3        1        1
```

List all unique values for a certain field

```
unique(SampleSet$Location)
```

```
## [1] "New York" "London"   "new york"
```

Column Headings

Listing all column headings

```
names(SampleSet)
```

```
## [1] "Location"    "Product"     "Revenue"     "YearFounded"
```

Substituting new column names using rename()

```
#Creating a vector of new column names (in any order), with old column names as the labels
FootballNames <- c("Y"= "Points",
           "X5" = "Shots",
           "X7" = "Box",
           "X11" = "Passes",
           "X15" = "Crosses",
           "X" = "Team",
           "X18" = "GoalsConceded",
           "X23" = "YellowCards")
FootballNames
```

```
##               Y              X5              X7             X11
##        "Points"         "Shots"           "Box"        "Passes"
##             X15               X             X18             X23
##       "Crosses"          "Team" "GoalsConceded"   "YellowCards"
```

```
#Load the 'plyr' package
library(plyr)
#Replace existing column names using plyr package
Football <- rename(Football, FootballNames)
#Show top lines of data to verify it has been read correctly
head(Football)
```

```
##               Team Points Shots   Box Passes Crosses GoalsConceded
## 1 ManchesterUnited     87 14.63 17.50   6.21     930          0.58
## 2          Chelsea     85 14.29 18.46   7.01     937          0.68
## 3          Arsenal     83 15.64 10.81   8.68     883          0.82
## 4        Liverpool     76 12.52 19.40   4.93     922          0.74
## 5          Everton     65 15.24  9.09   4.79     868          0.87
## 6       AstonVilla     60 17.53 16.90   4.77     926          1.34
##   YellowCards Tiers
## 1          51 Tier1
## 2          63 Tier1
## 3          55 Tier1
## 4          45 Tier1
## 5          40 Tier2
## 6          54 Tier2
```

Understanding structure of individual columns

See now many non-NA values are contained within a column

```
length(which(!is.na(SampleSet$Product)))
```

```
## [1] 4
```

Cleaning up content

Duplicating a field in the dataset (but giving it a new name)

```
SampleSet$OtherLocation <- SampleSet$Location
SampleSet
```

```
##   Location Product   Revenue YearFounded OtherLocation
## 1 New York     Tea USD 10000        1985      New York
## 2   London  Coffee      5000        2001        London
## 3 new york  Coffee     6,500          NA      new york
## 4   London    <NA>     $7850          NA        London
## 5   London  Coffee       300        1996        London
```

Delete a field from the dataset

```
SampleSet$OtherLocation <- NULL
SampleSet
```

```
##   Location Product   Revenue YearFounded
## 1 New York     Tea USD 10000        1985
## 2   London  Coffee      5000        2001
## 3 new york  Coffee     6,500          NA
## 4   London    <NA>     $7850          NA
## 5   London  Coffee       300        1996
```

Convert all contents of a column to proper case

```
SampleSet$Location
```

```
## [1] "New York" "London"   "new york" "London"   "London"
```

```
#Add a function to make proper case
properCase <- function(x) {
  lower <- tolower(x)
  s <- strsplit(lower, " ")[[1]]
  paste(toupper(substring(s, 1,1)), substring(s, 2),
      sep="", collapse=" ")
}
#Apply this function to a column
SampleSet$LocationProper <- sapply(SampleSet$Location, properCase)

SampleSet$LocationProper
```

```
## [1] "New York" "London"   "New York" "London"   "London"
```

Extract the digits in a column and make column into an integer column (by replacing all non-digit characters with blank space)

```
SampleSet$Revenue <- as.integer(gsub("[^0-9]", "", SampleSet$Revenue))
SampleSet
```

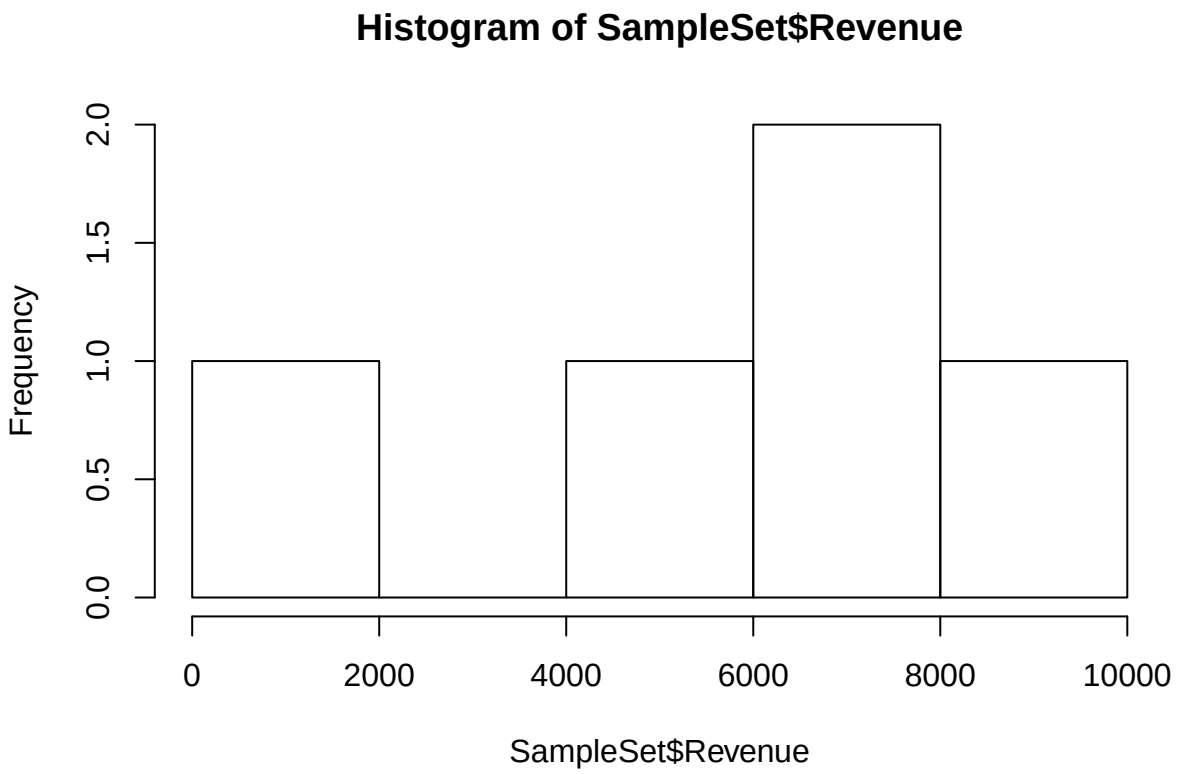
```
##   Location Product Revenue YearFounded LocationProper
## 1 New York     Tea   10000        1985       New York
## 2   London  Coffee    5000        2001         London
## 3 new york  Coffee    6500          NA       New York
## 4   London    <NA>    7850          NA         London
## 5   London  Coffee     300        1996         London
```

Plots

Simple Plots in Base R

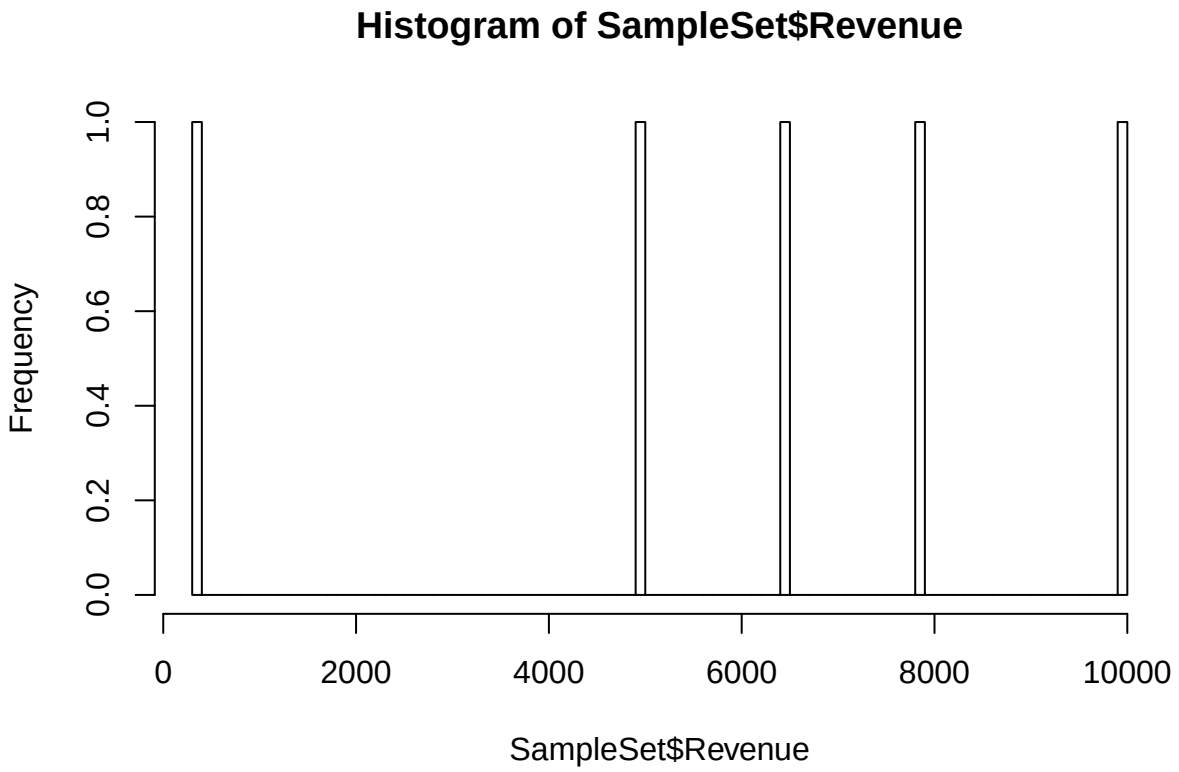
Show a histogram of the distribution of values

```
hist(SampleSet$Revenue)
```



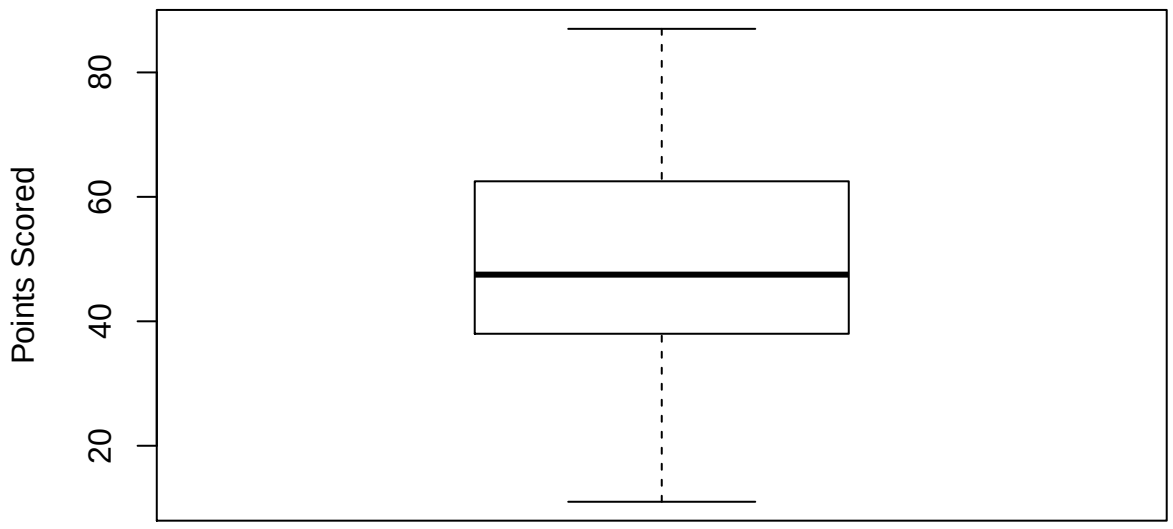
Same as previous, but with a lot more bins to the histogram

```
hist(SampleSet$Revenue, breaks=100)
```



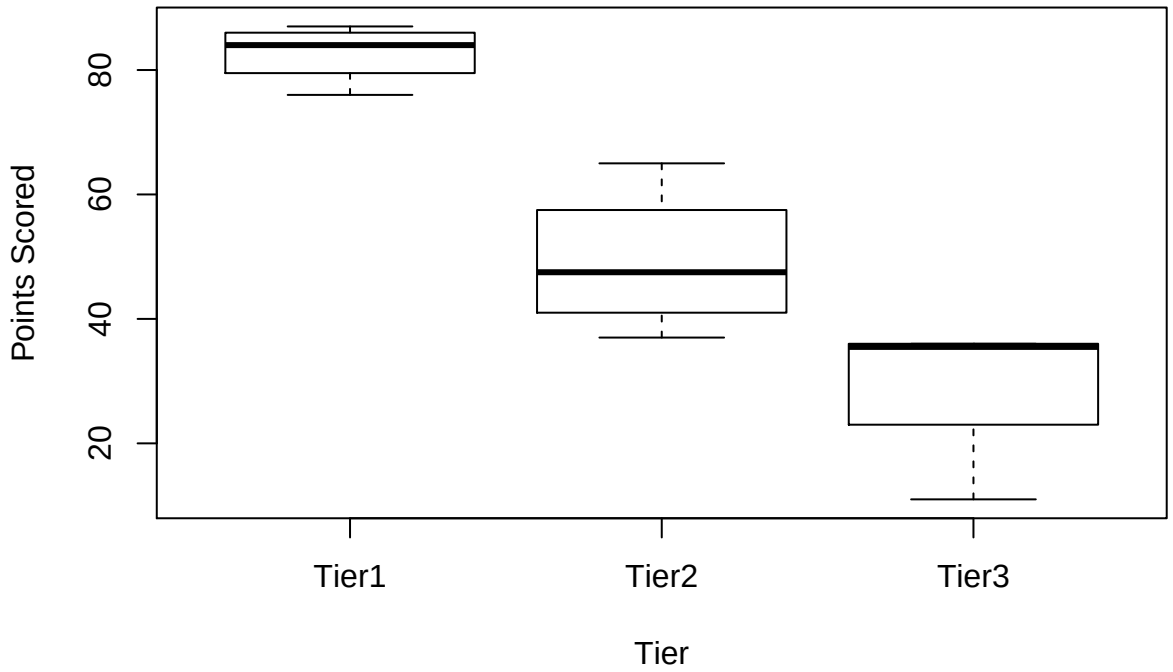
Boxplot of a variable

```
boxplot(Football$Points, ylab = "Points Scored")
```



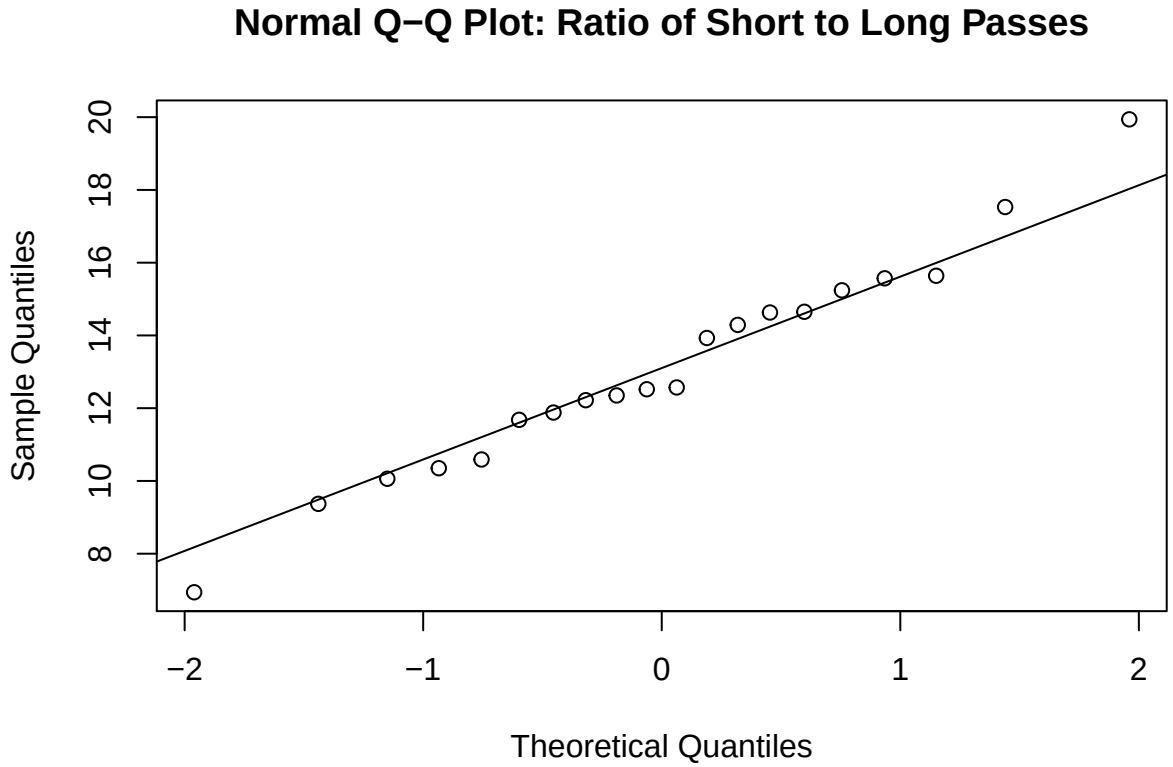
Boxplot of a variable, separated by the factor categories of another variable

```
boxplot(Football$Points ~ Football$Tiers, ylab = "Points Scored", xlab = "Tier")
```



Plot a QQ plot to see if a field is distributed normally

```
qqnorm(Football$Shots, main = "Normal Q-Q Plot: Ratio of Short to Long Passes")
qqline(Football$Shots)
```

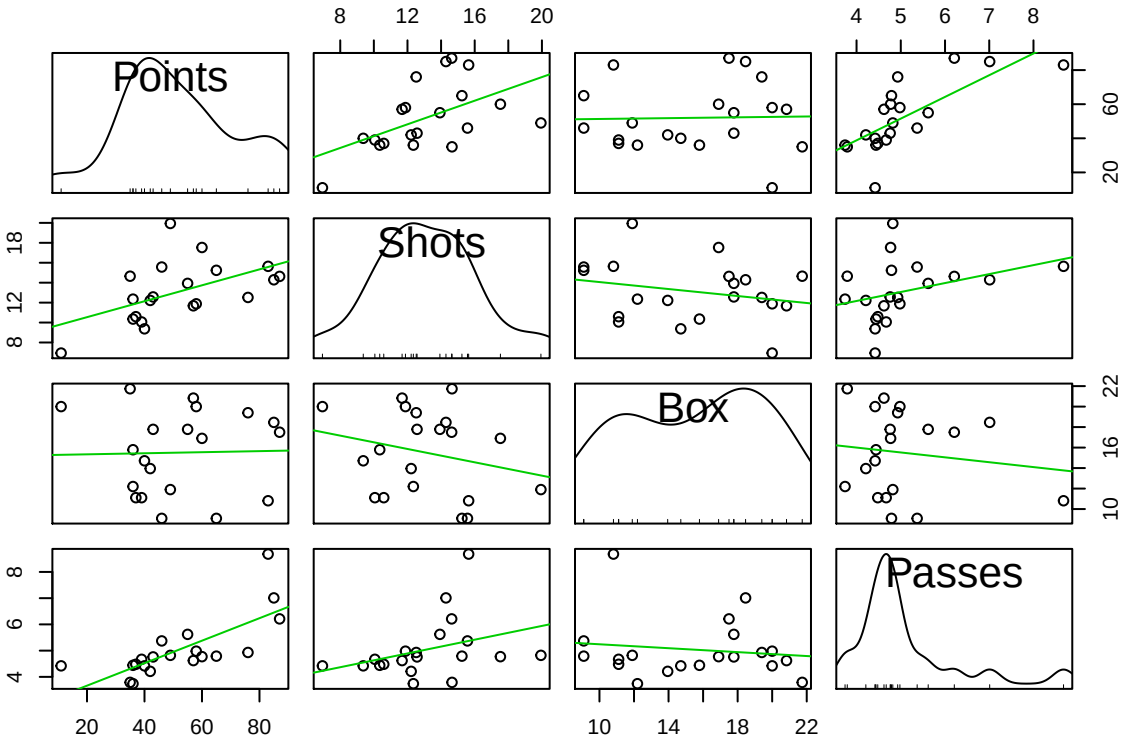


Plots with ggplot2

Fancier Plots from Other Packages

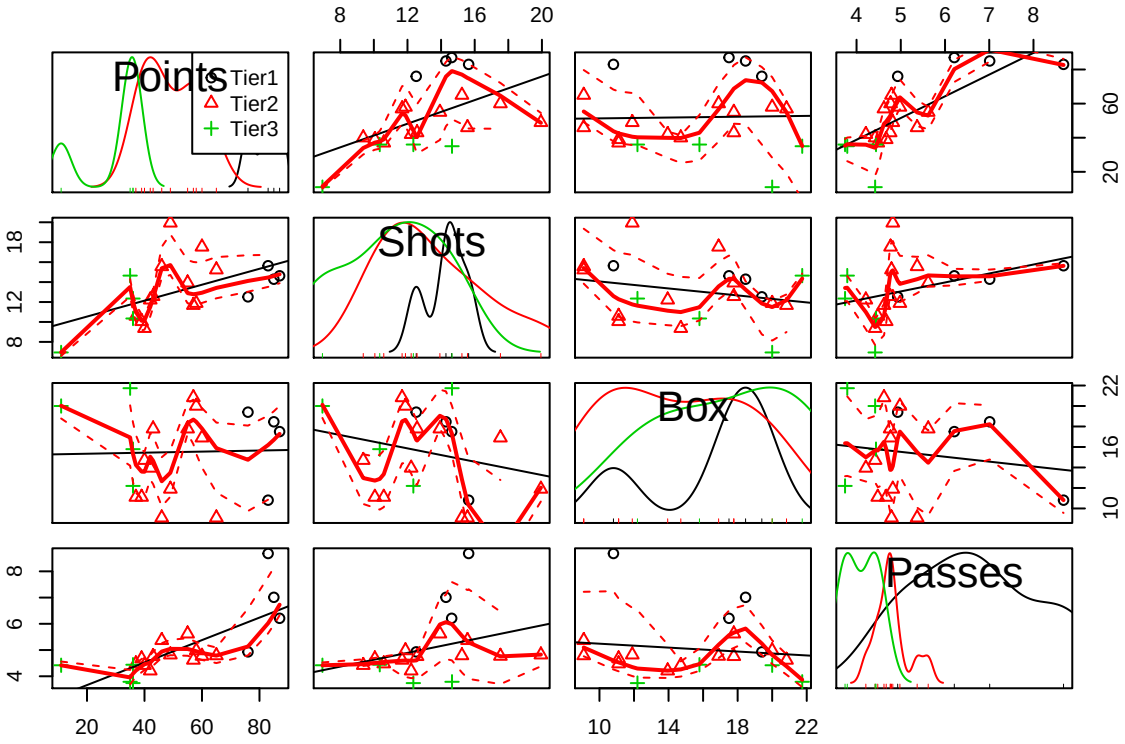
A pretty scatterplot matrix using the *cars* package

```
library(car)
spm(~ Points + Shots + Box + Passes, data = Football, smoother = FALSE)
```



Same as above but where factors from another fields are used to colour-code the dots, and with smoothed trend lines

```
library(car)
spm(~ Points + Shots + Box + Passes | Tiers, data = Football)
```



Statistical Tools

Statistical Tests and ANOVA

Shapiro-Wilk test for normal distribution

```
shapiro.test(Football$Passes)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Football$Passes
## W = 0.78868, p-value = 0.0005886
```

ANOVA One-Way test on a variable across categories (described by a factor fariable)

```
oneway.test(Football$Points ~ Football$Tiers)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  Football$Points and Football$Tiers
## F = 55.374, num df = 2.0000, denom df = 6.7904, p-value =
## 6.248e-05
```

Correlation

Output a Pearson correlation matrix (after reducing the default number of digits shown)

```
options(digits = 3)
cor(Football[,2:5])
```

```
##        Points  Shots     Box Passes
## Points 1.0000  0.526  0.0251  0.739
## Shots  0.5260  1.000 -0.2388  0.343
## Box    0.0251 -0.239  1.0000 -0.133
## Passes 0.7388  0.343 -0.1333  1.000
```

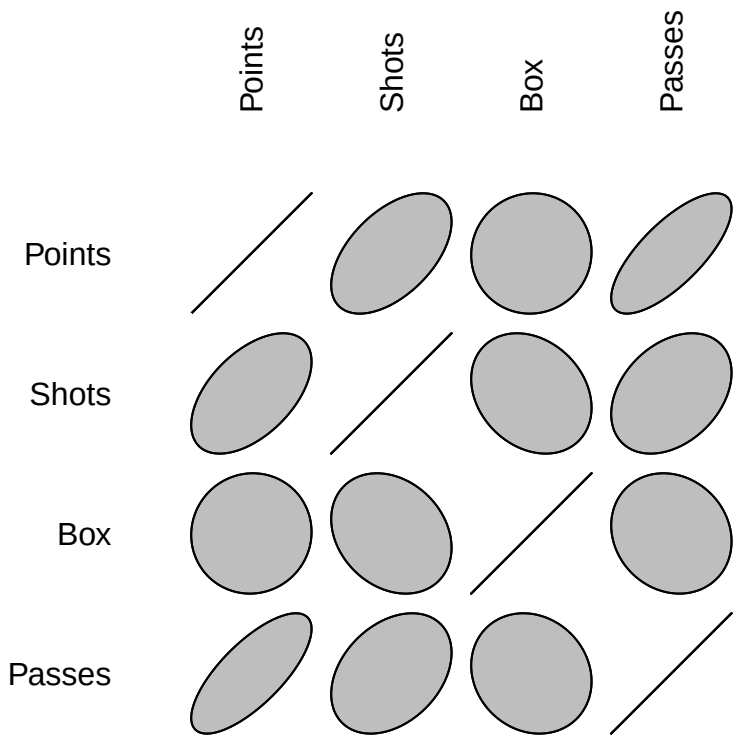
```
#Reset digits options back to original
options(digits = 7)
```

Plot a visual form of the Pearson correlation matrix using the *ellipse* package

```
library(ellipse)
```

```
##
## Attaching package: 'ellipse'
##
## The following object is masked from 'package:car':
##
##     ellipse
```

```
CorMatrix <- cor(Football[,2:5])
#Make the plot but modify the margins slightly
plotcorr(CorMatrix, mar = c(0.1, 0.1, 0.5, 0.1))
```



Test the correlation between two fields (Null hypothesis: no correlation)

```
cor.test(Football$Points, Football$Passes)
```

```
##
##  Pearson's product-moment correlation
##
## data:  Football$Points and Football$Passes
## t = 4.6508, df = 18, p-value = 0.0001988
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4401623 0.8902551
## sample estimates:
##       cor
## 0.7387823
```

Linear Regression

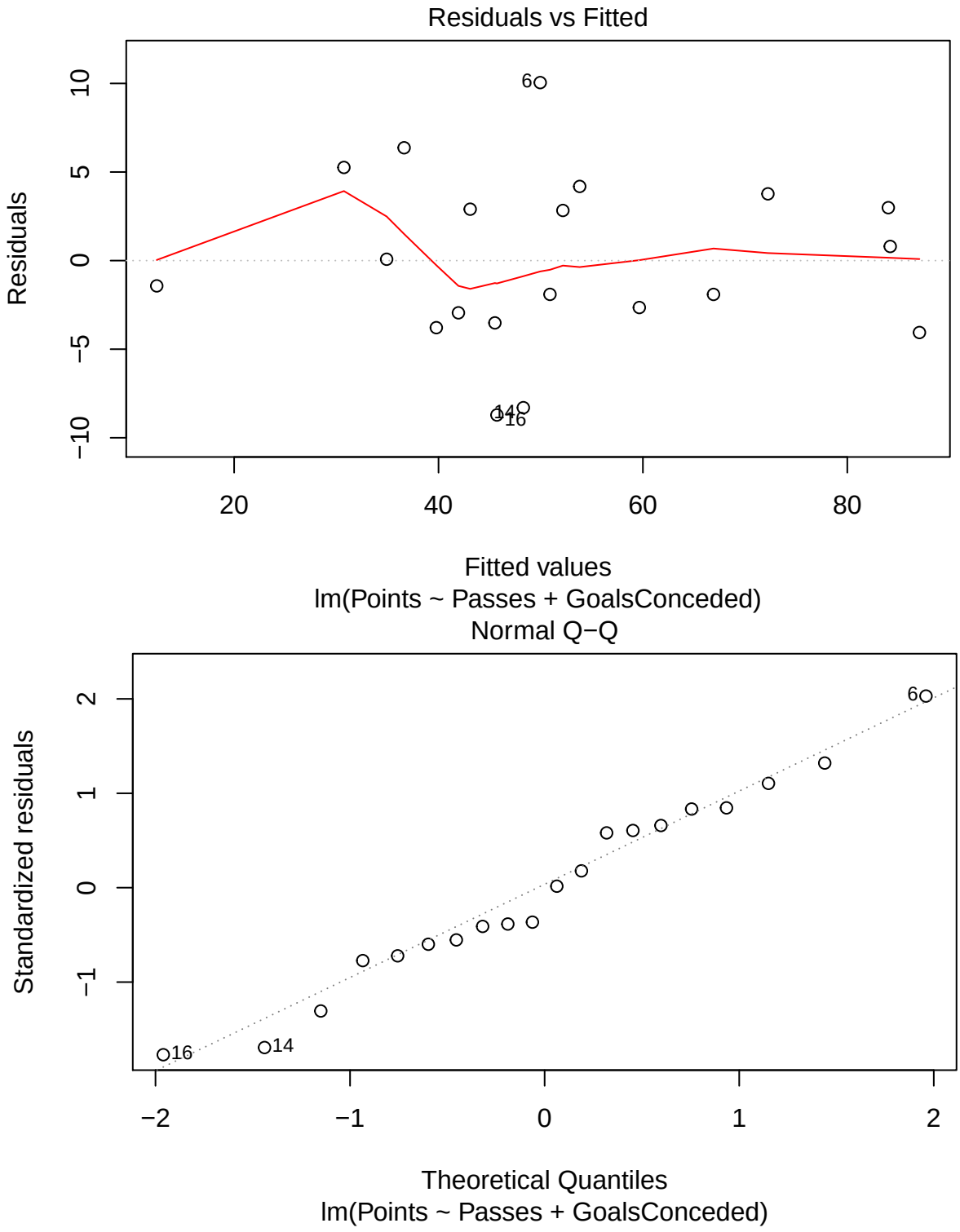
Building a linear regression model and summarising it

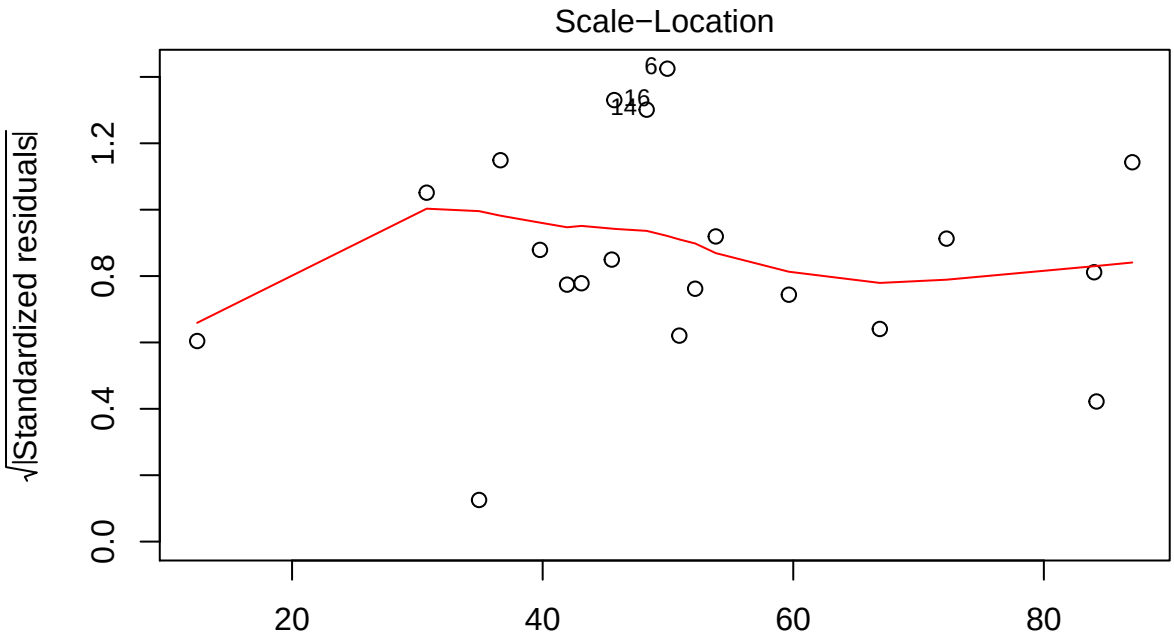
```
LinearReg <- lm(Points ~ Passes + GoalsConceded, data=Football)
summary(LinearReg)
```

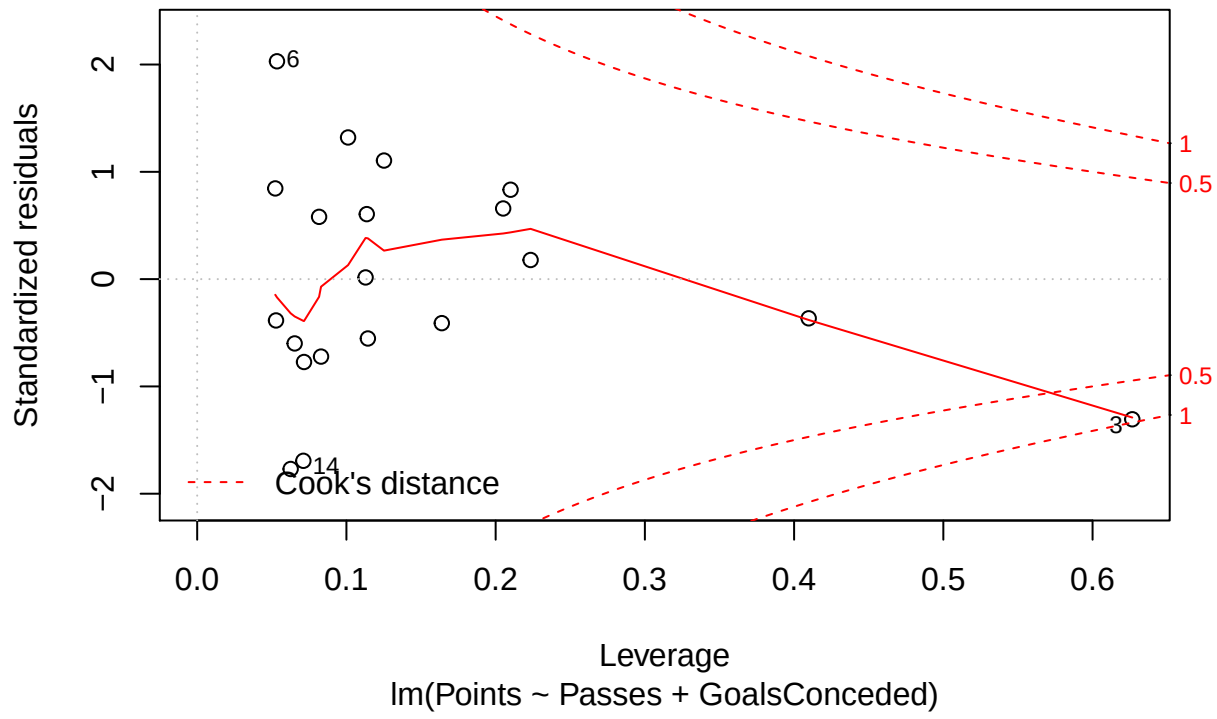
```
##
## Call:
## lm(formula = Points ~ Passes + GoalsConceded, data = Football)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.7146 -3.0898 -0.6759  3.1819 10.0469
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     75.514      9.781   7.721 5.90e-07 ***
## Passes           4.719      1.268   3.723  0.00169 **
## GoalsConceded  -35.874      3.376 -10.625 6.32e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.087 on 17 degrees of freedom
## Multiple R-squared:  0.9406, Adjusted R-squared:  0.9336
## F-statistic: 134.5 on 2 and 17 DF,  p-value: 3.801e-11
```

Base R functionality to plot regression results

```
plot(LinearReg)
```


Residuals vs Fitted
Residuals
10
5
0
−5
−10
6
14
16
20
40
60
80
Fitted values
lm(Points ~ Passes + GoalsConceded)
Normal Q−Q
Standardized residuals
2
1
0
−1
6
16
14
−2
−1
0
1
2
Theoretical Quantiles
lm(Points ~ Passes + GoalsConceded)

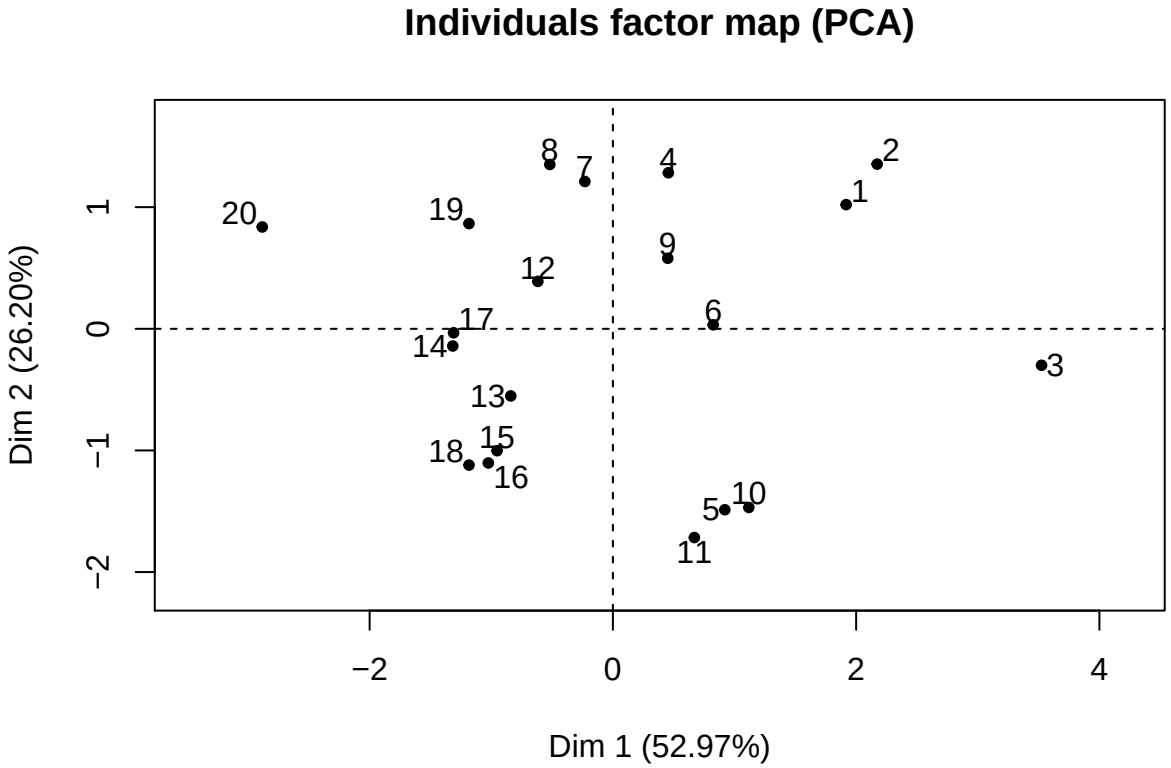


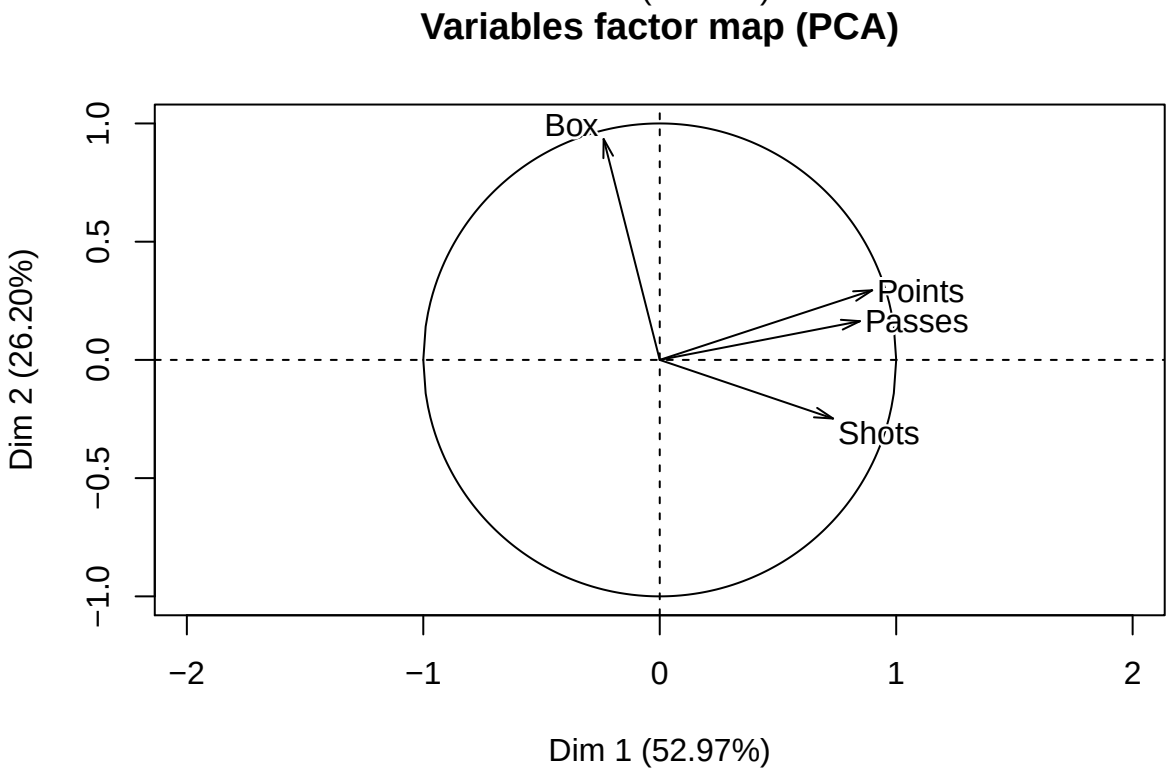


Principal Component Analysis

With the *FactoMineR* package, perform PCA and plot the cumulative percentage of variance gained with each eigenvector

```
library(FactoMineR)
FootballPC <- PCA(Football[,2:5])
```



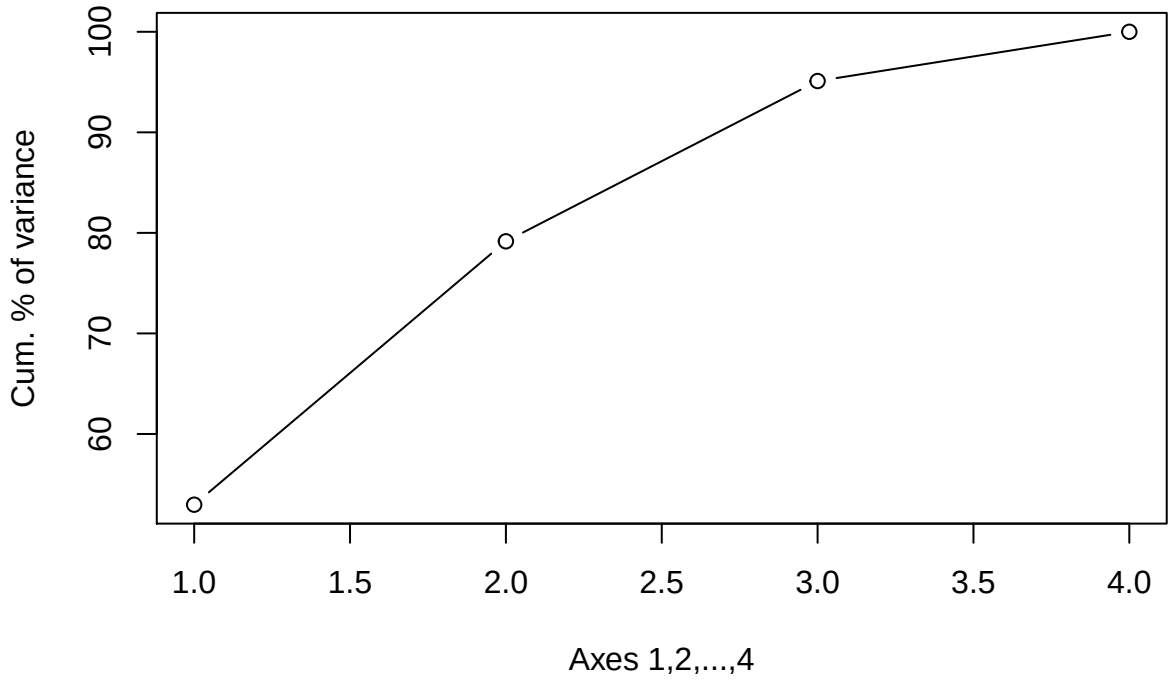


```r
summary(FootballPC)
```

```
##
## Call:
## PCA(X = Football[, 2:5])
##
##
## Eigenvalues
##                        Dim.1   Dim.2   Dim.3   Dim.4
## Variance               2.119   1.048   0.638   0.196
## % of var.             52.966  26.197  15.939   4.898
## Cumulative % of var.  52.966  79.163  95.102 100.000
##
## Individuals (the 10 first)
##            Dist    Dim.1    ctr   cos2    Dim.2    ctr   cos2    Dim.3
## 1      |  2.223 |  1.918  8.680  0.744 |  1.021  4.973  0.211 | -0.076
## 2      |  2.605 |  2.173 11.141  0.696 |  1.354  8.744  0.270 | -0.478
## 3      |  3.924 |  3.524 29.315  0.807 | -0.300  0.429  0.006 | -1.533
## 4      |  1.599 |  0.456  0.491  0.081 |  1.283  7.850  0.643 |  0.159
## 5      |  1.906 |  0.920  2.000  0.233 | -1.488 10.560  0.609 |  0.160
## 6      |  1.629 |  0.825  1.606  0.257 |  0.033  0.005  0.000 |  1.402
## 7      |  1.242 | -0.230  0.125  0.034 |  1.211  6.998  0.950 |  0.046
## 8      |  1.492 | -0.519  0.635  0.121 |  1.351  8.709  0.820 |  0.236
## 9      |  0.839 |  0.451  0.480  0.289 |  0.580  1.604  0.478 |  0.109
## 10     |  2.522 |  1.118  2.948  0.196 | -1.469 10.297  0.339 |  1.646
##            ctr   cos2
## 1        0.046  0.001 |
## 2        1.795  0.034 |
## 3       18.438  0.153 |
## 4        0.198  0.010 |
## 5        0.202  0.007 |
## 6       15.418  0.741 |
## 7        0.016  0.001 |
## 8        0.436  0.025 |
## 9        0.093  0.017 |
## 10      21.247  0.426 |
##
## Variables
##             Dim.1    ctr   cos2    Dim.2    ctr   cos2    Dim.3    ctr
## Points |  0.899 38.121  0.808 |  0.295  8.277  0.087 | -0.029  0.135
## Shots  |  0.733 25.371  0.538 | -0.248  5.869  0.061 |  0.620 60.294
## Box    | -0.237  2.657  0.056 |  0.934 83.291  0.873 |  0.250  9.822
## Passes |  0.847 33.851  0.717 |  0.164  2.564  0.027 | -0.436 29.750
##           cos2
## Points  0.001 |
## Shots   0.384 |
## Box     0.063 |
## Passes  0.190 |
```

```r
#Plot cumulative variances as explained by the components
plot(1:4, FootballPC$eig$"cumulative percentage of variance",
     type = "b",
     xlab = "Axes 1,2,...,4",
```

```
     ylab = "Cum. % of variance")
```




```
FootballPC$eig$"cumulative percentage of variance"
```

```
## [1]  52.96580  79.16301  95.10179 100.00000
```