# Lab 7: Principal Component Analysis

Submit your group report before the end of the lab.

Final report due: Final report Friday, October 2, 2020, by 11:59 pm.

Electronically submit one solution per group, including the Rmd file and the output file (either pdf or html). Please record the names of the group members who attended the lab.

# Purpose

• Learn how to perform and interpret principal component analysis (PCA). • Use principal componet scores to display and summarize differences in groups • Check the data prior to performing the PCA. • Diagnostic follow-up of the analysis.

# Data

This analysis examines scatter plot matrices and computes principal components for the 10k segments of a 100k road race. the data are from (Everitt 1994).

There is one line of data for each of 80 racers with eleven numbers on each line. The first ten columns give the times (minutes) to complete successive 10k segments of the race. The last column has the racer's age (in years).

race100 <- read.csv("race100k.csv")  
 head(race100)

## X0\_10k X10\_20k X20\_30k X30\_40k X40\_50k X50\_60k X60\_70k X70\_80k X80\_90k  
## 1 37.0 37.8 36.6 39.6 41.0 41.0 41.3 45.7 45.1  
## 2 39.5 42.2 40.0 42.3 40.6 40.8 42.0 43.7 41.0  
## 3 37.1 38.0 37.7 42.2 41.6 43.5 48.7 49.7 44.8  
## 4 37.0 37.8 36.6 39.6 41.0 44.8 44.5 49.4 44.6  
## 5 42.2 44.5 41.9 43.4 43.0 47.2 49.1 49.9 46.8  
## 6 43.0 44.6 41.2 42.1 42.5 46.8 47.5 55.8 56.6  
## X90\_100k age  
## 1 43.10 39  
## 2 43.96 39  
## 3 47.00 40  
## 4 47.70 36  
## 5 52.30 34  
## 6 58.60 46

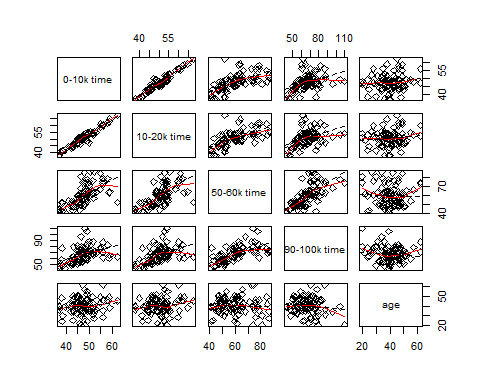
First compute the number of columns in the data frame and the sample size (number of runners).

p1<-dim(race100)[2]  
 n<-dim(race100)[1]  
 p<-p1-1

Use the pairs function to create a scatter plot matrix. Note that the columns to be included in the plot are put into the "choose" list. The panel.smooth function uses locally weighted regression to pass a smooth curve through each plot. The abline function uses least squares to fit a straight line to each plot. Including the line helps you to see if most of the marginal association between two variables on can be described by a straight line.

Recall that principal components are computed from variances and covariances (or correlations), which can only account for straight line relationships.

choose<-c(1,2,6,10,11)  
 par(pch=5)  
 pairs(race100[ ,choose],labels=c("0-10k time",  
 "10-20k time","50-60k time", "90-100k time","age"),  
 panel=function(x,y){panel.smooth(x,y)   
 abline(lsfit(x,y),lty=2) })



Compute principal components from the covariance matrix. This function creates a list with the following components sdev: standard deviations of the component scores (square roots of eigenvalues of the covariance matrix) rotation: The coefficients needed to compute the scores (elements of eigenvectors) x: a nxp matrix of scores

race.pc <- prcomp(race100[ ,-p1])  
 race.pc$sdev ## variation explained by each PC

## [1] 27.123463 9.923923 7.297834 6.102917 5.102212 4.151834 2.834300  
## [8] 2.060942 1.547235 1.135819

race.pc$rotation ## PC direction

## PC1 PC2 PC3 PC4 PC5  
## X0\_10k 0.1287926 -0.21059911 0.3615464 -0.033543077 0.147271116  
## X10\_20k 0.1519795 -0.24907923 0.4168216 -0.070771273 0.223835122  
## X20\_30k 0.1991613 -0.31427990 0.3411287 -0.053862467 0.247016251  
## X30\_40k 0.2397402 -0.33004401 0.2026687 -0.006573526 0.004696149  
## X40\_50k 0.3144251 -0.30213368 -0.1350869 0.110735209 -0.356368957  
## X50\_60k 0.4223146 -0.21465890 -0.2222736 -0.086787834 -0.373032863  
## X60\_70k 0.3358642 0.04958843 -0.1936251 -0.601557104 -0.189706738  
## X70\_80k 0.4066759 0.00858601 -0.5380052 0.128950685 0.719755126  
## X80\_90k 0.3990475 0.26746202 0.1491748 0.717507174 -0.209788222  
## X90\_100k 0.3853990 0.68882130 0.3482143 -0.278947530 0.054501733  
## PC6 PC7 PC8 PC9 PC10  
## X0\_10k -0.20575194 0.43236280 -0.28021009 0.038988136 0.690088073  
## X10\_20k -0.13094125 0.32564920 -0.22935824 0.046365827 -0.712785245  
## X20\_30k 0.05256055 -0.34345700 0.45763871 -0.586752802 0.082695763  
## X30\_40k 0.14386151 -0.44789166 0.10450365 0.745122555 0.070880091  
## X40\_50k 0.28455724 -0.24499887 -0.64624348 -0.306079913 -0.005450226  
## X50\_60k 0.29158828 0.53900089 0.44941277 0.037927421 -0.022906843  
## X60\_70k -0.64355431 -0.18441808 -0.02193712 -0.019261057 -0.019014886  
## X70\_80k 0.03482145 0.02932921 -0.08174832 0.036551593 0.018002601  
## X80\_90k -0.41424585 -0.04611157 0.11324179 -0.002675703 -0.039845245  
## X90\_100k 0.40508138 -0.03004076 -0.09451298 -0.002468560 0.032020909

## PC1 is the average time for a 10k segment

## PC2 is the contrast of the time between first half and the second half of the race; or we can also say PC2 is the contrast of the time between the first 6 10K segments and the last 10K segments

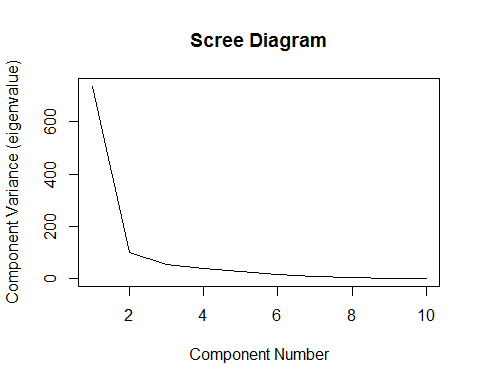
Compute proportion of total variance explained by each component.

summary(race.pc)

## Importance of components:  
## PC1 PC2 PC3 PC4 PC5 PC6  
## Standard deviation 27.1235 9.9239 7.29783 6.10292 5.10221 4.15183  
## Proportion of Variance 0.7477 0.1001 0.05413 0.03785 0.02646 0.01752  
## Cumulative Proportion 0.7477 0.8478 0.90194 0.93980 0.96625 0.98377  
## PC7 PC8 PC9 PC10  
## Standard deviation 2.83430 2.06094 1.54723 1.13582  
## Proportion of Variance 0.00816 0.00432 0.00243 0.00131  
## Cumulative Proportion 0.99194 0.99626 0.99869 1.00000

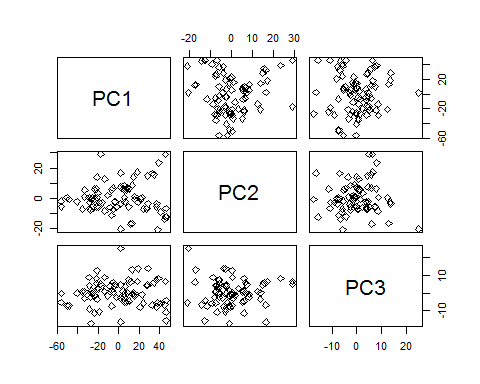
Produce a scree plot.

plot(race.pc$sdev^2, xlab="Component Number",  
 ylab="Component Variance (eigenvalue)",  
 main="Scree Diagram", type="l")



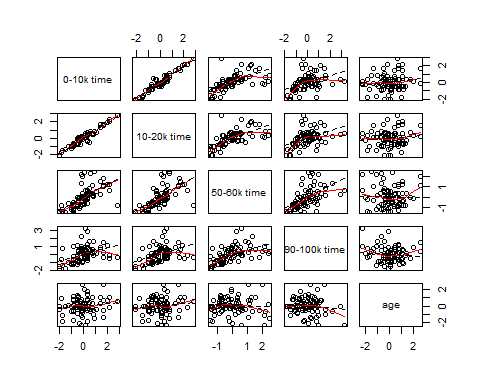
Plot component scores for the first three principal components

par(pch=5)  
 pairs(race.pc$x[,c(1,2,3)],labels=c("PC1","PC2","PC3"))



To compute principal components from a correlation matrix, you can first standardize the data

races <- scale(race100, center=T, scale=T)  
   
 choose<-c(1,2,5,10,11)  
  
 pairs(races[ ,choose],labels=c("0-10k time",  
 "10-20k time","50-60k time", "90-100k time", "age"),  
 panel=function(x,y){panel.smooth(x,y)  
 abline(lsfit(x,y),lty=2) })



racecor <- var(races)  
 racecor

## X0\_10k X10\_20k X20\_30k X30\_40k X40\_50k X50\_60k  
## X0\_10k 1.0000000 0.9510603 0.84458736 0.78585596 0.62053457 0.61789171  
## X10\_20k 0.9510603 1.0000000 0.89031061 0.82612495 0.64144268 0.63276548  
## X20\_30k 0.8445874 0.8903106 1.00000000 0.92108596 0.75594631 0.72509902  
## X30\_40k 0.7858560 0.8261249 0.92108596 1.00000000 0.88690905 0.84185641  
## X40\_50k 0.6205346 0.6414427 0.75594631 0.88690905 1.00000000 0.93641488  
## X50\_60k 0.6178917 0.6327655 0.72509902 0.84185641 0.93641488 1.00000000  
## X60\_70k 0.5313965 0.5409319 0.60502621 0.69065419 0.75419742 0.83957633  
## X70\_80k 0.4773723 0.5054520 0.61998205 0.69821518 0.78578147 0.84032251  
## X80\_90k 0.5423438 0.5338073 0.58357645 0.66735326 0.74134973 0.77257354  
## X90\_100k 0.4142609 0.4381283 0.46725334 0.50857719 0.54174220 0.65591894  
## age 0.1491725 0.1271041 0.01218286 0.04680206 -0.01607529 -0.04241971  
## X60\_70k X70\_80k X80\_90k X90\_100k age  
## X0\_10k 0.53139648 0.4773723 0.5423438 0.4142609 0.14917250  
## X10\_20k 0.54093190 0.5054520 0.5338073 0.4381283 0.12710409  
## X20\_30k 0.60502621 0.6199821 0.5835765 0.4672533 0.01218286  
## X30\_40k 0.69065419 0.6982152 0.6673533 0.5085772 0.04680206  
## X40\_50k 0.75419742 0.7857815 0.7413497 0.5417422 -0.01607529  
## X50\_60k 0.83957633 0.8403225 0.7725735 0.6559189 -0.04241971  
## X60\_70k 1.00000000 0.7796014 0.6972448 0.7191956 -0.04059097  
## X70\_80k 0.77960144 1.0000000 0.7637562 0.6634709 -0.20674428  
## X80\_90k 0.69724482 0.7637562 1.0000000 0.7797619 -0.12320048  
## X90\_100k 0.71919560 0.6634709 0.7797619 1.0000000 -0.11289354  
## age -0.04059097 -0.2067443 -0.1232005 -0.1128935 1.00000000

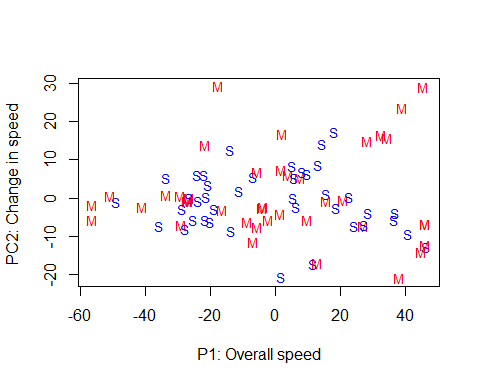
races.pcor <- prcomp(races[ ,-p1])  
  
 races.pcor$sdev

## 2.6912189 1.1331038 0.7439637 0.5451001 0.4536530 0.4279130 0.3300239 0.2204875 0.1984028 0.1923427

plot(races.pcor$sdev^2, xlab="Component Number",  
 ylab="Component Variance (eigenvalue)",  
 main="Scree Diagram", type="l")

Use the principal component scores from the raw data to look for differences among mature (age < 40) and senior (age > 40) runners. Mature runners will be indicated by "M" and senior runners will be indicated by "S".

race.type <-rep("M",n)  
 race.type[race100[ ,p1]>=40] <- "S"  
 race.col <- rep("red",n)  
 race.col[race100[ ,p1]>=40] <- "blue"  
   
 plot(race.pc$x[,1],race.pc$x[,2],  
 xlab="P1: Overall speed",  
 ylab="PC2: Change in speed ",type="n")  
 text(race.pc$x[,1],race.pc$x[,2],labels=race.type, cex=0.9, col=race.col)



# Data on Ames Livability

Ten rating variables of the livability in American small cities data. The data come from a book that was published in the late 90s, on livability of small cities in the USA. The Des Moines Register featured the book, because Ames was ranked the number 2 best small city in which to live! The study used 10 ratings variables -- Climate, Diversions, Economic, Education, Community, Health, House, Safety, Transportation, Urban -- with each city getting a rating between 0-100 on each of these varaibles. The scores were combined to give an overall rating for each city, Score.

towns <- read.csv("towns2.csv")  
dim(towns)

## [1] 193 16

head(towns)

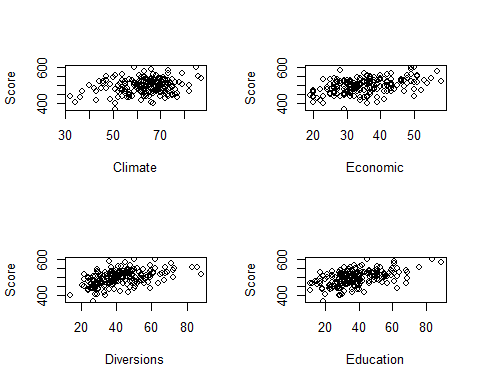
## City ID Population Climate Diversions Economic Education  
## 1 1.Albertville.AL 1 76802 68 44 43 12  
## 2 2.Auburn-Opelika.AL 2 91869 65 27 30 56  
## 3 3.Cullman.AL 3 71615 71 26 32 11  
## 4 4.Selma.AL 4 47991 64 28 26 28  
## 5 5.Talladega.AL 5 76034 62 26 20 15  
## 6 6.Fairbanks.AK 6 84711 32 59 50 63  
## Community Health House Safety Transportation Urban Score Latitude  
## 1 25 30 87 68 17 76 470 34.268  
## 2 31 55 77 60 36 72 509 32.610  
## 3 37 36 85 69 24 78 469 34.175  
## 4 49 35 83 32 36 61 442 32.407  
## 5 34 39 87 62 32 84 461 33.436  
## 6 45 37 45 55 56 0 442 50.000  
## Longitude  
## 1 -86.209  
## 2 -85.481  
## 3 -86.844  
## 4 -87.021  
## 5 -86.106  
## 6 -130.000

towns$ames <-0  
towns$ames[towns$Score==602] <- 1

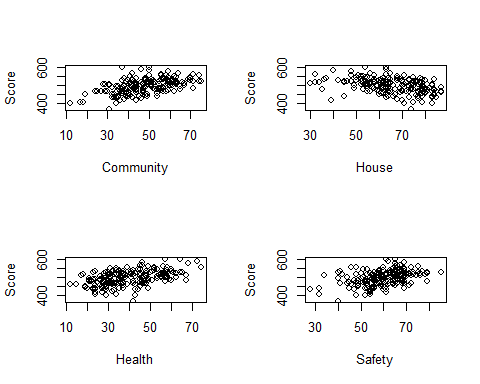
# Exercise 1

Examine how the the ten ratings variables are correlated with the livability scores (the Score variable) given to the cities. Write a short summary of your findings.

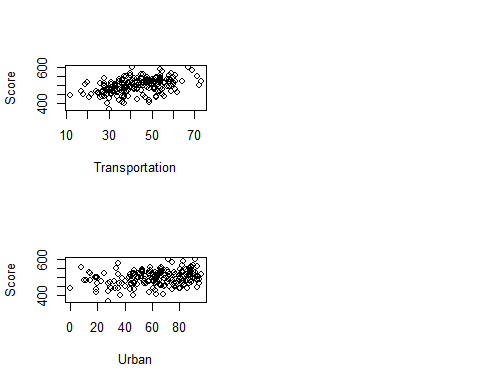
layout(matrix(1:4, nrow=2, ncol=2))  
plot(Score ~ Climate, data=towns, pch=1)  
plot(Score ~ Diversions, data=towns, pch=1)  
plot(Score ~ Economic, data=towns, pch=1)  
plot(Score ~ Education, data=towns, pch=1)



plot(Score ~ Community, data=towns, pch=1)  
plot(Score ~ Health, data=towns, pch=1)  
plot(Score ~ House, data=towns, pch=1)  
plot(Score ~ Safety, data=towns, pch=1)



plot(Score ~ Transportation, data=towns, pch=1)  
plot(Score ~ Urban, data=towns, pch=1)  
layout(matrix(1:1, nrow=1, ncol=1))



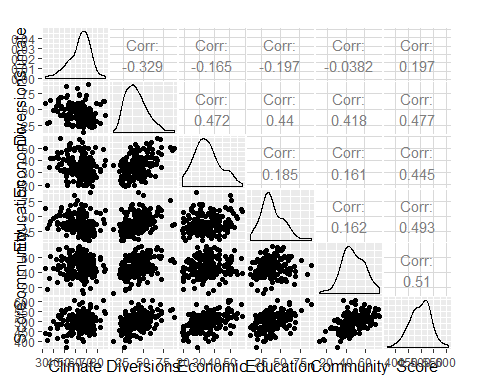
cor(towns$Score, towns[ ,4:13])

## Climate Diversions Economic Education Community Health  
## [1,] 0.1974325 0.4768799 0.4449764 0.4930572 0.5098451 0.477609  
## House Safety Transportation Urban  
## [1,] -0.3868395 0.381705 0.436461 0.2447653

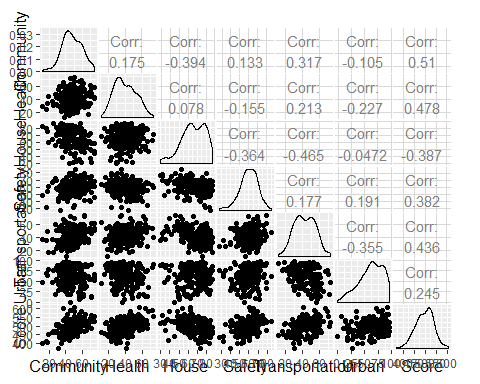
# Exercise 2

All the scores are measured on a scale of 0-100, so why is it still necessary to use the correlation matrix, or standardize the data, before doing a principla component analysis (PCA)? (Hint: Compute some summary statistics or make some plots.)

ggpairs(towns[,c(4:8, 14)])



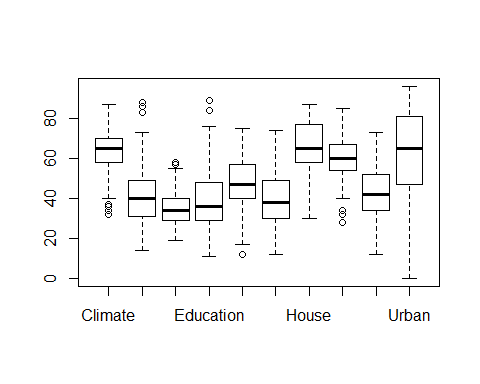
ggpairs(towns[,c(8:14)])



sapply(towns[ , 4:13], sd)

## Climate Diversions Economic Education Community   
## 9.933313 13.356393 8.444753 14.089614 11.923271   
## Health House Safety Transportation Urban   
## 12.878895 12.984470 9.648961 11.988998 22.057218

boxplot(towns[ ,4:13])



The best function to use in R is prcomp which handles a large number of variables better than other functions. The scale argument tells R to use the correlation matrix (TRUE) or covariance matrix (FALSE, default), and the retx argument tells R to compute the principal component scores and save them in the object that is created by the prcomp function.

The output has several main components: sdev contains the square root of the eigenvalues, rotation are the eigenvectors, and x contains the PC scores.

# Exercise 3

Present a summary of the PCA including the table of eigenvectors, a list of eigenvalues (variance), and cumulative percentage of total variance explained by the principal compontnes. (Be sure to make your output readable, eg rounding digits appropriately.) Make a scree plot.

1. How many PCs would you need to use to explain 80% of the total variation?
2. Explain how the Cumulative Proportion row of the summary of the PC was calculated.

towns.pca <- prcomp(towns[,3:12], scale=T, retx=T) # Not include Urban  
summary(towns.pca)

## Importance of components:  
## PC1 PC2 PC3 PC4 PC5 PC6  
## Standard deviation 1.7980 1.3025 1.0264 0.94756 0.90154 0.83711  
## Proportion of Variance 0.3233 0.1696 0.1053 0.08979 0.08128 0.07008  
## Cumulative Proportion 0.3233 0.4929 0.5983 0.68808 0.76935 0.83943  
## PC7 PC8 PC9 PC10  
## Standard deviation 0.79949 0.64644 0.5586 0.48642  
## Proportion of Variance 0.06392 0.04179 0.0312 0.02366  
## Cumulative Proportion 0.90335 0.94514 0.9763 1.00000

towns.pca$rotation

## PC1 PC2 PC3 PC4  
## Population -0.1045615 0.396162801 -0.152446509 0.780408448  
## Climate 0.2450912 0.199751718 -0.707747283 0.008226015  
## Diversions -0.4421788 -0.164787976 -0.025370601 -0.153244971  
## Economic -0.2577131 -0.417521458 -0.272632249 0.286267707  
## Education -0.4007243 0.142643866 0.107787682 0.078332808  
## Community -0.2957278 -0.004752030 -0.459786811 -0.419439821  
## Health -0.1579759 -0.514403231 -0.301283657 0.210877365  
## House 0.4322304 -0.314283286 -0.004989388 0.061116107  
## Safety -0.1608481 0.466780123 -0.196428808 -0.219655469  
## Transportation -0.4258786 0.006069204 0.220178853 0.083324706  
## PC5 PC6 PC7 PC8 PC9  
## Population -0.35191376 0.04401371 -0.11117390 0.07560671 0.14288044  
## Climate 0.17895538 -0.07337373 0.39071802 -0.42956837 0.02411383  
## Diversions -0.09818886 -0.37597628 0.01792978 -0.29760647 0.67622417  
## Economic 0.18193394 -0.50498741 -0.32195368 -0.02602258 -0.45643018  
## Education 0.42797389 0.03515338 0.55567841 0.15427659 -0.23770102  
## Community -0.56189570 0.19532823 -0.02017982 0.19145493 -0.27840400  
## Health 0.23480397 0.55882779 0.02888838 0.27438319 0.32737839  
## House 0.02546797 0.20906266 -0.18847283 -0.34616960 -0.05710667  
## Safety 0.49057746 0.15198303 -0.61894131 0.02428272 0.09420178  
## Transportation -0.07781047 0.42169328 -0.04871901 -0.68142675 -0.24509948  
## PC10  
## Population 0.18860887  
## Climate -0.15451833  
## Diversions 0.23669751  
## Economic -0.04701248  
## Education 0.47732549  
## Community 0.23819719  
## Health -0.15643894  
## House 0.71250994  
## Safety 0.11443060  
## Transportation -0.22912759

# Exercise 4

1. Explain how the variables contribute to the first two principal components.
2. Using three pieces of information, where the elbow is in the scree plot, the proportion of total variation, and the interpretation of the PCs, make an argument for how many PCs would you recommend to summarise this data?

# Exercise 5

Compare the scores for the first principal component with the Score variable in the data (this is the rating the article gives for each city). Which city would be rated first using the Score variable? Which city would be rated first using the scores for the first principal component? (You could make a plot of the Score variable against PC1, and compute the correlation between the two variables.) Do the two approaches give cities similar ratings? (You may need to multiple the scores for the first principal component by -1).

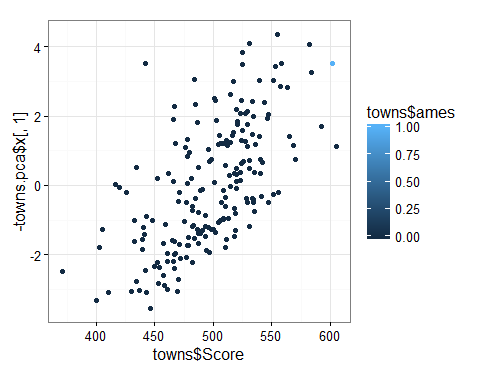
head(rownames(towns)[order(towns$Score, decreasing=T)])

## [1] "181" "52" "188" "108" "74" "182"

head(rownames(towns)[order(towns.pca$x[,1], decreasing=T)])

## [1] "113" "26" "169" "9" "176" "3"

library(ggplot2)  
qplot(towns$Score, -towns.pca$x[,1], col=as.factor(towns$ames)) + theme\_bw() + theme(aspect.ratio=1)



# Final Instructions

Put the names of all members of your group who participated in the Lab on Lab Report before you knit it.

Submit your report in Canvas before the end of the lab, including the Rmd file and the output file (either pdf or html).

Final report due: Final report Friday, October 2, 2020, by 11:59 pm.