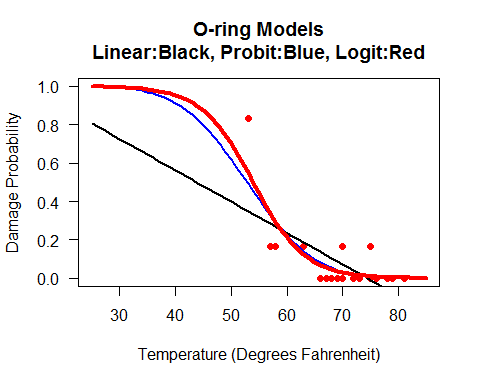
Homework2

Sevilla

July 25, 2018

# Log Regression

#Due Plot from 6   
  
#1. Set up  
library(faraway)  
data(orings)  
  
#2. Space shuttle O-ring seals data and the Challenger 1986 explosion   
orings

## temp damage  
## 1 53 5  
## 2 57 1  
## 3 58 1  
## 4 63 1  
## 5 66 0  
## 6 67 0  
## 7 67 0  
## 8 67 0  
## 9 68 0  
## 10 69 0  
## 11 70 1  
## 12 70 0  
## 13 70 1  
## 14 70 0  
## 15 72 0  
## 16 73 0  
## 17 75 0  
## 18 75 1  
## 19 76 0  
## 20 76 0  
## 21 78 0  
## 22 79 0  
## 23 81 0

#3. Regression Models for Binomial Data  
  
#4. Estimating model parameters  
  
#5. Fitting three models  
##Naive linear model  
linearModel = lm(damage/6 ~ temp,data=orings)  
summary(linearModel)

##   
## Call:  
## lm(formula = damage/6 ~ temp, data = orings)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.13786 -0.10345 -0.02369 0.06601 0.48345   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.21429 0.29993 4.049 0.000578 \*\*\*  
## temp -0.01631 0.00429 -3.801 0.001043 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.142 on 21 degrees of freedom  
## Multiple R-squared: 0.4076, Adjusted R-squared: 0.3794   
## F-statistic: 14.45 on 1 and 21 DF, p-value: 0.001043

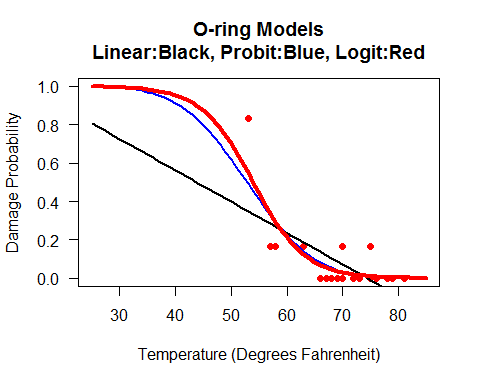
##Logit model  
  
logitModel =   
 glm(cbind(damage,6-damage) ~ temp,  
 family=binomial, data = orings)   
summary(logitModel)

##   
## Call:  
## glm(formula = cbind(damage, 6 - damage) ~ temp, family = binomial,   
## data = orings)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.9529 -0.7345 -0.4393 -0.2079 1.9565   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 11.66299 3.29626 3.538 0.000403 \*\*\*  
## temp -0.21623 0.05318 -4.066 4.78e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 38.898 on 22 degrees of freedom  
## Residual deviance: 16.912 on 21 degrees of freedom  
## AIC: 33.675  
##   
## Number of Fisher Scoring iterations: 6

## Probit   
probitModel =   
 glm(cbind(damage,6-damage) ~ temp,  
 family=binomial(probit),data=orings)  
summary(probitModel)

##   
## Call:  
## glm(formula = cbind(damage, 6 - damage) ~ temp, family = binomial(probit),   
## data = orings)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.0134 -0.7761 -0.4467 -0.1581 1.9983   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 5.59145 1.71055 3.269 0.00108 \*\*   
## temp -0.10580 0.02656 -3.984 6.79e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 38.898 on 22 degrees of freedom  
## Residual deviance: 18.131 on 21 degrees of freedom  
## AIC: 34.893  
##   
## Number of Fisher Scoring iterations: 6

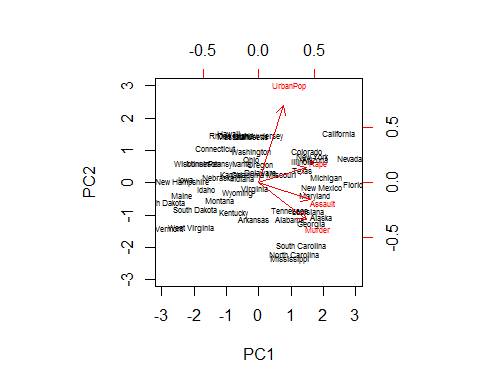
#6. Plots of fitted values  
windows()  
plot(damage/6~temp,orings,xlim=c(25,85),ylim=c(0,1),las=1,  
xlab="Temperature (Degrees Fahrenheit)", ylab="Damage Probability",  
pch=21,bg="red",col="red",  
main="O-ring Models\nLinear:Black, Probit:Blue, Logit:Red")  
  
tempGrid = 25:85  
a=coef(linearModel)  
lines(tempGrid,a[1]+a[2]\*tempGrid,col="black",lwd=2)  
  
a=coef(probitModel)  
lines(tempGrid,pnorm(a[1]+a[2]\*tempGrid),col="blue",lwd=2)  
  
a = coef(logitModel)  
lines(tempGrid,ilogit(a[1]+a[2]\*tempGrid),col="red",lwd=4)



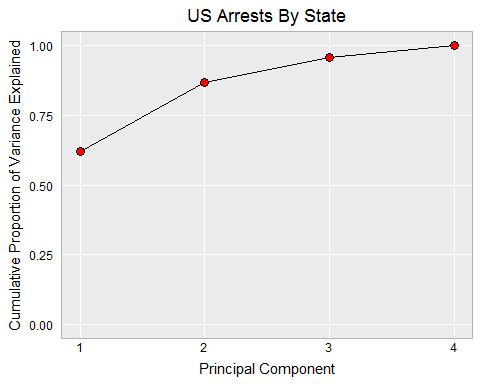
#7. Comment

# Crime Data

# Script from ISLR with slight modification  
  
**# Due 2 points  
# 3. Second Biplot**

****

**# 4. Second qplot**



# 0. Setup  
  
library(ggplot2)  
source("hw.R")  
  
# 1. Look at the Arrest Data  
states=row.names(USArrests)  
states

## [1] "Alabama" "Alaska" "Arizona" "Arkansas"   
## [5] "California" "Colorado" "Connecticut" "Delaware"   
## [9] "Florida" "Georgia" "Hawaii" "Idaho"   
## [13] "Illinois" "Indiana" "Iowa" "Kansas"   
## [17] "Kentucky" "Louisiana" "Maine" "Maryland"   
## [21] "Massachusetts" "Michigan" "Minnesota" "Mississippi"   
## [25] "Missouri" "Montana" "Nebraska" "Nevada"   
## [29] "New Hampshire" "New Jersey" "New Mexico" "New York"   
## [33] "North Carolina" "North Dakota" "Ohio" "Oklahoma"   
## [37] "Oregon" "Pennsylvania" "Rhode Island" "South Carolina"  
## [41] "South Dakota" "Tennessee" "Texas" "Utah"   
## [45] "Vermont" "Virginia" "Washington" "West Virginia"   
## [49] "Wisconsin" "Wyoming"

names(USArrests)

## [1] "Murder" "Assault" "UrbanPop" "Rape"

head(USArrests)

## Murder Assault UrbanPop Rape  
## Alabama 13.2 236 58 21.2  
## Alaska 10.0 263 48 44.5  
## Arizona 8.1 294 80 31.0  
## Arkansas 8.8 190 50 19.5  
## California 9.0 276 91 40.6  
## Colorado 7.9 204 78 38.7

colMeans(USArrests)

## Murder Assault UrbanPop Rape   
## 7.788 170.760 65.540 21.232

apply(USArrests, 2, var)

## Murder Assault UrbanPop Rape   
## 18.97047 6945.16571 209.51878 87.72916

# 2. Principal Components  
  
pr.out=prcomp(USArrests, scale=TRUE)  
names(pr.out)

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

pr.out$center

## Murder Assault UrbanPop Rape   
## 7.788 170.760 65.540 21.232

pr.out$scale

## Murder Assault UrbanPop Rape   
## 4.355510 83.337661 14.474763 9.366385

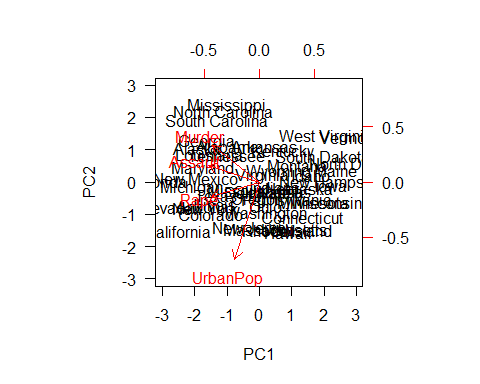
pr.out$rotation

## PC1 PC2 PC3 PC4  
## Murder -0.5358995 0.4181809 -0.3412327 0.64922780  
## Assault -0.5831836 0.1879856 -0.2681484 -0.74340748  
## UrbanPop -0.2781909 -0.8728062 -0.3780158 0.13387773  
## Rape -0.5434321 -0.1673186 0.8177779 0.08902432

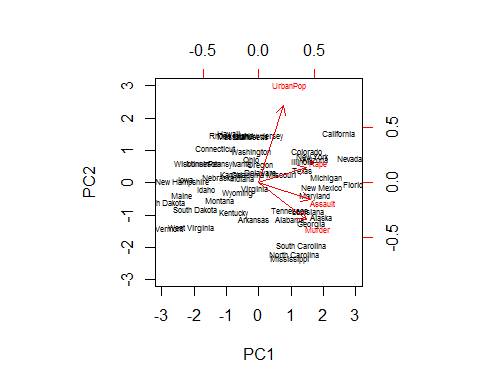
dim(pr.out$x)

## [1] 50 4

# 3. Biplots  
  
biplot(pr.out, scale=0,las=1)



pr.out$rotation=-pr.out$rotation  
pr.out$x=-pr.out$x  
  
windows(width=10, height=10)  
biplot(pr.out, scale=0,cex=.5)



# 4. Percent of Variance Explained  
  
pr.out$sdev

## [1] 1.5748783 0.9948694 0.5971291 0.4164494

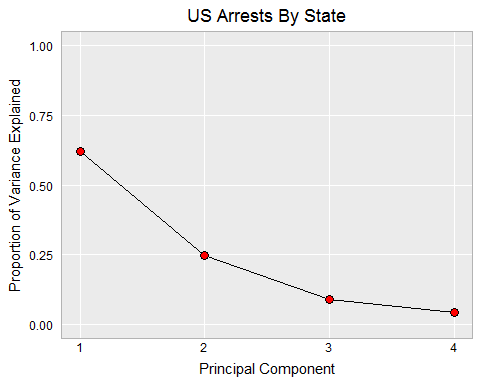
pr.var=pr.out$sdev^2  
pr.var

## [1] 2.4802416 0.9897652 0.3565632 0.1734301

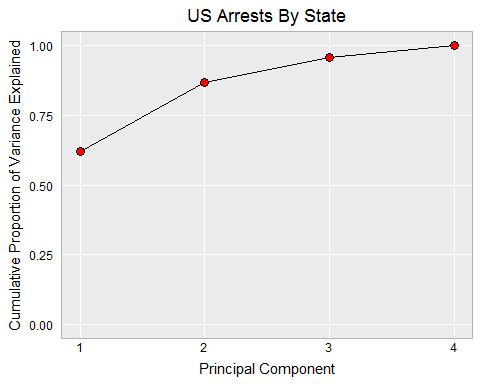
pve=pr.var/sum(pr.var)  
pve

## [1] 0.62006039 0.24744129 0.08914080 0.04335752

x = 1:length(pve)  
qplot(x,pve, xlab="Principal Component",   
 ylab="Proportion of Variance Explained",   
 main="US Arrests By State",ylim=c(0,1)) +  
 geom\_line()+geom\_point(shape=21,fill="red",cex=3)+hw

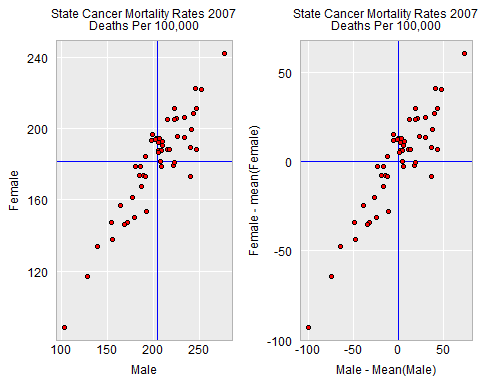


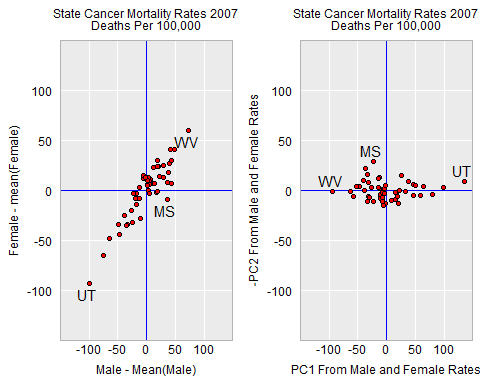
qplot(x,cumsum(pve), xlab="Principal Component",  
 ylab="Cumulative Proportion of Variance Explained",  
 main="US Arrests By State",ylim=c(0,1))+  
 geom\_line()+geom\_point(shape=21,fill="red",cex=3)+hw



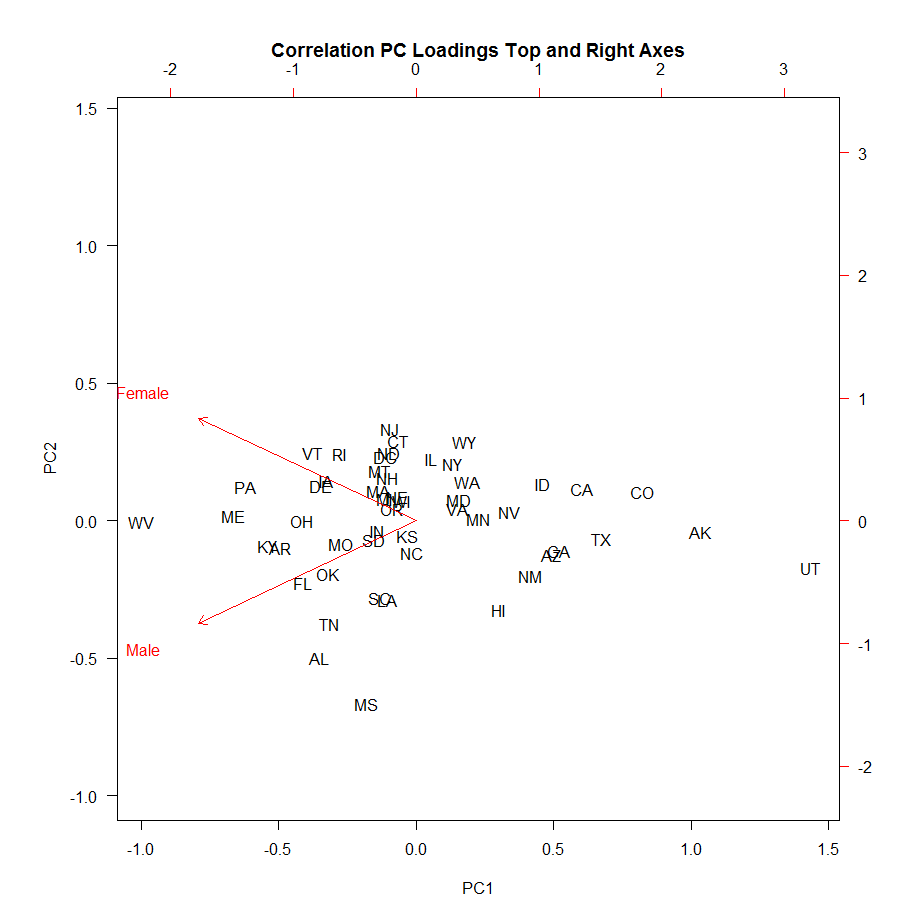
#2D Data

# Principal Components 2D   
# Illustrates centering and rotation of 2D data  
  
**#Due: 3 points  
#2. Juxtaposed plot**

**  
#4. Juxtaposed plot**

****

**#5. Second Biplot**

****

# 0. Setup  
library(MASS)  
library(ggplot2)  
library(gridExtra)  
  
hwThemeSmallText <- hw +  
 theme(  
 axis.text=element\_text(size=rel(.8)),  
 axis.title=element\_text(size=rel(.8)),  
 plot.title=element\_text(size=rel(.8))  
 )  
  
# 1. Read data and check for missing values  
   
mfcancer <- read.csv(file="US\_CancerMortality\_byGender.csv",  
 header=TRUE, as.is=TRUE)  
head(mfcancer)

## State Gender X1999 X2000 X2001 X2002 X2003 X2004 X2005 X2006 X2007  
## 1 AL Female 190.4 192.2 197.2 190.9 193.2 193.4 196.6 189.8 188.3  
## 2 AK Female 97.9 109.3 103.1 102.2 102.3 99.4 101.9 117.0 117.0  
## 3 AZ Female 166.7 165.2 166.1 156.5 159.1 154.8 153.0 149.3 147.5  
## 4 AR Female 204.3 203.1 200.4 206.0 199.8 211.3 201.8 196.2 208.5  
## 5 CA Female 154.2 154.4 154.0 151.9 151.5 147.5 149.4 147.5 147.6  
## 6 CO Female 136.8 135.1 140.9 139.3 137.7 134.8 137.5 136.3 134.0

tail(mfcancer)

## State Gender X1999 X2000 X2001 X2002 X2003 X2004 X2005 X2006 X2007  
## 97 VT Male 223.8 207.4 214.4 209.5 209.6 206.3 201.2 192.1 222.7  
## 98 VA Male 204.0 200.4 193.5 199.5 198.9 191.3 195.0 191.3 191.0  
## 99 WA Male 190.1 185.4 186.9 184.9 185.7 183.5 181.0 177.7 185.0  
## 100 WV Male 285.6 277.4 274.0 280.2 274.4 281.0 275.4 278.0 277.3  
## 101 WI Male 216.3 208.3 209.1 206.6 204.5 209.2 204.4 201.5 205.1  
## 102 WY Male 195.7 177.9 199.0 176.1 200.3 176.8 176.0 194.3 180.6

any(is.na(mfcancer))

## [1] FALSE

# Make a data frame for 2007   
# with male and female columns  
# in separate columns  
  
# Get the rate for column X2007  
rate <- mfcancer[, "X2007"]  
  
# Make a logical vector with TRUE for "Male"  
# FALSE otherwise (Females0  
male <- mfcancer[, "Gender"]=="Male"  
  
# column bind the male and female rates into  
# two column matrix  
mat <- cbind(rate[male], rate[!male])  
  
# Add state postal codes as rownames   
# Male and Female as column names  
rownames(mat) <- mfcancer$State[1:51]  
colnames(mat) <- c("Male","Female")  
mat

## Male Female  
## AL 246.9 188.3  
## AK 128.9 117.0  
## AZ 171.5 147.5  
## AR 243.5 208.5  
## CA 154.9 147.6  
## CO 139.3 134.0  
## CT 198.1 193.3  
## DE 224.6 205.7  
## DC 203.7 194.5  
## FL 241.7 199.6  
## GA 168.9 146.2  
## HI 192.8 153.6  
## ID 164.8 156.7  
## IL 191.8 184.3  
## IN 215.2 188.5  
## IA 222.6 205.1  
## KS 207.9 181.6  
## KY 246.8 211.5  
## LA 221.4 179.3  
## ME 251.8 222.1  
## MD 189.2 173.6  
## MA 210.0 192.6  
## MI 209.5 190.5  
## MN 186.7 167.5  
## MS 240.0 173.0  
## MO 226.6 195.3  
## MT 207.1 194.5  
## NE 205.6 187.8  
## NV 177.4 161.4  
## NH 205.8 192.0  
## NJ 198.7 196.4  
## NM 179.8 150.0  
## NY 186.6 178.9  
## NC 208.7 178.6  
## ND 202.1 194.1  
## OH 234.0 206.3  
## OK 233.8 194.9  
## OR 208.3 187.6  
## PA 245.2 222.6  
## RI 215.6 205.0  
## SC 222.8 180.9  
## SD 217.0 188.3  
## TN 239.8 189.5  
## TX 156.1 138.1  
## UT 103.8 88.8  
## VT 222.7 211.2  
## VA 191.0 173.2  
## WA 185.0 173.8  
## WV 277.3 241.8  
## WI 205.1 186.6  
## WY 180.6 178.7

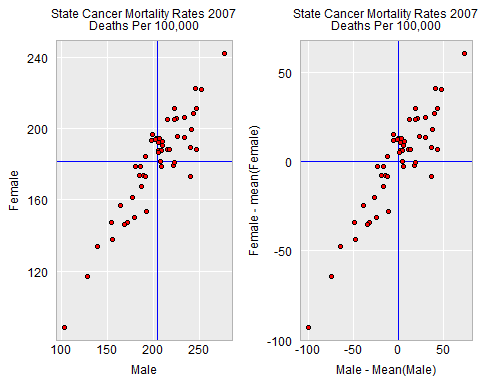
# Make a data.frame  
df <- as.data.frame(mat)  
head(df)

## Male Female  
## AL 246.9 188.3  
## AK 128.9 117.0  
## AZ 171.5 147.5  
## AR 243.5 208.5  
## CA 154.9 147.6  
## CO 139.3 134.0

# 2. Center the variables   
# and compare plots of the data and  
# centered data. Just the axis scales  
# should look different   
  
# The scale function   
# scale(x,cen=TRUE,scale=TRUE) works on a numeric matrix  
  
# By default it subtracts the mean from each column  
# and then divides each column by its standard deviation.  
  
matCen <- scale(mat,scale=FALSE)  
dfCen <- as.data.frame(matCen)  
head(dfCen)

## Male Female  
## AL 42.80196 6.762745  
## AK -75.19804 -64.537255  
## AZ -32.59804 -34.037255  
## AR 39.40196 26.962745  
## CA -49.19804 -33.937255  
## CO -64.79804 -47.537255

# Save the means for reference lines  
maleM <- mean(df$Male)  
femaleM <- mean(df$Female)  
  
# Produce female rate versus male rate scatterplots  
# Produce both uncentered and centered scatterplot objects  
# Juxtapose the plots using grid.arrange()  
  
title <-paste("State Cancer Mortality Rates 2007",  
 "\nDeaths Per 100,000",sep="")   
  
p <- ggplot(df,aes(x=Male,y=Female))+  
 geom\_hline(yintercept=femaleM,color='blue')+   
 geom\_vline(xintercept=maleM,color='blue')+  
 geom\_point(fill="red",shape=21)+  
 labs( title=title)+hwThemeSmallText  
  
pCen<- ggplot(dfCen,aes(x=Male,y=Female))+  
 geom\_hline(yintercept=0,color='blue')+   
 geom\_vline(xintercept=0,color='blue')+  
 geom\_point(fill="red",shape=21)+  
 labs(x="Male - Mean(Male)",  
 y="Female - mean(Female)",  
 title=title)+hwThemeSmallText  
  
windows(width=6, height=3)  
grid.arrange(p,pCen,ncol=2)



pc <- prcomp(mat,scale=FALSE)  
pcDat <- pc$x # principal components  
pcRotate <- pc$rotation #rotation matrix  
det(pcRotate) # -1 means there is a reflection

## [1] -1

# Rotate the centered data and compare  
  
matRot <- matCen %\*% pcRotate  
head(pcDat)

## PC1 PC2  
## AL -37.50711 -21.701587  
## AK 99.05604 -2.775543  
## AZ 46.75710 -5.911030  
## AR -47.58711 -3.869211  
## CA 59.58888 4.620616  
## CO 80.27153 3.880451

head(matRot)

## PC1 PC2  
## AL -37.50711 -21.701587  
## AK 99.05604 -2.775543  
## AZ 46.75710 -5.911030  
## AR -47.58711 -3.869211  
## CA 59.58888 4.620616  
## CO 80.27153 3.880451

all.equal(pcDat,matRot)

## [1] TRUE

round(pcRotate,2)

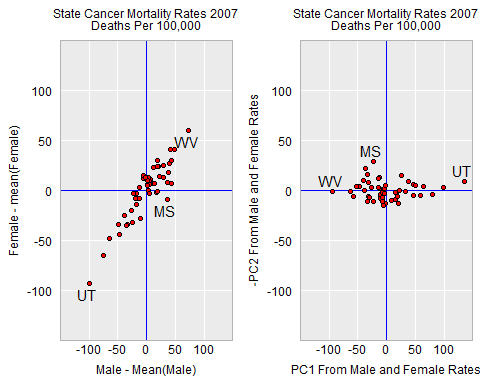
## PC1 PC2  
## Male -0.78 -0.63  
## Female -0.63 0.78

a <- -pcRotate[1,1]  
asin(a)\*180/pi

## [1] 50.9678

# 4. Juxtapose the centered data and   
# principle component scatterplots  
# using the same x and y axis scale  
# limits.  
#  
# Find x and y axis limits to accommodate  
# rotating the data.   
# The first principal component has   
# large variance but not necessarily  
# the point furthest from the origin  
  
big <- max(abs(pcDat))  
  
dfPC <- data.frame(PC1=pcDat[,1],PC2 = pcDat[,2])  
rownames(dfPC) = rownames(dfCen)  
  
# Find some extreme points to label  
# with state postal codes  
  
id1 <- which.min(dfPC$PC1)  
id2 <- which.max(dfPC$PC1)  
id3 <- which.min(dfPC$PC2)  
subs <- c(id1,id2,id3)  
  
# Add a column to dfPC and to dfCen  
# for postal codes. Fill the columns  
# with NAs except for the 3 ids  
# we want to show.   
   
dfPC$State <- NA  
dfPC$State[subs] <- row.names(dfPC)[subs]   
  
dfCen$State <- NA  
dfCen$State[subs] <- row.names(dfPC)[subs]  
  
# Store the two plots and then juxtapose  
#  
pCen <- ggplot(dfCen,aes(x=Male,y=Female))+  
 geom\_hline(yintercept=0,color='blue')+   
 geom\_vline(xintercept=0,color='blue')+  
 geom\_point(fill="red",shape=21)+  
 ylim(-big,big)+  
 xlim(-big,big)+  
 geom\_text(aes(y=Female-5,label=State),size=4,vjust=1)+  
 labs(x="Male - Mean(Male)",  
 y="Female - mean(Female)",  
 title=title)+hwThemeSmallText  
  
pRot <- ggplot(dfPC,aes(x=PC1,y=-PC2))+  
 geom\_hline(yintercept=0,color='blue')+   
 geom\_vline(xintercept=0,color='blue')+  
 geom\_point(fill="red",shape=21)+  
 ylim(-big,big)+  
 xlim(-big,big)+  
 geom\_text(aes(y=-PC2+5,label=State),size=4,vjust=0)+  
 labs(x="PC1 From Male and Female Rates",  
 y="-PC2 From Male and Female Rates",  
 title=title)+hwThemeSmallText  
  
grid.arrange(pCen,pRot,ncol=2)

## Warning: Removed 48 rows containing missing values (geom\_text).  
  
## Warning: Removed 48 rows containing missing values (geom\_text).



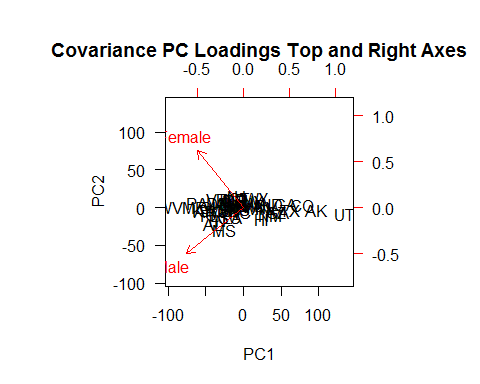
with(dfPC, sum(PC1 \* PC2 ))

## [1] -1.099587e-11

with(dfPC, sum(PC1 \* (-PC2)))

## [1] 1.099587e-11

# Both are zero for practical purposes.  
  
  
# 5. Biplot  
  
  
# Without scaling the data Covariance version  
  
windows(width=6,height=6)  
biplot(pc,scale=0,las=1,  
 main= "Covariance PC Loadings Top and Right Axes")



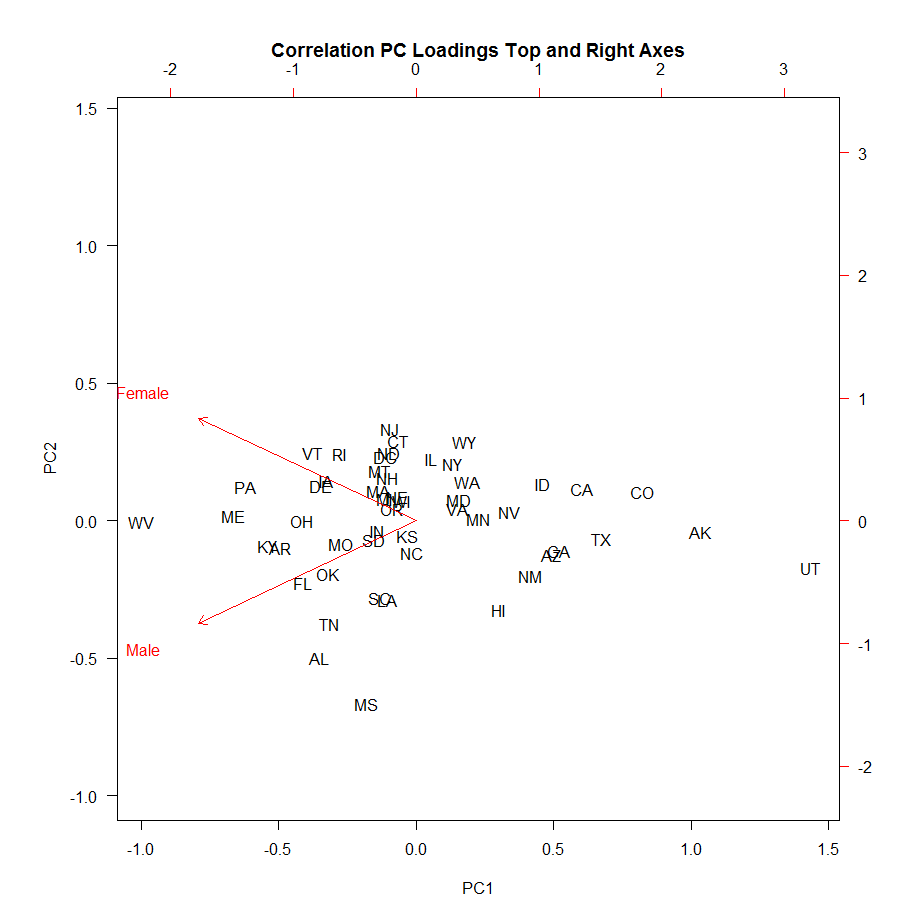
# The biplot show two superimposed plot  
# that have different scales.  
# Such plots run the risk of being  
# confusing.   
  
# The first plot shows the first two principal  
# components as points with respect to the  
# bottom and left axes.   
  
head(pc$x)

## PC1 PC2  
## AL -37.50711 -21.701587  
## AK 99.05604 -2.775543  
## AZ 46.75710 -5.911030  
## AR -47.58711 -3.869211  
## CA 59.58888 4.620616  
## CO 80.27153 3.880451

# This principal components did not divide the variables  
# by their standard deviations so are based on the covariance  
# matrix of the variables.   
  
# The second plot in red show values from the  
# from the rotation matrix using red arrows  
# tips with the arrows starting from the origin.  
# The Male arrow tip values can be read  
# from left column of the rotation matrix show  
# below. The Female arrow tip values are from  
# right column  
  
pc$rot

## PC1 PC2  
## Male -0.7767922 -0.6297570  
## Female -0.6297570 0.7767922

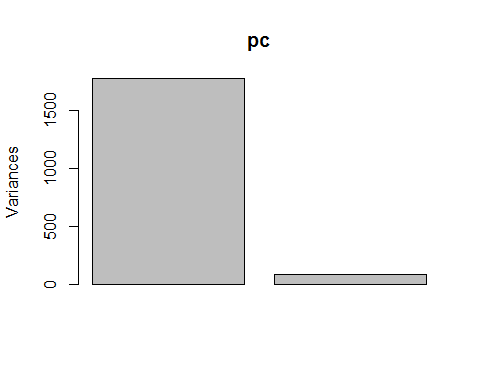
# The axes for the second plot appear on the top amd right  
# of biplot. Ideally tick mark labels should also  
# be red to match the red tick marks, arrows and arrow  
# labels.   
  
# If we read down the rotation matrix PC1 column  
# we see linear combination of coefficents that multiply  
# the center male and female values for each state to   
# produce PC1. Roughly speaking the first (PC1) linear combination  
# is a negative scale sum of the centered male and female  
# values for each states. Roughly speaking the second linear  
# combination is the scaled difference of the center male  
# and female value for each state.  
# The columns are sometimes called factor loadings.   
  
  
# With scaling the Correlation version of PCs  
  
windows(width=6, height=6)  
pcCor <- prcomp(mat,scale=TRUE)  
biplot(pcCor,scale=1,las=1,  
 main="Correlation PC Loadings Top and Right Axes")



# 6. Percent of variability represented  
  
# Guidance suggest looking the   
# standard deviations of the principal  
# components  
  
pc$sdev

## [1] 42.102563 9.196496

# The screeplot shows variances  
screeplot(pc)



# Another choice is to look   
# cumulative percent of total  
# variance   
dfPC$State <- NULL  
vec <- diag(var(dfPC))  
100\*cumsum(vec)/sum(vec)

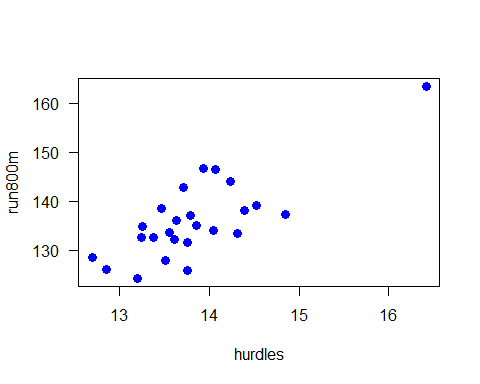
## PC1 PC2   
## 95.44608 100.00000

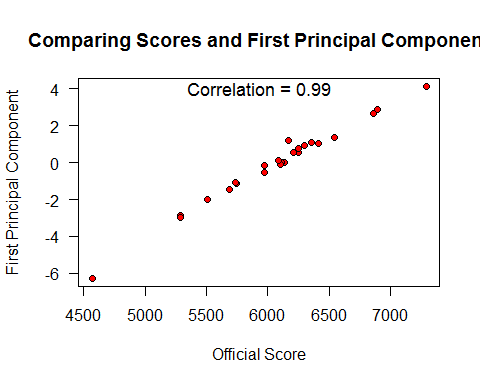
# The first principal component accounts for  
# 95% of the variability   
  
# 6. Ordering States by the first principal component  
  
ord <- with(dfPC,order(PC1))  
round( dfPC[ord,1:2],1)

## PC1 PC2  
## WV -94.8 0.7  
## ME -62.6 1.5  
## PA -57.8 6.0  
## KY -52.0 -3.6  
## AR -47.6 -3.9  
## FL -40.6 -9.6  
## OH -38.8 0.4  
## AL -37.5 -21.7  
## VT -33.1 11.3  
## TN -32.7 -16.3  
## OK -31.5 -8.3  
## DE -31.1 5.9  
## IA -29.2 6.7  
## MO -26.1 -3.5  
## RI -23.7 11.0  
## MS -22.5 -29.2  
## SD -14.3 -2.9  
## SC -14.1 -12.3  
## IN -13.0 -1.6  
## LA -12.0 -12.6  
## MA -11.6 4.9  
## MT -10.5 8.2  
## MI -9.8 3.6  
## NH -7.9 7.1  
## DC -7.9 10.3  
## OR -7.1 2.1  
## ND -6.4 11.0  
## NJ -5.2 14.9  
## NE -5.1 3.9  
## WI -4.0 3.3  
## KS -3.0 -2.3  
## CT -2.7 12.9  
## NC -1.7 -5.2  
## IL 7.8 9.9  
## NY 15.3 9.0  
## VA 15.4 1.8  
## MD 16.6 3.2  
## WA 19.7 6.0  
## WY 20.0 12.6  
## MN 22.4 0.1  
## HI 26.4 -14.6  
## NV 33.4 1.2  
## NM 38.7 -9.2  
## ID 46.2 5.5  
## AZ 46.8 -5.9  
## GA 49.6 -5.3  
## CA 59.6 4.6  
## TX 64.6 -3.5  
## CO 80.3 3.9  
## AK 99.1 -2.8  
## UT 136.3 -8.9

# The first principal component is often useful for sorting cases

Heptathlon

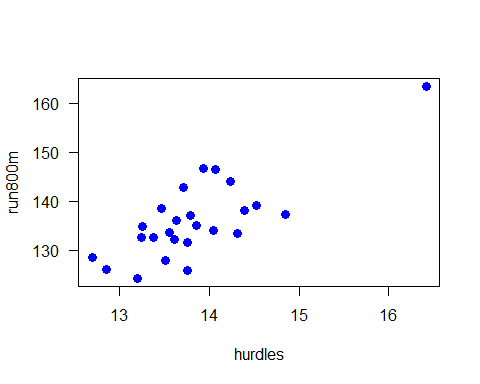
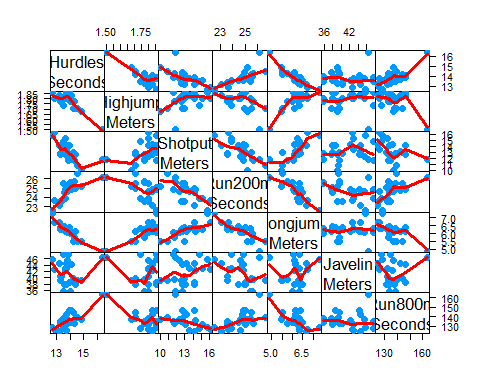
#Sections  
#1. Read the data and make a scatterplot matrix  
#2. Use identify() to name an outlier   
#3. Transforming data before input to principal components   
#4. Obtaining principal components  
#5. Assess the variation in the principal components  
#6. Relate the first principal component to the judges score  
  
**#Due: Plot from 2 with the name of the outlier and the plot from 6.   
Launa (PNG)**

****

#1. Read the data and make a scatterplot matrix  
heptathlon = read.csv(file="heptathlon.csv",row.names=1)  
heptathlon

## hurdles highjump shot run200m longjump javelin  
## Joyner-Kersee (USA) 12.69 1.86 15.80 22.56 7.27 45.66  
## John (GDR) 12.85 1.80 16.23 23.65 6.71 42.56  
## Behmer (GDR) 13.20 1.83 14.20 23.10 6.68 44.54  
## Sablovskaite (URS) 13.61 1.80 15.23 23.92 6.25 42.78  
## Choubenkova (URS) 13.51 1.74 14.76 23.93 6.32 47.46  
## Schulz (GDR) 13.75 1.83 13.50 24.65 6.33 42.82  
## Fleming (AUS) 13.38 1.80 12.88 23.59 6.37 40.28  
## Greiner (USA) 13.55 1.80 14.13 24.48 6.47 38.00  
## Lajbnerova (CZE) 13.63 1.83 14.28 24.86 6.11 42.20  
## Bouraga (URS) 13.25 1.77 12.62 23.59 6.28 39.06  
## Wijnsma (HOL) 13.75 1.86 13.01 25.03 6.34 37.86  
## Dimitrova (BUL) 13.24 1.80 12.88 23.59 6.37 40.28  
## Scheider (SWI) 13.85 1.86 11.58 24.87 6.05 47.50  
## Braun (FRG) 13.71 1.83 13.16 24.78 6.12 44.58  
## Ruotsalainen (FIN) 13.79 1.80 12.32 24.61 6.08 45.44  
## Yuping (CHN) 13.93 1.86 14.21 25.00 6.40 38.60  
## Hagger (GB) 13.47 1.80 12.75 25.47 6.34 35.76  
## Brown (USA) 14.07 1.83 12.69 24.83 6.13 44.34  
## Mulliner (GB) 14.39 1.71 12.68 24.92 6.10 37.76  
## Hautenauve (BEL) 14.04 1.77 11.81 25.61 5.99 35.68  
## Kytola (FIN) 14.31 1.77 11.66 25.69 5.75 39.48  
## Geremias (BRA) 14.23 1.71 12.95 25.50 5.50 39.64  
## Hui-Ing (TAI) 14.85 1.68 10.00 25.23 5.47 39.14  
## Jeong-Mi (KOR) 14.53 1.71 10.83 26.61 5.50 39.26  
## Launa (PNG) 16.42 1.50 11.78 26.16 4.88 46.38  
## run800m score  
## Joyner-Kersee (USA) 128.51 7291  
## John (GDR) 126.12 6897  
## Behmer (GDR) 124.20 6858  
## Sablovskaite (URS) 132.24 6540  
## Choubenkova (URS) 127.90 6540  
## Schulz (GDR) 125.79 6411  
## Fleming (AUS) 132.54 6351  
## Greiner (USA) 133.65 6297  
## Lajbnerova (CZE) 136.05 6252  
## Bouraga (URS) 134.74 6252  
## Wijnsma (HOL) 131.49 6205  
## Dimitrova (BUL) 132.54 6171  
## Scheider (SWI) 134.93 6137  
## Braun (FRG) 142.82 6109  
## Ruotsalainen (FIN) 137.06 6101  
## Yuping (CHN) 146.67 6087  
## Hagger (GB) 138.48 5975  
## Brown (USA) 146.43 5972  
## Mulliner (GB) 138.02 5746  
## Hautenauve (BEL) 133.90 5734  
## Kytola (FIN) 133.35 5686  
## Geremias (BRA) 144.02 5508  
## Hui-Ing (TAI) 137.30 5290  
## Jeong-Mi (KOR) 139.17 5289  
## Launa (PNG) 163.43 4566

# remove the official score  
hepDat = heptathlon[,-ncol(heptathlon)]  
  
{windows(width=7,height=7)   
nam <- c("Hurdles\nSeconds","Highjump\nMeters","Shotput\nMeters",  
 "Run200m\nSeconds","Longjump\nMeters","Javelin\nMeters",  
 "Run800m\nSeconds")  
myPanelSmooth = function(x,y,...)panel.smooth(x,y,lwd=3,col.smooth='red',...)  
pairs(hepDat,lab=nam, panel=myPanelSmooth,pch=21,cex=1.5,  
 col=rgb(0,.6,1),bg=rgb(0,.6,1),gap=0,las=1)  
}  
#2 Identify  
plot(hepDat[c(1,7)],pch=16,col='blue',cex=1.2,las=1)



#identify(heptathlon[,1],heptathlon[,7],lab=row.names(heptathlon)) #Launa(PNG)  
  
#3. Transforming data before input to principal components   
  
hepDat$hurdles=max(hepDat$hurdles)-hepDat$hurdles  
hepDat$run200m=max(hepDat$run200m)-hepDat$run200m  
hepDat$run800m=max(hepDat$run800m)-hepDat$run800m  
  
#4. Obtaining principal components  
  
heptPca = prcomp(hepDat,scale=TRUE)   
  
heptPca$center # means of the variables

## hurdles highjump shot run200m longjump javelin run800m   
## 2.5800 1.7820 13.1176 1.9608 6.1524 41.4824 27.3760

heptPca$scale # standard deviation of the variables

## hurdles highjump shot run200m longjump javelin   
## 0.73664781 0.07794229 1.49188438 0.96955712 0.47421233 3.54565612   
## run800m   
## 8.29108809

heptPca$x # principal components

## PC1 PC2 PC3 PC4  
## Joyner-Kersee (USA) -4.121447626 -1.24240435 0.36991309 0.02300174  
## John (GDR) -2.882185935 -0.52372600 0.89741472 -0.47545176  
## Behmer (GDR) -2.649633766 -0.67876243 -0.45917668 -0.67962860  
## Sablovskaite (URS) -1.343351210 -0.69228324 0.59527044 -0.14067052  
## Choubenkova (URS) -1.359025696 -1.75316563 -0.15070126 -0.83595001  
## Schulz (GDR) -1.043847471 0.07940725 -0.67453049 -0.20557253  
## Fleming (AUS) -1.100385639 0.32375304 -0.07343168 -0.48627848  
## Greiner (USA) -0.923173639 0.80681365 0.81241866 -0.03022915  
## Lajbnerova (CZE) -0.530250689 -0.14632191 0.16122744 0.61590242  
## Bouraga (URS) -0.759819024 0.52601568 0.18316881 -0.66756426  
## Wijnsma (HOL) -0.556268302 1.39628179 -0.13619463 0.40503603  
## Dimitrova (BUL) -1.186453832 0.35376586 -0.08201243 -0.48123479  
## Scheider (SWI) 0.015461226 -0.80644305 -1.96745373 0.73341733  
## Braun (FRG) 0.003774223 -0.71479785 -0.32496780 1.06604134  
## Ruotsalainen (FIN) 0.090747709 -0.76304501 -0.94571404 0.26883477  
## Yuping (CHN) -0.137225440 0.53724054 1.06529469 1.63144151  
## Hagger (GB) 0.171128651 1.74319472 0.58701048 0.47103131  
## Brown (USA) 0.519252646 -0.72696476 -0.31302308 1.28942720  
## Mulliner (GB) 1.125481833 0.63479040 0.72530080 -0.57961844  
## Hautenauve (BEL) 1.085697646 1.84722368 0.01452749 -0.25561691  
## Kytola (FIN) 1.447055499 0.92446876 -0.64596313 -0.21493997  
## Geremias (BRA) 2.014029620 0.09304121 0.64802905 0.02454548  
## Hui-Ing (TAI) 2.880298635 0.66150588 -0.74936718 -1.11903480  
## Jeong-Mi (KOR) 2.970118607 0.95961101 -0.57118753 -0.11547402  
## Launa (PNG) 6.270021972 -2.83919926 1.03414797 -0.24141489  
## PC5 PC6 PC7  
## Joyner-Kersee (USA) -0.42600624 0.339329222 0.347921325  
## John (GDR) 0.70306588 -0.238087298 0.144015774  
## Behmer (GDR) -0.10552518 0.239190707 -0.129647756  
## Sablovskaite (URS) 0.45392816 -0.091805638 -0.486577968  
## Choubenkova (URS) 0.68719483 -0.126303968 0.239482044  
## Schulz (GDR) 0.73793351 0.355789386 -0.103414314  
## Fleming (AUS) -0.76299122 -0.084844490 -0.142871612  
## Greiner (USA) 0.09086737 0.151561253 0.034237928  
## Lajbnerova (CZE) 0.56851477 -0.265359696 -0.249591589  
## Bouraga (URS) -1.02148109 -0.396397714 -0.020405097  
## Wijnsma (HOL) 0.29221101 0.344582964 -0.182701990  
## Dimitrova (BUL) -0.78103608 -0.233718538 -0.070605615  
## Scheider (SWI) -0.02177427 0.004249913 0.036155878  
## Braun (FRG) -0.18389959 -0.272903729 0.044351160  
## Ruotsalainen (FIN) -0.18416945 -0.141403697 0.135136482  
## Yuping (CHN) -0.21162048 0.280043639 -0.171160984  
## Hagger (GB) -0.05781435 -0.147155606 0.520000710  
## Brown (USA) -0.49779301 0.071211150 -0.005529394  
## Mulliner (GB) -0.15611502 0.427484048 0.081522940  
## Hautenauve (BEL) 0.19143514 0.100087033 0.085430091  
## Kytola (FIN) 0.49993839 0.072673266 -0.125585203  
## Geremias (BRA) 0.24445870 -0.640572055 -0.215626046  
## Hui-Ing (TAI) -0.47418755 0.180568513 -0.207364881  
## Jeong-Mi (KOR) 0.58055249 -0.183940799 0.381783751  
## Launa (PNG) -0.16568672 0.255722133 0.061044365

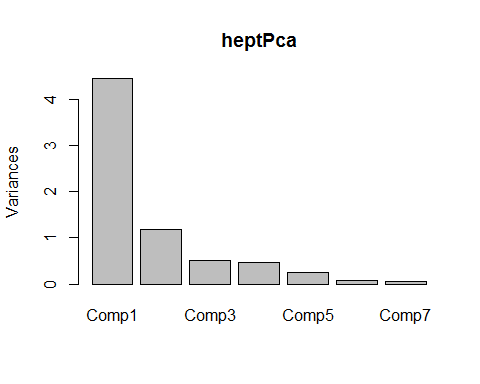
heptPca$sdev # standand deviations of the principal components

## [1] 2.1119364 1.0928497 0.7218131 0.6761411 0.4952441 0.2701029 0.2213617

round(heptPca$rotation,2)

## PC1 PC2 PC3 PC4 PC5 PC6 PC7  
## hurdles -0.45 0.16 -0.05 0.03 -0.09 -0.78 0.38  
## highjump -0.38 0.25 -0.37 0.68 0.02 0.10 -0.43  
## shot -0.36 -0.29 0.68 0.12 0.51 -0.05 -0.22  
## run200m -0.41 -0.26 0.08 -0.36 -0.65 0.02 -0.45  
## longjump -0.46 0.06 0.14 0.11 -0.18 0.59 0.61  
## javelin -0.08 -0.84 -0.47 0.12 0.14 -0.03 0.17  
## run800m -0.37 0.22 -0.40 -0.60 0.50 0.16 -0.10

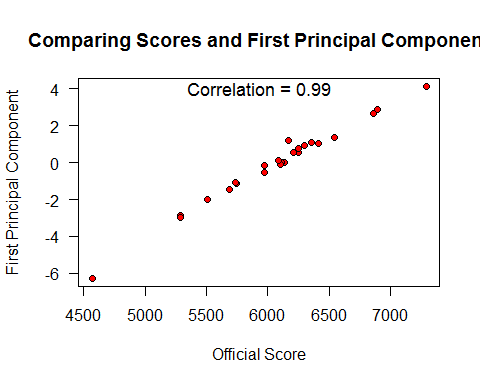
#5. Assess the variation in the principal components  
names(heptPca$sdev) = paste('Comp',1:length(heptPca$sdev),sep='')  
plot(heptPca)



heptVar = heptPca$sdev\*\*2  
100\*cumsum(heptVar)/sum(heptVar)

## Comp1 Comp2 Comp3 Comp4 Comp5 Comp6 Comp7   
## 63.71822 80.77994 88.22300 94.75395 98.25776 99.29999 100.00000

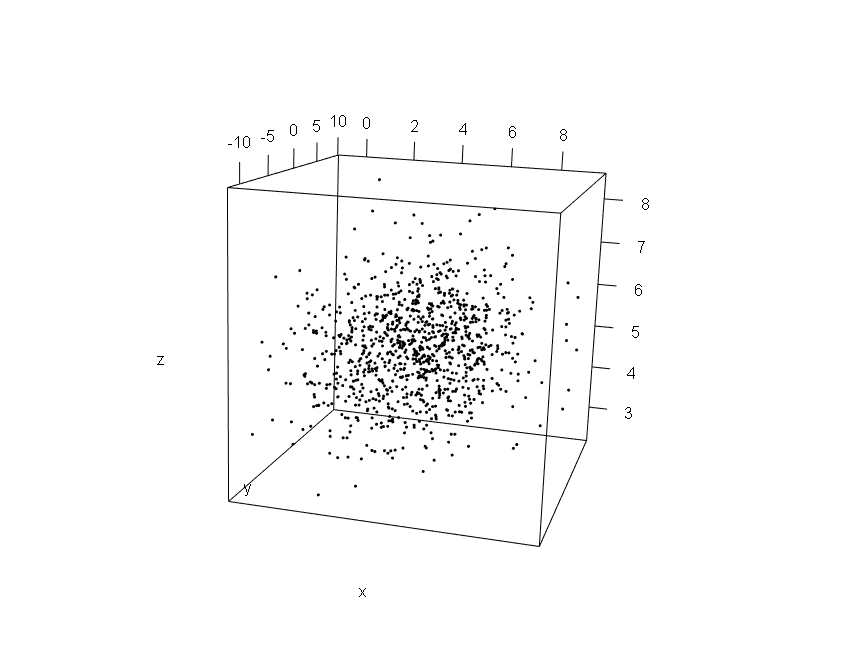
#6. Relate the first principal component to the judges score  
plot(heptathlon$score,-heptPca$x[,1],las=1,pch=21,bg="red",  
 xlab="Official Score",ylab="First Principal Component",  
 main="Comparing Scores and First Principal Component")  
correl = cor(-heptPca$x[,1],heptathlon$score)  
xloc = mean(par()$usr[1:2])  
text(xloc,4,paste("Correlation =",round(correl,2)),adj=.5,cex=1.1)



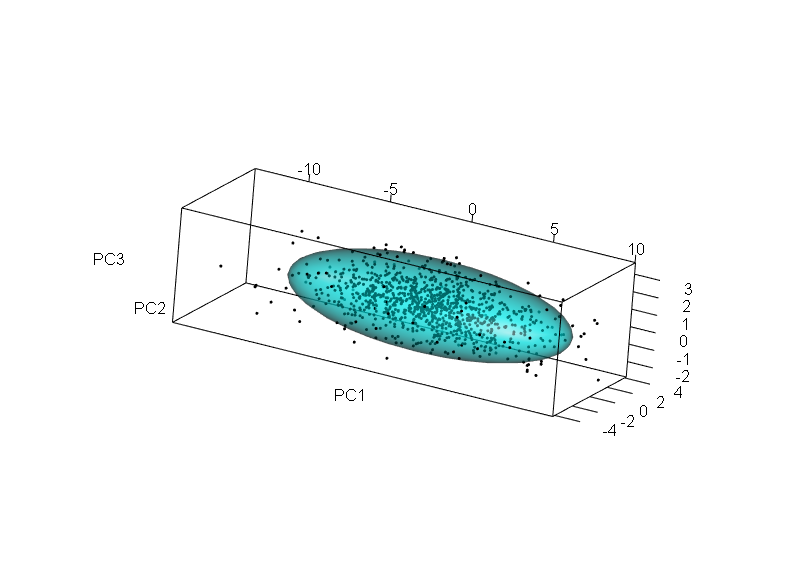
#3D Data

# =========================================  
 **#Due**

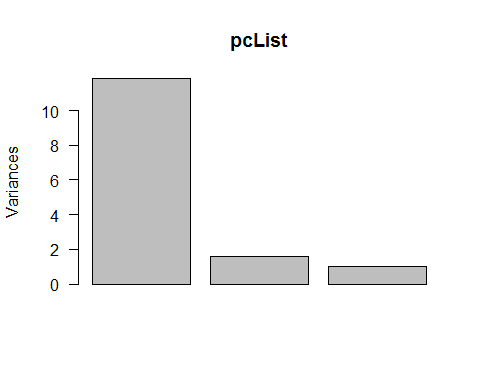
**Plots from 2**

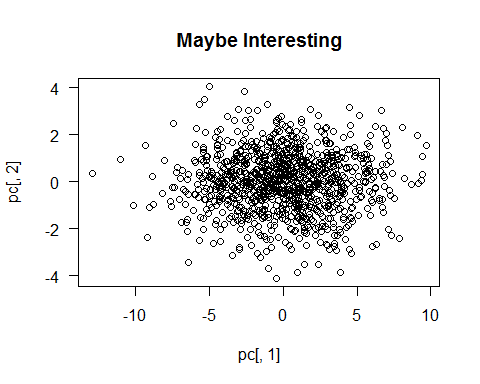
****

**Plots from 4**

****

**Plots from 8**

****

**9-first plot  
**

#==========================================  
  
#0. Setup  
library(MASS)  
library(rgl)  
  
#1. Generating random sample from a trivariate normal distribution  
  
Mean <- c(4,-2, 5)  
Mean

## [1] 4 -2 5

Cov <- matrix(c(3,3.5,0,3.5,10,0,0,0,1), 3,3)  
Cov

## [,1] [,2] [,3]  
## [1,] 3.0 3.5 0  
## [2,] 3.5 10.0 0  
## [3,] 0.0 0.0 1

set.seed(37)  
xyz <- mvrnorm(1000, Mean, Cov)  
  
sampMean <- apply(xyz,2,mean)   
sampMean

## [1] 4.016345 -2.074159 5.013260

Mean

## [1] 4 -2 5

sampMean - Mean

## [1] 0.01634452 -0.07415938 0.01325957

sampCov <- var(xyz)  
round(sampCov,2)

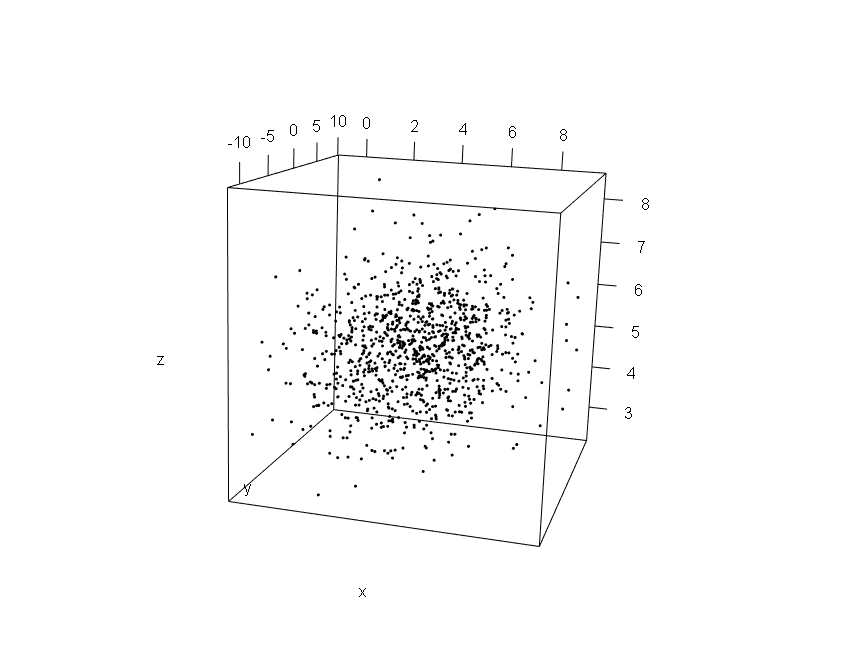
## [,1] [,2] [,3]  
## [1,] 2.96 3.47 0.08  
## [2,] 3.47 10.48 0.13  
## [3,] 0.08 0.13 1.00

Cov

## [,1] [,2] [,3]  
## [1,] 3.0 3.5 0  
## [2,] 3.5 10.0 0  
## [3,] 0.0 0.0 1

round(sampCov-Cov,2)

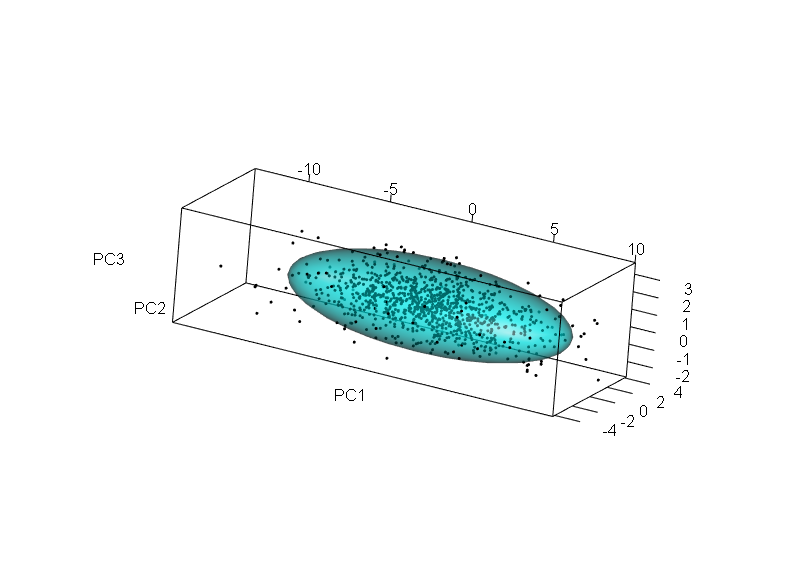
## [,1] [,2] [,3]  
## [1,] -0.04 -0.03 0.08  
## [2,] -0.03 0.48 0.13  
## [3,] 0.08 0.13 0.00

#2. A 3-D data scatterplot with a translucent ellipoid as a density reference.   
  
{open3d(FOV=0)  
plot3d(xyz, box=TRUE,  
 xlab="x", ylab="y", zlab="z")  
aspect3d("iso")  
xzyMean <- colMeans(xyz)  
xCov <- var(xyz)  
plot3d( ellipse3d(Cov,centre=Mean, level=.9),  
 col="green", alpha=0.5, add = TRUE)  
}  
  
snapshot3d("Data and 90% Ellipsoid.png" )  
  
#3. Producing principal components   
  
pcList <- prcomp(xyz)  
  
#4. A 3-D principal components plot with a reference ellipsoid.  
  
pc <- pcList$x  
pcMeans <- colMeans(pc)  
round(pcMeans,2)

## PC1 PC2 PC3   
## 0 0 0

pcCov <- var(pc)  
round(pcCov,2)

## PC1 PC2 PC3  
## PC1 11.84 0.0 0  
## PC2 0.00 1.6 0  
## PC3 0.00 0.0 1

{open3d(FOV=0)  
plot3d(pc, box=TRUE,  
xlab="PC1",ylab="PC2",zlab="PC3")  
aspect3d("iso")  
plot3d(ellipse3d(pcCov, centre=pcMeans, level=.9),  
 col="cyan", alpha=0.5, add = TRUE)  
}  
  
snapshot3d("Principle components and 90% Ellipsoid.png" )  
  
#5. Results returned by prcomp()  
  
pcList$center # Contains the means of the input variables that were subtracted to center the data

## [1] 4.016345 -2.074159 5.013260

pcList$rotation #Is the rotation used to rotate the case points about the origin.

## PC1 PC2 PC3  
## [1,] -0.3645034 0.93003685 -0.046569842  
## [2,] -0.9311016 -0.36474269 0.003554171  
## [3,] -0.0136805 0.04465676 0.998908713

pcList$sdev #has standard deviations of the principal components.

## [1] 3.440747 1.265712 1.000394

pcList$scale #is FALSE in this case since the default uses the covariance matrix.

## [1] FALSE

#pcList$x #contains the principal components as indicated above.

#6. A computation check using matrix multiplication   
  
rot <- pcList$rot  
rot

## PC1 PC2 PC3  
## [1,] -0.3645034 0.93003685 -0.046569842  
## [2,] -0.9311016 -0.36474269 0.003554171  
## [3,] -0.0136805 0.04465676 0.998908713

det(rot)

## [1] 1

xyzCentered <- scale(xyz,center=T,scale=FALSE)  
head(xyzCentered)

## [,1] [,2] [,3]  
## [1,] 0.6993544 0.2346239 -0.4748087  
## [2,] 1.4190001 0.8789844 -2.4189353  
## [3,] 0.3593754 2.0400320 -0.2808532  
## [4,] -0.7043213 -0.7167361 -1.6000431  
## [5,] 0.8853364 -3.3331926 -0.5141387  
## [6,] 0.9765611 -1.5556931 -0.1144950

pcCheck<- xyzCentered %\*% rot  
  
all.equal(pc, pcCheck)

## [1] TRUE

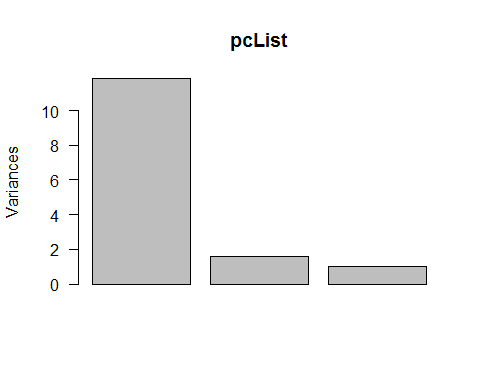
#7. Principal component interpretation in terms of rotation coefficients   
  
round(rot,2)

## PC1 PC2 PC3  
## [1,] -0.36 0.93 -0.05  
## [2,] -0.93 -0.36 0.00  
## [3,] -0.01 0.04 1.00

round(var(xyz),1)

## [,1] [,2] [,3]  
## [1,] 3.0 3.5 0.1  
## [2,] 3.5 10.5 0.1  
## [3,] 0.1 0.1 1.0

#8. Picking the number principal components to use  
screeplot(pcList, las=1)



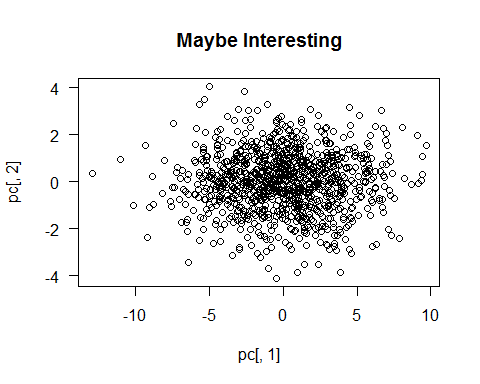
#9. 2-D scatterplots   
  
tmp <- apply(pc,2,range)  
tmp

## PC1 PC2 PC3  
## [1,] -12.955960 -4.108469 -2.783951  
## [2,] 9.686798 4.060475 3.565905

round(diff(tmp),1)

## PC1 PC2 PC3  
## [1,] 22.6 8.2 6.3

winX <- 7.5  
winY <- winX\*8.1/22.6   
windows(w=winX,h=winY)  
plot(pc[,1],pc[,2],las=T,  
main="Maybe Interesting")



windows()  
plot(pc[,1],pc[,2],las=T,  
main="Not Very Interesting")

