Homework 3

Cody Frisby

3/3/2017

3.1

I display the principal components as a matrix with the rows corresponding to the original variable names and the columns corresponding to the principal components.

	PC1	PC2	PC3	PC4	PC5	PC6	PC7
head length	-0.276	-0.365	0.882	-0.086	-0.067	0.005	-0.016
head breadth	-0.212	-0.639	-0.258	0.687	0.081	0.035	0.018
face breadth	-0.295	-0.512	-0.381	-0.699	-0.101	0.034	-0.075
Left finger lenght	-0.438	0.235	-0.070	0.102	-0.619	0.318	0.503
left forearm length	-0.456	0.277	-0.037	0.113	-0.039	0.290	-0.785
left foot lengther	-0.450	0.178	-0.059	0.053	-0.034	-0.870	0.014
height	-0.436	0.180	-0.006	-0.082	0.770	0.233	0.353

PC1 coefficients are all negative. This component could be the **overall size** component of the criminals. PC2 could be the **head size** component where head length having the largest influence while PC3 might be interpreted as the **head height** component, since *head breadth* and *face breadth* have the largest coefficients.

3.2

The test statistic proposed by Bartlett (1947)

$$\phi_0^2 = -\{n - \frac{1}{2}(q_1 + q_2 + 1)\} \sum_{i=1}^{s} log(1 - \lambda_i)$$

has a χ^2 distribution with $q_1 \times q_2$ degrees of freedom.

Applying this test statistic using `R`, the result for the *headsize* data set is

Bartlett	pValue
21.93926	0.0002061

and we would conclude that at least one of the canonical correlations is significant.

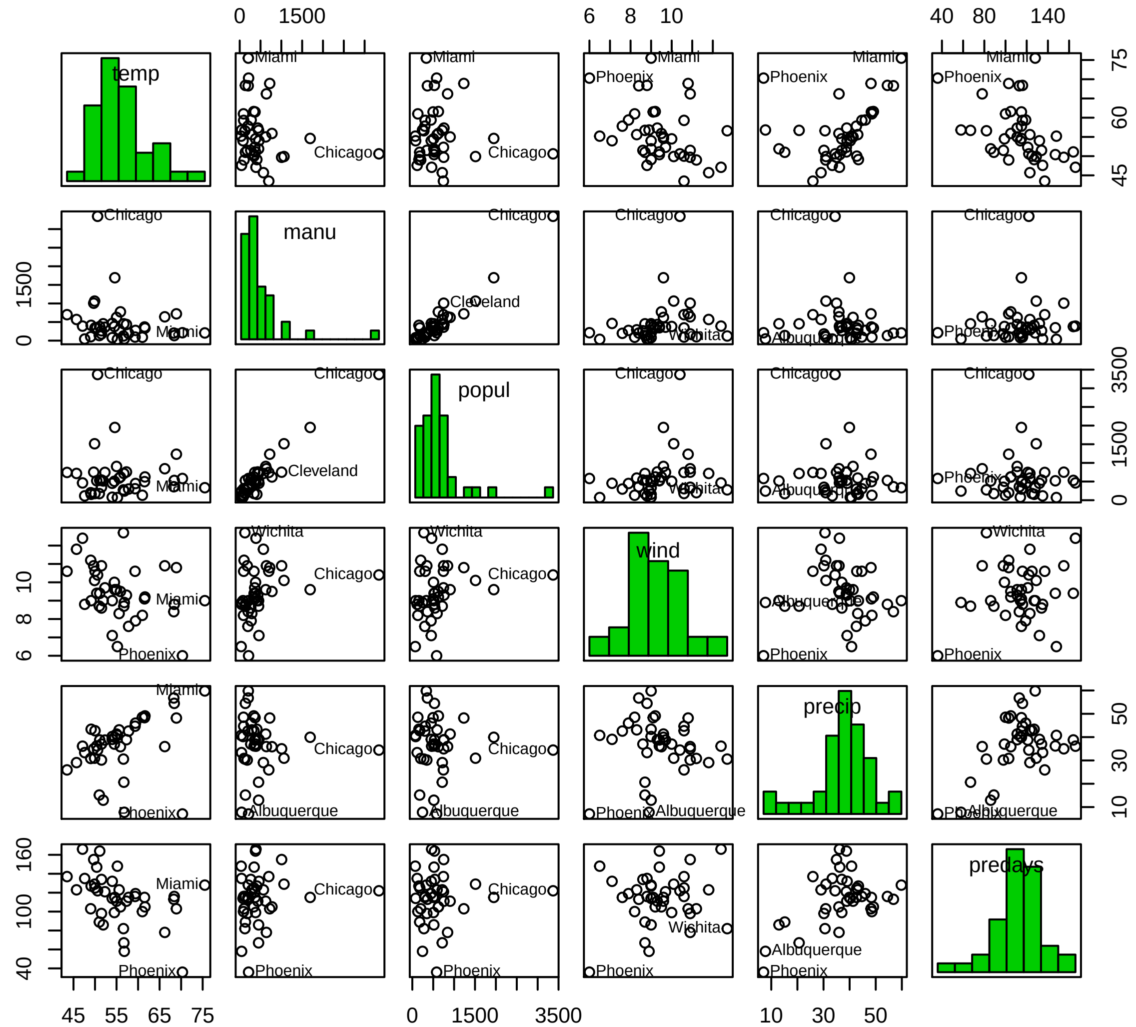
For the depression data from table 3.3, where we partition the data where **X** is the variables *CESD* and *Health* and **Y** is the variables *Gender*, *Age*, *Edu*, and *Income*, the result is

Bartlett	pValue
67.45031	0

and we would conclude that at least one of the canonical correlations is significant.

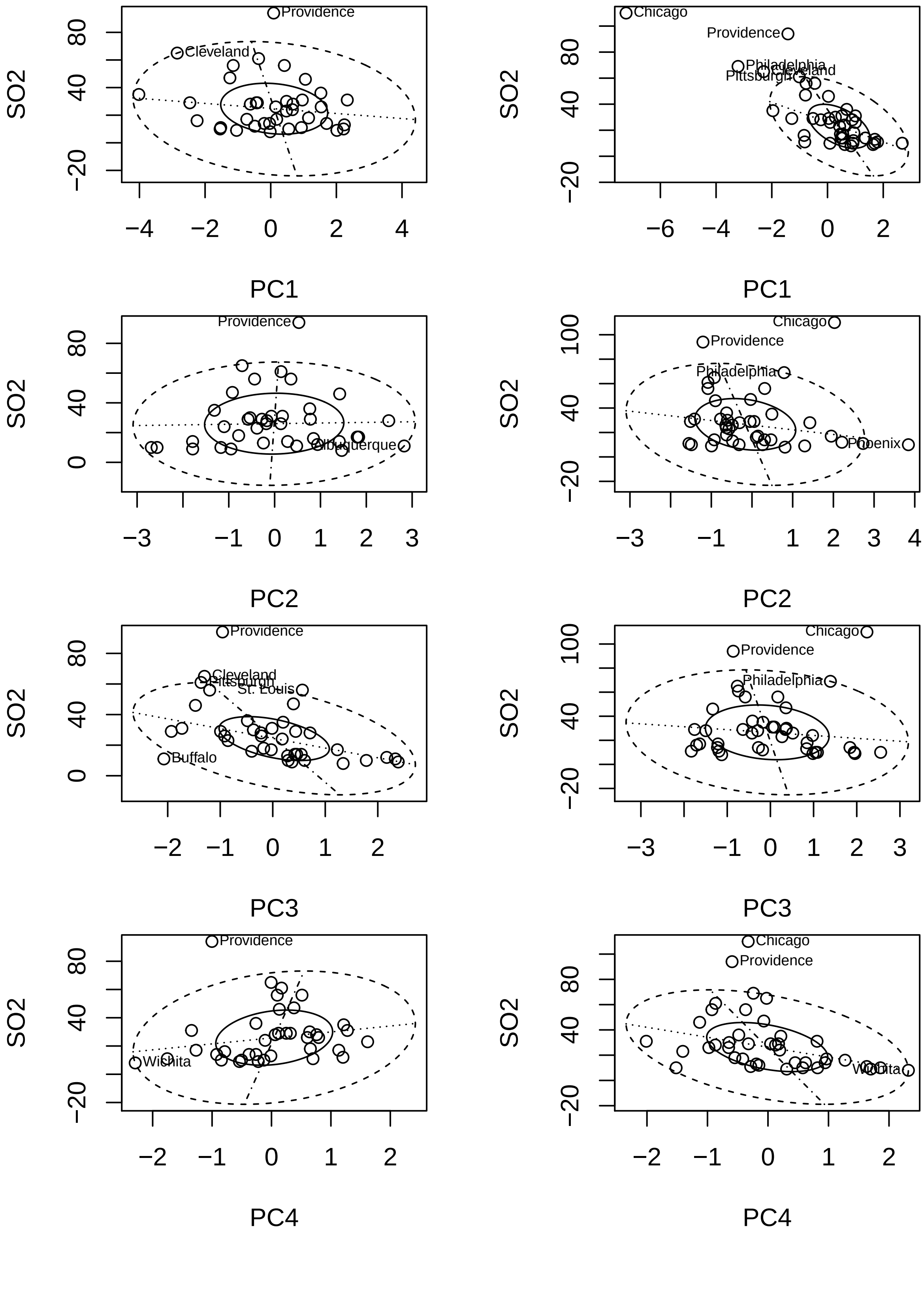
3.3

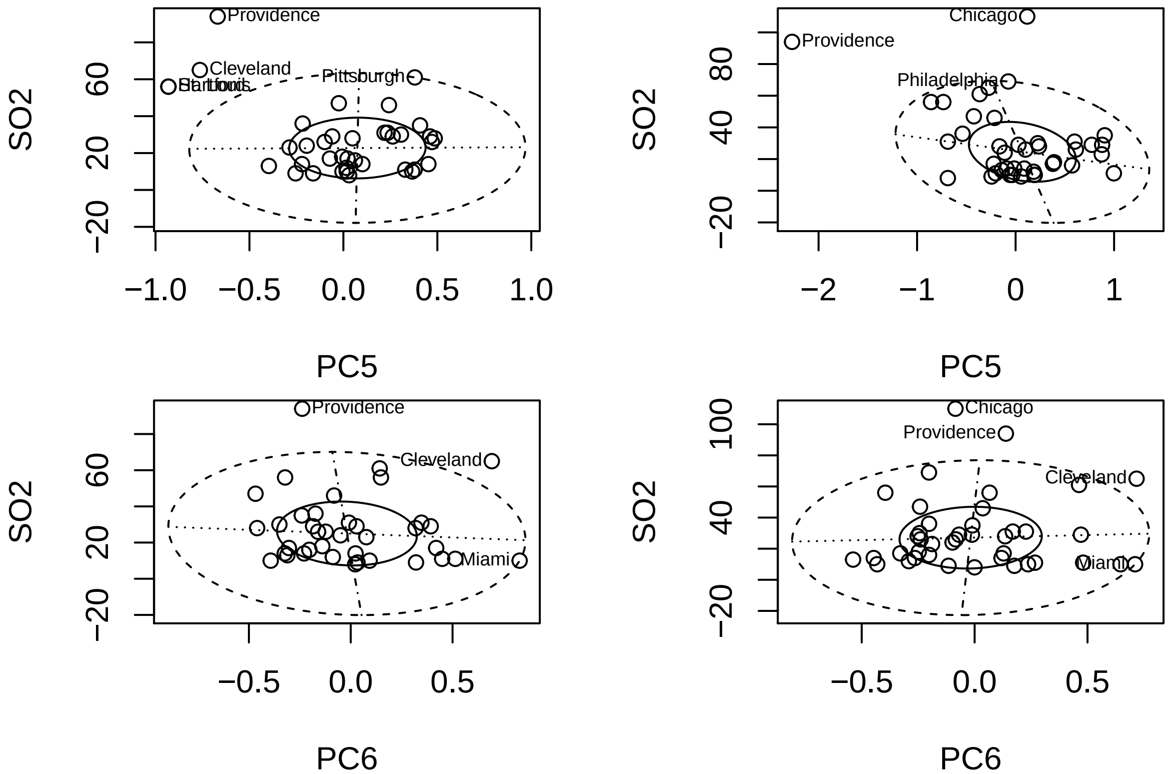
As in the book, I first take a look at independent variables via a scatter-plot matrix.



There appears to be some observations that could be considered outliers. Using the methods from the `bvbox` function, we can identify some of those outliers (here I've only labeled 2 per plot). Based on the analysis performed in chapter 2 homework **2.1** where I summarized the number of times a city was outside or on the outer ellipse, I'm going to proceed with excluding **Chicago**, **Philadelphia**, and **Phoenix**.

Below, I display side-by-side each of the *SO2* vs. PC_i without (left) and with the three outliers mentioned above.


Providence
Cleveland
SO2
PC1
Chicago
Providence
Philadelphia
Pittsburgh
Cleveland
SO2
PC1
Providence
Albuquerque
SO2
PC2
Chicago
Providence
Philadelphia
Phoenix
SO2
PC2
Providence
Cleveland
Pittsburgh
St. Louis
Buffalo
SO2
PC3
Chicago
Providence
Philadelphia
SO2
PC3
Providence
Wichita
SO2
PC4
Chicago
Providence
Wichita
SO2
PC4



After removing the three aforementioned cities we have a new "outlier" city based on the `bvbox` function methods, **Providence**. Also, it appears that plotting *SO2* by each principal component may change how you identify an outlier. I would definitely be considering **Providence** as a possible outlier now. Interestingly, most of the plots look similar before and after except for *SO2* vs. PC_1.

The explanatory ability of all the PC_i's definitely goes down when fitting a linear model than when they are included. In the book example, where no observations are excluded from the model, the R^2 value is around 0.67. Here, ours is around 0.50. I display the model coefficients below.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	27.4473684	2.394246	11.4638885	0.0000000
xPC1	-2.6465119	1.661894	-1.5924673	0.1214268
xPC2	0.8711375	1.920246	0.4536595	0.6532340
xPC3	-8.3750874	2.147333	-3.9002275	0.0004817
xPC4	2.7146185	2.796966	0.9705581	0.3392806
xPC5	-21.9910778	6.494201	-3.3862636	0.0019407
xPC6	-5.1978489	7.568584	-0.6867664	0.4973355

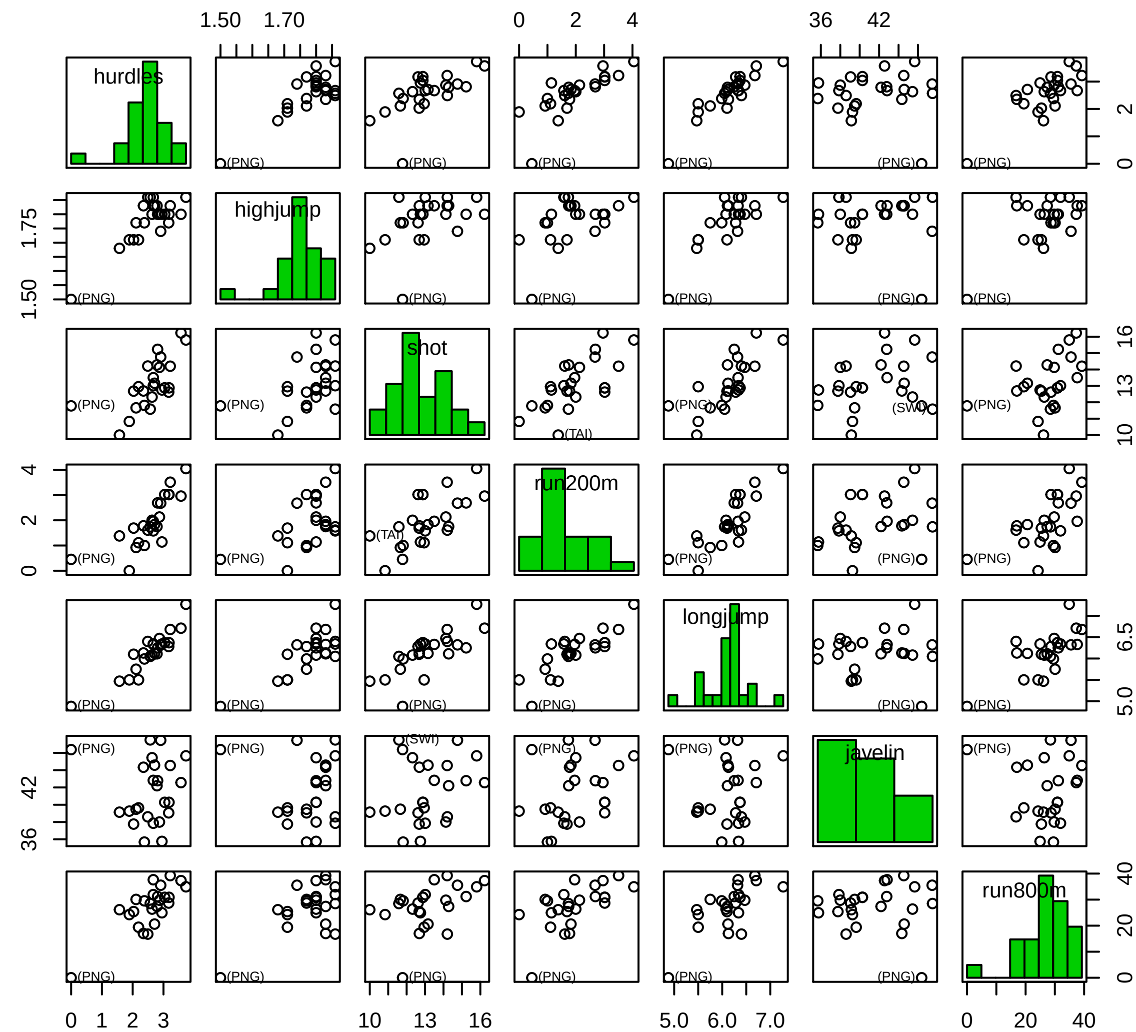
3.4

First, after transforming *hurdles*, *run200m*, and *run800m*, we take a look at the correlation matrix.

	hurdles	highjump	shot	run200m	longjump	javelin	run800m
hurdles	1.0000000	0.8114025	0.6513347	0.7737205	0.9121336	0.0077625	0.7792571
highjump	0.8114025	1.0000000	0.4407861	0.4876637	0.7824423	0.0021530	0.5911628
shot	0.6513347	0.4407861	1.0000000	0.6826704	0.7430730	0.2689888	0.4196196
run200m	0.7737205	0.4876637	0.6826704	1.0000000	0.8172053	0.3330427	0.6168101

	hurdles	highjump	shot	run200m	longjump	javelin	run800m
longjump	0.9121336	0.7824423	0.7430730	0.8172053	1.0000000	0.0671084	0.6995112
javelin	0.0077625	0.0021530	0.2689888	0.3330427	0.0671084	1.0000000	-0.0200491
run800m	0.7792571	0.5911628	0.4196196	0.6168101	0.6995112	-0.0200491	1.0000000

And here is the scatter plot matrix:



It can be seen that **(PNG)** is quite far from the rest of the pack on *highjump*, *hurdles*, *longjump*, and *run800m* to name a few.

Perfroming principal component analysis using `prcomp` with `scale. = TRUE` so that the eigen values will be computed based on the correlation matrix we get

	PC1	PC2	PC3	PC4	PC5	PC6	PC7
hurdles	-0.4528710	0.1579206	-0.0451500	0.0265387	-0.0949479	-0.7833410	0.3802471
highjump	-0.3771992	0.2480739	-0.3677790	0.6799917	0.0187989	0.0993998	-0.4339311
shot	-0.3630725	-0.2894074	0.6761892	0.1243172	0.5116520	-0.0508598	-0.2176249
run200m	-0.4078950	-0.2603855	0.0835921	-0.3610658	-0.6498340	0.0249564	-0.4533848
longjump	-0.4562318	0.0558739	0.1393165	0.1112925	-0.1842981	0.5902097	0.6120639
javelin	-0.0754090	-0.8416921	-0.4715602	0.1207992	0.1351067	-0.0272408	0.1729467

	PC1	PC2	PC3	PC4	PC5	PC6	PC7
run800m	-0.3749594	0.2244898	-0.3958567	-0.6034113	0.5043212	0.1555552	-0.0983096

which are each of the principal component coefficients for each of the original variables.

For example, the linear function for the first principal component, Y_1 would be

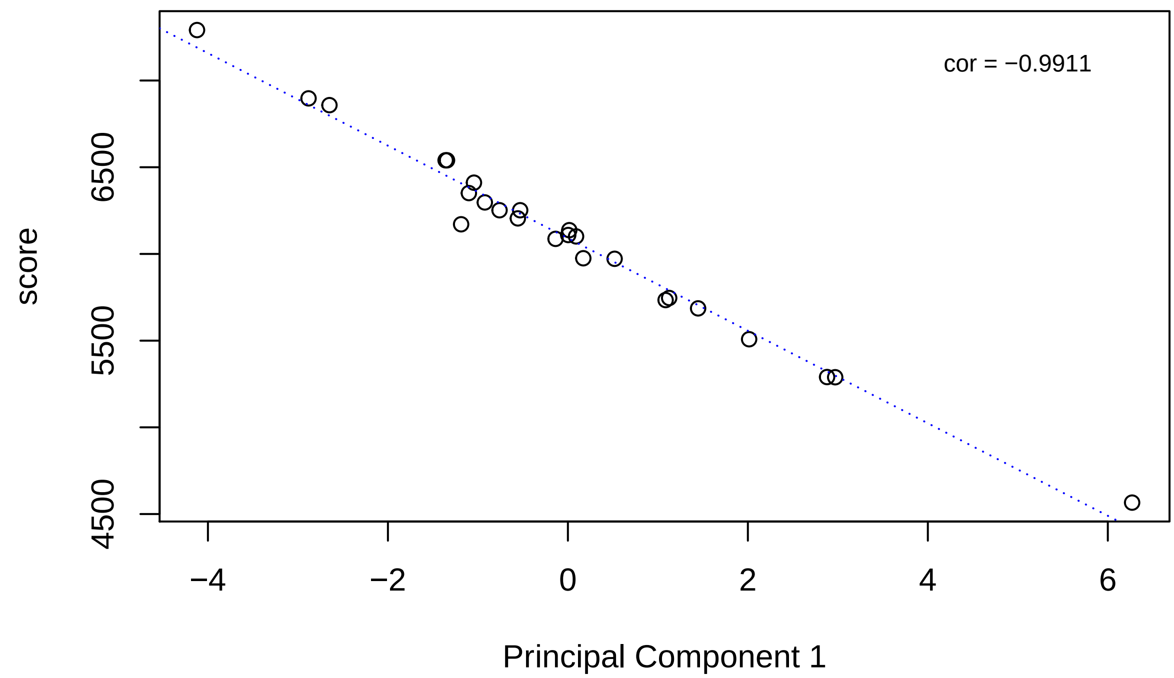
$$Y_1 = -0.4529X_1 - 0.3772X_2 - 0.3631X_3 - 0.4079X_4 - 0.4562X_5 - 0.0754X_6 - 0.375X_7.$$

where

$$X_1 = hurdles, X_2 = highjump, X_3 = shot, X_4 = run200m, X_5 = longjump, X_6 = javelin, X_7 = run800m$$

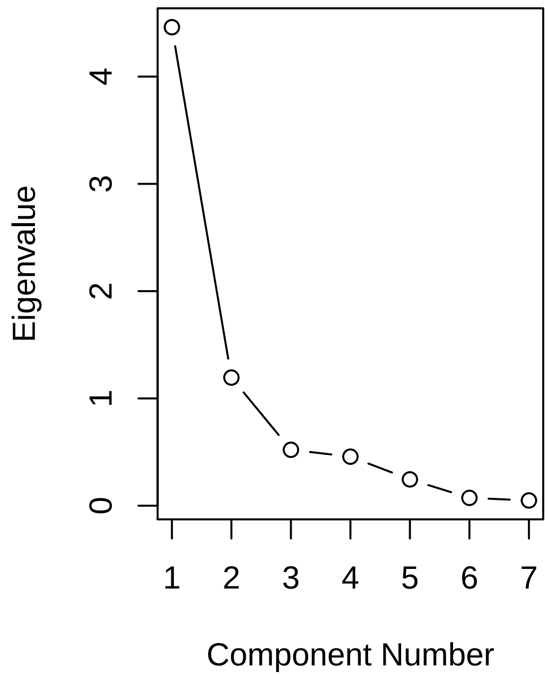
All the signs on the coefficients are negative here, which intuitely may not make sense for Y_1 since this is the principal component associated with the largest variance and we may think of *score* being positively associated with all the X_i's. But this is due to the arbitrainess of the signs in principal component analysis and would have no effect on the predictive ability of Y_1 on the response variable *score*.

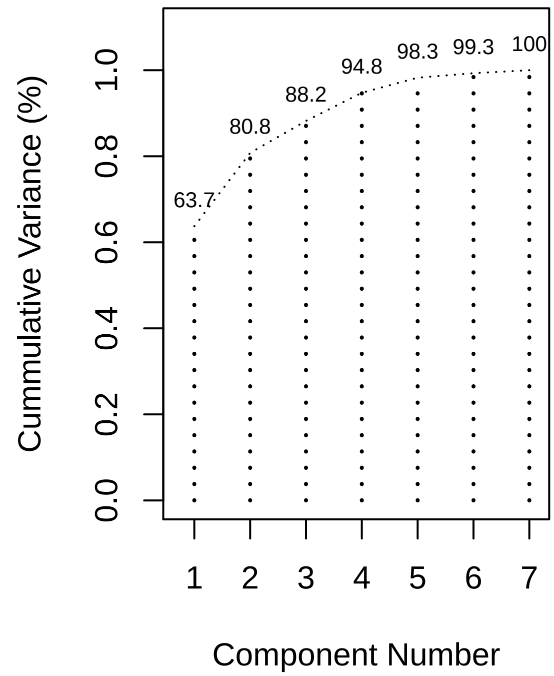
A plot of *score* vs. Y_1 illustrates the strong correlation between the two.



And we can see that they have a very high correlation, -0.9911.

A plot of the variances and the cummulative variances of all principal components is displayed as well.





These plots illustrate the relative variances of all 7 principal components. We can see that the first 2 principal components contain over 80 percent of the variance. Depending on the research question, we may want to continue with 2 - 3 principal components rather than all 7.

3.5

For this problem I define **X** as *breaklength*, *elasticmod*, *stressfail*, and *burststren* and **Y** as *arithmeticleng*, *longfractin*, *finefaraction*, and *zerotensile*.

a)

The problem does not specify which test to use to determine the number of significant cannonical variates. Using a function I wrote that returns similar results to `PROC cancorr` in SAS, there are **two** significant canonical variate pairs. Using Bartlett's test we would conclude similarly. The below table displays all variate pairs correlation and the corresponding Wilks Lambda test results.

rho	WilksL	F	df1	df2	p
0.9173293	0.0485989	17.5022188	16	165.6104	0.000000
0.8169269	0.3066043	9.3119381	9	134.0062	0.000000
0.2653854	0.9217567	1.1641878	4	112.0000	0.330513
0.0916840	0.9915940	0.4832014	1	57.0000	0.489800

Essentially what this means is that the correlation between the two sets of variables is significant for cannonical variate pairs 1 and 2 but not for pairs 3 and 4.

b)

All 4 canonical variates are

$$(paper_1, fiber_1), (paper_2, fiber_2), (paper_3, fiber_3), (paper_4, fiber_4)$$

These can be calculated multiplying the cannonical coefficients for **X** which will return the raw cannonical coefficients.

	paper1	paper2	paper3	paper4
breaklength	0.5224426	-1.2131371	1.978681	1.7646646
elasticmod	0.2957899	-2.1536462	-4.920057	0.8188946
stressfail	-1.3660328	0.7355096	3.222027	-4.1489040
burststren	-0.9760405	5.4369966	-10.321875	0.9899703

by each of our x values. Similar procedure goes for **Y**.

	fiber1	fiber2	fiber3	fiber4
arithmeticleng	0.6383772	2.7594527	2.0556736	-9.3494101
longfractin	-0.0425405	0.0674576	-0.0051920	0.1466427
finefaraction	-0.0185013	0.0002853	0.0947041	-0.0012658
zerotensile	-27.7331245	-52.9592008	26.4000888	-3.0226090

We can standardize the raw cannonical coefficients by multiplying them by the square root of the variances of the corresponding **X** variables.

	paper1	paper2	paper3	paper4
breaklength	1.5054030	-3.495619	5.701510	5.0848285
elasticmod	0.2119308	-1.543068	-3.525176	0.5867306
stressfail	-1.9983550	1.075969	4.713469	-6.0693882
burststren	-0.6764123	3.767929	-7.153231	0.6860659

And for those corresponding to **Y**

	fiber1	fiber2	fiber3	fiber4
arithmeticleng	0.1592987	0.6885854	0.5129665	-2.3330231
longfractin	-0.6324836	1.0029461	-0.0771936	2.1802555
finefaraction	-0.3249077	0.0050107	1.6631282	-0.0222295
zerotensile	-0.8178993	-1.5618612	0.7785857	-0.0891421

c)

Below I display the correlations between the cannonical variables and the observed variables for both sets.

	paper1	paper2	paper3	paper4
breaklength	-0.9350634	-0.1260513	0.0533839	0.3269827
elasticmod	-0.8869185	-0.4279727	-0.1305587	0.1147575
stressfail	-0.9767068	-0.1453158	0.0306675	0.1548759
burststren	-0.9517945	0.0146863	-0.0126989	0.3061214

	fiber1	fiber2	fiber3	fiber4
arithmeticleng	-0.8165891	0.3683249	-0.1661476	-0.4122062
longfractin	-0.9055935	0.3848247	-0.1779068	0.0126294
finefaraction	0.6496328	0.0122691	0.7309408	0.2086915
zerotensile	-0.9394546	-0.2307228	-0.1851478	-0.1729519

We can see large values for all **X** variables and **paper1**. There is strong linear relationship between all **X** variables and **paper1**.

d)

To summarize, the linear function for cannonical variate u_1 would be

$$paper_1 = 0.5224X_{breaklength} + 0.2958X_{elasticmod} - 1.366X_{stressfail} - 0.976X_{burststren}$$

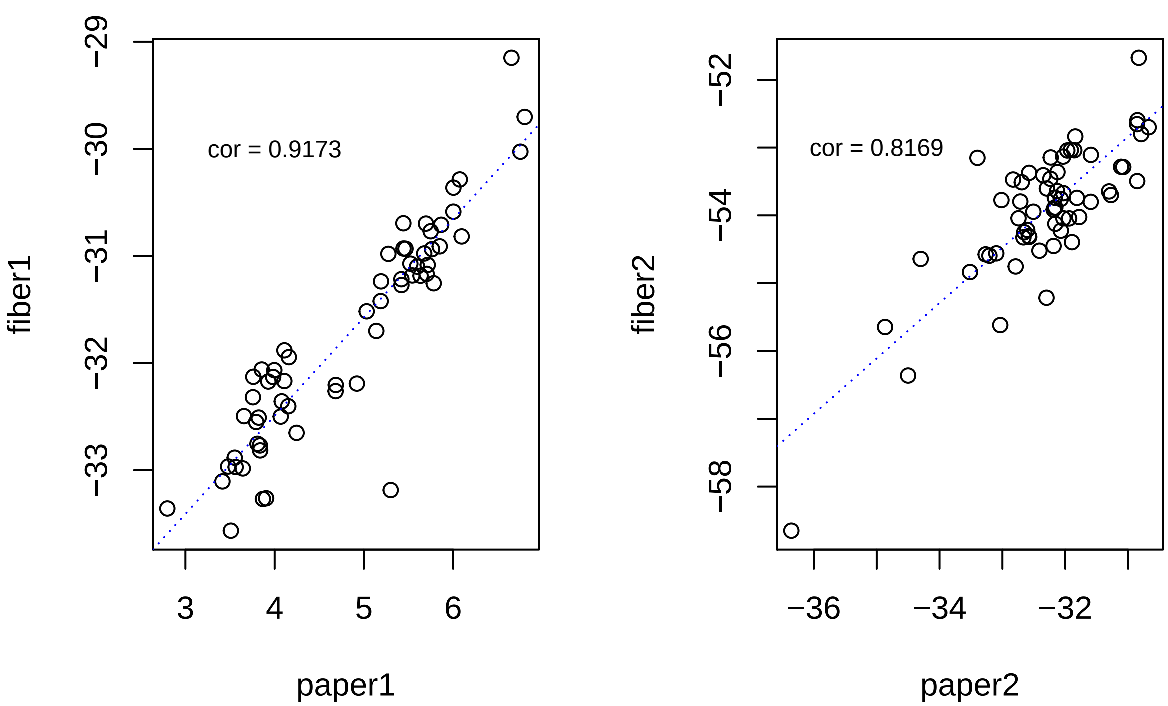
and

$$fiber_1 = 0.6384X_{arithmeticleng} + 0.0425X_{longfractin} - 0.0185X_{finefaraction} - 27.7331X_{zerotensile}$$

resulting in

$$Cor(u_1, v_1) = 0.9173293.$$

For reference I plot the linear relationships of $paper_1$ vs. $fiber_1$ and $paper_2$ vs. $fiber_2$ where it can be seen that they are strongly correlated with eachother.



R code:

```
#####
# 3.1
# create the appropriate matrix
r <- matrix(ncol = 7, nrow = 7)
diag(r) <- 1
x <- c(.402,.396,.301,.305,.339,.34,.618,.15,.135,.206,.183,
       .321,.289,.363,.345,.846,.759,.661,.797,.8,.736)
r[lower.tri(r)] <- x # fill the lower diagonal
r[upper.tri(r)] <- t(r)[upper.tri(r)] # fill the upper diag
e <- eigen(r)
pc <- prcomp(r); pcnames <- colnames(pc$rotation)
rm(pc)
# original variable names, for reference
vars <- c("head length", "head breadth", "face breadth",
              "Left finger lenght", "left forearm length",
              "left foot lengther", "height")
# clean way to display the principal comps with the
# original corresponding variables.
m <- matrix(data = e$vectors, ncol = 7)
colnames(m) <- pcnames
rownames(m) <- vars
knitr::kable(round(m, 3))
######
# 3.2
# headsize data that I'm going to use my function on:
df <- read.csv("~/Documents/STAT4400/data/headsize.csv")
x <- cbind(df$head1, df$breadth1)
y <- cbind(df$head2, df$breadth2)
# writing my own function to return test statistic
cc.test <- function(x,y){
  n <- dim(x)[1]
  q1 <- dim(x)[2]
  q2 <- dim(y)[2]
  lam <- cancor(x,y)$cor^2
  STAT <- -1 * (n - 0.5 * (q1 + q2 + 1)) * sum(log(1 - lam))
  p <- 1 - pchisq(STAT, df = q1 * q2)
  return(cbind(Bartlett = STAT, pValue = p))
}
knitr::kable(cc.test(x,y))
# note: the depression data is already the correlation matrix
# so I won't be using the function I wrote above.
df <- read.csv("~/Documents/STAT4400/data/LAdepr.csv")
df <- as.matrix(df)
r11 <- df[1:2, 1:2]
r22 <- df[-(1:2), -(1:2)]
r12 <- df[1:2, -(1:2)]
r21 <- df[-(1:2), 1:2]
E2 <- solve(r22) %*% r21 %*% solve(r11) %*% r12
e2 <- eigen(E2)
lam <- e2$values
### borrowing from the above function:
q1 <- 2; q2 <- 4; n <- 294
STAT <- -1 * (n - 0.5 * (q1 + q2 + 1)) * sum(log(1 - lam))
```

```r
p <- 1 - pchisq(STAT, df = q1 * q2)
# print the results "pretty" knitting a PDF.
knitr::kable(cbind(Bartlett = STAT, pValue = p))
######
# 3.3
# scatterplot matrix, labeling a few outliers per plot.
id <- USairpollution$X
car::scatterplotMatrix(USairpollution[-(1:2)], diagonal = "histogram", smoother = NULL, reg.line = NULL
# we need to fit a linear model, excluding the "outliers" regresssing the PCs onto SO2
# here I subset the USairpollution data set, excluding those 3 citys.
df <- USairpollution[!USairpollution$X %in%
          c("Chicago", "Philadelphia", "Phoenix"), -1]
pc_air <- prcomp(df[-1], scale. = TRUE)
pc_air2 <- prcomp(USairpollution[-1], scale. = TRUE)
x <- pc_air$x # principal components rotation matrix
y <- pc_air2$x
id <- USairpollution$X[as.numeric(rownames(df))]
par(mfrow = c(1,2))
for (i in 1:6) { bvbox(cbind(x[,i], df$SO2),
    xlab = colnames(x)[i],
    ylab = "SO2", labels = id)
  MVA::bvbox(cbind(y[,i], USairpollution$SO2),
    xlab = colnames(y)[i],
    ylab = "SO2", labels = USairpollution$X)
}
# fit a linear model using the PC as predictors
fit <- lm(SO2 ~ x, df)
temp <- summary(fit)
######
# 3.4
df <- read.csv("~/Documents/STAT4400/data/heptathlon.csv")
id <- df$X
df <- df[-1]
# first we need to alter some of the variables:
# for hudles, 200m and 800m, shorter is better.  But for
# the others, longer is better.  Let's get them all going in
# the same direction.
df$hurdles <- max(df$hurdles) - df$hurdles
df$run200m <- max(df$run200m) - df$run200m
df$run800m <- max(df$run800m) - df$run800m
# how correlated are some of these variables?
knitr::kable(cor(df[-8]))
# scatterplot
# shorten our ID variable, keeping the country
temp <- strsplit(as.character(id), " ")
id <- unlist(temp)[c(FALSE, TRUE)]
# scatterplot matrix
car::spm(df[-8], diagonal = "histogram", smoother = NULL,
         reg.line = NULL, labels = id, id.method = "mahal",
         id.n = 1, id.cex = 0.6, cex.labels = 1)
# perform PCA analysis
A <- cor(df[-8]) # correlation matrix
E <- eigen(A) # egien values/vectos of correlation matrix
```

```r
pc <- prcomp(df[-8], scale. = TRUE) # R principal component function.
# scale. = TRUE so that we compute based on the correlation matrix.
y1 <- pc$x[,1]
plot(y1, df$score, xlab = "Principal Component 1",
     ylab = "score")
text(5, 7100, paste("cor =", round(cor(y1, df$score), 4)), cex = 0.75)
abline(lm(df$score ~ y1), lty = 3, col = "blue")
# Scree Diagram
par(mfrow = c(1,2))
plot(pc$sdev^2, type = "b", xlab = "Component Number",
     ylab = "Eigenvalue")
# plotting the cummulative variance of the principal components
plot(cumsum(pc$sdev)/sum(pc$sdev), ylim = c(0, 1.1), type = "h",
     xlab = "Principal Component", ylab = "Cummulative Variance",
     lty = 3, xlim = c(0.9, 7), lwd = 2)
text(1:7, cumsum(pc$sdev)/sum(pc$sdev), pos = 3,
     paste(round(cumsum(pc$sdev)/sum(pc$sdev), 3)*100,"%",
           sep = ""), cex = 0.75)
lines(1:7, cumsum(pc$sdev)/sum(pc$sdev), lty = 3)
########
# 3.5
#####
# part a
rm(list = ls())
# property data
df <- read.csv("~/Documents/STAT4400/data/propertydata.csv")
# first we should standardize all the data.
# create x and y
x <- cbind(df$breaklength, df$elasticmod, df$stressfail,
           df$burststren)
colnames(x) <- colnames(df)[1:4]
y <- cbind(df$arithmeticleng, df$longfractin,
           df$finefaraction, df$zerotensile)
colnames(y) <- colnames(df)[5:8]
# function to determine significant canonical correlations:
cc1 <- CCA::cc(x,y)
source("~/Documents/STAT4400/R/cc.wilks.R")
temp <- cc.wilks(x, y) # test for significance
knitr::kable(temp) # for pretty printing
#####
# part b
temp <- cc1$xcoef # basically the rotation matrix for x
colnames(temp) <- paste("paper", 1:4, sep = "")
knitr::kable(temp) # for pretty priinting
temp <- cc1$ycoef # rotation matrix fo y
colnames(temp) <- paste("fiber", 1:4, sep = "")
knitr::kable(temp) # pretty printing
# standardize the cannonical coefs
rows <- row.names(var(x))
s1 <- sqrt(diag(diag(var(x))))
temp <- s1 %*% cc1$xcoef
colnames(temp) <- paste("paper", 1:4, sep = "")
row.names(temp) <- rows
```

```r
knitr::kable(temp)
#####
# part c
temp <- cc1$scores$corr.X.xscores # cor(X, paper)
colnames(temp) <- paste("paper", 1:4, sep = "")
knitr::kable(temp) # pretty printing
temp <- cc1$scores$corr.Y.yscores  # cor(Y, fiber)
colnames(temp) <- paste("fiber", 1:4, sep = "")
knitr::kable(temp) # pretty printing
# coeficients for u1
b <- round(abs(cc1$xcoef[,1]), 4)
# coefficients for v1
b <- round(abs(cc1$ycoef[,1]), 4)
######
# part d
# side by side plots
par(mfrow = c(1, 2))
# plot the cannonical variates for u1, v1:
bx1 <- as.matrix(cc1$xcoef[,1])
by1 <- as.matrix(cc1$ycoef[,1])
u1 <- x %*% bx1
v1 <- y %*% by1
plot(u1, v1, xlab = "paper1", ylab = "fiber1")
text(4, -30, paste("cor =", round(cor(u1, v1), 4)), cex = 0.75)
abline(lm(v1 ~ u1), lty = 3, col = "blue")
# plot the cannonical variates for u2, v2
bx2 <- as.matrix(cc1$xcoef[,2])
by2 <- as.matrix(cc1$ycoef[,2])
u2 <- x %*% bx2
v2 <- y %*% by2
plot(u2, v2, xlab = "paper2", ylab = "fiber2")
text(-35, -53, paste("cor =", round(cor(u2, v2), 4)), cex = 0.75)
abline(lm(v2 ~ u2), lty = 3, col = "blue")
```