Principal Component Analysis Notes

Linear Combination

In PCA, we are looking for a linear combination of variables of the matrix X (assumed correlated) that are

1. uncorrelated with each other (orthogonal) and that
2. maximize a reduction in the order of the variance of the original matrix.

In fact, this is identical to maximizing the R^2 in a regression model.

Y is "latent"

In PCA, we do not know what Y, our new matrix, will be. It is latent. We build it solely from making a linear combination of X that satisfies the two conditions above.

On the Iris Dataset

We typically have a data matrix of n observations on p correlated variables $x_1, x_2, ...x_p$. PCA looks for a transformation of x_i on p new variables y_i that are uncorrelated.

We parse the variance.

```
set.seed(1337)

data <- iris[,1:4]

# Correlation:
cor(data)
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    1.0000000  -0.1175698    0.8717538   0.8179411
## Sepal.Width    -0.1175698   1.0000000   -0.4284401  -0.3661259
## Petal.Length    0.8717538  -0.4284401    1.0000000   0.9628654
## Petal.Width     0.8179411  -0.3661259    0.9628654   1.0000000
```

We are looking for a transformation of the data matrix $X(n \times p)$ such that:

$Y = a^T$ and $X = a_1X_1 + a_2X_2 + ... + a_pX_p$

where $a = (a_1, a_2, ..., a_p)^T$ is a column vector.

New variables Y_i are linear combination of the original variables $x_i : Y_i = a_{i1}x_1 + a_{i2}x_2 + ... + a_{ip}x_p$

The variables Y_i are derived in decreasing order of importance, they are called principal components.

This is sensitive to scale! Always use it with standardized data! Use correlation matrix, NOT covariance matrix.

The new variables have a variance equal to their corresponding eigenvalue:

$$Var(Y_i) = \lambda_i$$

for all $i = 1, 2, ..., p$

A small λ_i is small variance! Data changes little in the Y_i direction.

The relative variance explained by each principal component is:

$$\lambda_i / \Sigma\lambda_i$$

How many Y_i to keep?

Enough principal components to have a cumulative variance explained by them. 70%? 90%? It's an art...

Retain those Y_i that have eigenvalues greater than average variance.

Kaiser criterion: keep PCs with eigenvalues > 1

Jollife criterion: keep PCs with eigenvalues $> .7$

Scree plot: represents the ability of PCs to explain the variation in the original dataset.

Generating the Principal Components

Maximize the variance of the projection of the observations on the Y variables. Remember:

$$Var(aX) = a^2Var(X)$$

$$Max_{a,l}L1 = a_1^T C a_1 - \lambda_1(a_1^T a_1 - 1)$$
$$Max_{a,l}L2 = a_2^T C a_2 - \lambda_2(a_2^T a_2 - 1) - \lambda_1(a_1^T a_2 - 0)$$

The matrix $C = Var(X)$ is the covariance matrix of the X_i variables. Remember, the covariance matrix is not standardized, and any analysis using it is NOT invariant to scale. Using the correlation matrix makes more sense, normally, and the covariance / correlation matrices of standardized variables are... the same!

```r
cor(data)
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    1.0000000  -0.1175698    0.8717538   0.8179411
## Sepal.Width    -0.1175698   1.0000000   -0.4284401  -0.3661259
## Petal.Length    0.8717538  -0.4284401    1.0000000   0.9628654
## Petal.Width     0.8179411  -0.3661259    0.9628654   1.0000000
```

```r
cov(data)
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    0.6856935  -0.0424340    1.2743154   0.5162707
## Sepal.Width    -0.0424340   0.1899794   -0.3296564  -0.1216394
## Petal.Length    1.2743154  -0.3296564    3.1162779   1.2956094
## Petal.Width     0.5162707  -0.1216394    1.2956094   0.5810063
```

```r
(myPCA <- princomp(data,
                   scores = TRUE,
                   cor = TRUE))
```

```
## Call:
## princomp(x = data, cor = TRUE, scores = TRUE)
##
## Standard deviations:
##    Comp.1    Comp.2    Comp.3    Comp.4
## 1.7083611 0.9560494 0.3830886 0.1439265
##
##  4  variables and  150 observations.
```

```r
(myPCA$scores)
```

```
##             Comp.1       Comp.2       Comp.3       Comp.4
##   [1,] -2.26470281  0.480026597  0.127706022  0.024168204
##   [2,] -2.08096115 -0.674133557  0.234608854  0.103006775
##   [3,] -2.36422905 -0.341908024 -0.044201485  0.028377053
##   [4,] -2.29938422 -0.597394508 -0.091290106 -0.065955560
##   [5,] -2.38984217  0.646835383 -0.015738196 -0.035922813
##   [6,] -2.07563095  1.489177523 -0.026968294  0.006608180
##   [7,] -2.44402884  0.047644198 -0.335470401 -0.036775557
##   [8,] -2.23284716  0.223148073  0.088695498 -0.024612096
##   [9,] -2.33464048 -1.115327675 -0.145076864 -0.026859221
##  [10,] -2.18432817 -0.469013561  0.253765567 -0.039899288
##  [11,] -2.16631010  1.043690653  0.268681102  0.016731367
##  [12,] -2.32613087  0.133078335 -0.093759244 -0.133483413
##  [13,] -2.21845090 -0.728676165  0.230911237  0.002425038
##  [14,] -2.63310070 -0.961506729 -0.180796084 -0.019215534
##  [15,] -2.19874060  1.860057113  0.472900998  0.194731769
##  [16,] -2.26221453  2.686284485 -0.030526609  0.050533737
##  [17,] -2.20758770  1.483609363  0.005344094  0.188817432
##  [18,] -2.19034951  0.488838316  0.044215316  0.093090438
##  [19,] -1.89857200  1.405018794  0.374343275  0.061095967
```

```
##  [20,] -2.34336905  1.127849382 -0.132630467 -0.037756420
##  [21,] -1.91432300  0.408855708  0.421292594  0.010921286
##  [22,] -2.20701284  0.924121427 -0.159865277  0.059597330
##  [23,] -2.77434470  0.458343668 -0.332179098  0.019648430
##  [24,] -1.81866953  0.085558526 -0.034488596  0.151140999
##  [25,] -2.22716331  0.137254455 -0.117993536 -0.270140352
##  [26,] -1.95184633 -0.625618588  0.305640982  0.043561651
##  [27,] -2.05115137  0.242163553 -0.086364011  0.067680060
##  [28,] -2.16857717  0.527149525  0.206816248  0.010275393
##  [29,] -2.13956345  0.313217810  0.271150240  0.084259221
##  [30,] -2.26526149 -0.337731904 -0.068435776 -0.108279885
##  [31,] -2.14012214 -0.504540690  0.075008442 -0.048188868
##  [32,] -1.83159477  0.423695068  0.270467377  0.239870381
##  [33,] -2.61494794  1.793575856 -0.047228419 -0.229235932
##  [34,] -2.44617739  2.150727877  0.082668045 -0.048214393
##  [35,] -2.10997488 -0.460201841  0.170274861  0.029022947
##  [36,] -2.20780890 -0.206107398  0.225441580  0.168907873
##  [37,] -2.04514621  0.661558111  0.484537410  0.196358525
##  [38,] -2.52733191  0.592292774 -0.019435812 -0.136504550
##  [39,] -2.42963258 -0.904180040 -0.193254662 -0.009738423
##  [40,] -2.16971071  0.268878961  0.175883821  0.007047406
##  [41,] -2.28647514  0.441715388 -0.034894909  0.106983249
##  [42,] -1.85812246 -2.337415158  0.204234223  0.289863919
##  [43,] -2.55363840 -0.479100690 -0.305766453 -0.066601453
##  [44,] -1.96444768  0.472326668 -0.309601318  0.177093014
##  [45,] -2.13705901  1.142229262 -0.248433561 -0.151043437
##  [46,] -2.06974430 -0.711052725  0.063929826  0.140269507
##  [47,] -2.38473317  1.120429702 -0.057217858 -0.152230967
##  [48,] -2.39437631 -0.386246873 -0.139467905 -0.048834762
##  [49,] -2.22944655  0.997959764  0.181492780 -0.014928135
##  [50,] -2.20383344  0.009216358  0.153029490  0.049371732
##  [51,]  1.10178118  0.862972418  0.684586163  0.034833776
##  [52,]  0.73133743  0.594614726  0.094121716  0.004903623
##  [53,]  1.24097932  0.616297654  0.554006835  0.009423397
##  [54,]  0.40748306 -1.754403989  0.023101768  0.065768835
##  [55,]  1.07547470 -0.208421046  0.398255523  0.104736873
##  [56,]  0.38868734 -0.593283636 -0.124191550 -0.240831300
##  [57,]  0.74652974  0.773019312 -0.148969403 -0.077369785
##  [58,] -0.48732274 -1.852429087 -0.249265266 -0.040520205
##  [59,]  0.92790164  0.032226078  0.596169361 -0.029879609
##  [60,]  0.01142619 -1.034018275 -0.538899390 -0.028461184
##  [61,] -0.11019628 -2.654072819  0.046790444  0.013760731
##  [62,]  0.44069345 -0.063295188 -0.205073815  0.040126082
##  [63,]  0.56210831 -1.764724381  0.765771394  0.045731157
##  [64,]  0.71956189 -0.186224606  0.068658945 -0.164807198
##  [65,] -0.03335470 -0.439003210 -0.194932893  0.109048499
##  [66,]  0.87540719  0.509063957  0.503511382  0.104943723
##  [67,]  0.35025167 -0.196311735 -0.490873075 -0.191509364
##  [68,]  0.15881005 -0.792095742  0.302037174 -0.205297735
##  [69,]  1.22509363 -1.622243803  0.482304024  0.225899769
##  [70,]  0.16491790 -1.302609230  0.172837808 -0.051726849
##  [71,]  0.73768265  0.396571562 -0.616526306 -0.083284123
##  [72,]  0.47628719 -0.417320281  0.264952227  0.113568273
##  [73,]  1.23417810 -0.933325729  0.368412272 -0.009944526
```

```
##  [74,]  0.63285820 -0.416387721  0.291896252 -0.274220152
##  [75,]  0.70266118 -0.063411820  0.446027008  0.043458325
##  [76,]  0.87427365  0.250793393  0.472578954  0.101715736
##  [77,]  1.25650912 -0.077256020  0.727155002  0.039688518
##  [78,]  1.35840512  0.331311682  0.260826577  0.066828064
##  [79,]  0.66480037 -0.225927855 -0.085863889 -0.036439840
##  [80,] -0.04025861 -1.058718547  0.319573330  0.064788156
##  [81,]  0.13079518 -1.562271834  0.149983478 -0.009402523
##  [82,]  0.02345269 -1.572475594  0.241552281 -0.032772444
##  [83,]  0.24153827 -0.777256383  0.151211957  0.023651360
##  [84,]  1.06109461 -0.633843245 -0.105311387 -0.183968453
##  [85,]  0.22397877 -0.287773512 -0.665249720 -0.254828368
##  [86,]  0.42913912  0.845582241 -0.450634071 -0.109675181
##  [87,]  1.04872805  0.522051797  0.395786384  0.037209019
##  [88,]  1.04453138 -1.382988719  0.688295960  0.136835600
##  [89,]  0.06958832 -0.219503335 -0.291579274 -0.147144581
##  [90,]  0.28347724 -1.329324639 -0.089410023  0.008905805
##  [91,]  0.27907778 -1.120028524 -0.094487601 -0.270657196
##  [92,]  0.62456979  0.024923029  0.020481147 -0.147686401
##  [93,]  0.33653037 -0.988404018  0.199389755  0.006530562
##  [94,] -0.36218338 -2.019237873 -0.105821048  0.019570812
##  [95,]  0.28858624 -0.855730320 -0.130889685 -0.107402349
##  [96,]  0.09136066 -0.181192126 -0.128978343 -0.229959626
##  [97,]  0.22771687 -0.384920081 -0.156213154 -0.132605877
##  [98,]  0.57638829 -0.154873597  0.271650362 -0.019860679
##  [99,] -0.44766702 -1.543792034 -0.190400930  0.199946457
## [100,]  0.25673059 -0.598851796 -0.091879161 -0.058622049
## [101,]  1.84456887  0.870421312 -1.005401018 -0.049249743
## [102,]  1.15788161 -0.698869862 -0.530160149 -0.040520754
## [103,]  2.20526679  0.562010477  0.202914170  0.059184194
## [104,]  1.44015066 -0.046987588 -0.163630107 -0.235770073
## [105,]  1.86781222  0.295044824 -0.395628375 -0.016298272
## [106,]  2.75187334  0.800409201  0.582309103 -0.101384486
## [107,]  0.36701769 -1.561502891 -0.986893267 -0.133123834
## [108,]  2.30243944  0.420065580  0.651706439 -0.238041242
## [109,]  2.00668647 -0.711438654  0.393990571 -0.086510630
## [110,]  2.25977735  1.921010376 -0.397551897  0.104838918
## [111,]  1.36417549  0.692756454 -0.284612074  0.107860420
## [112,]  1.60267867 -0.421700450 -0.023186408  0.058331633
## [113,]  1.88390070  0.419249651 -0.026338410  0.146414939
## [114,]  1.26011510 -1.162260421 -0.580249290  0.099157321
## [115,]  1.46764520 -0.442271587 -1.003869574  0.275658903
## [116,]  1.59007732  0.676244806 -0.638428708  0.191862996
## [117,]  1.47143146  0.255621824 -0.037431260 -0.155330271
## [118,]  2.42632899  2.556661251  0.127881459 -0.273807183
## [119,]  3.31069558  0.017780949  0.703305304  0.045188606
## [120,]  1.26376667 -1.706745380  0.267536893 -0.065180800
## [121,]  2.03771630  0.910467410 -0.234799484  0.167951254
## [122,]  0.97798073 -0.571764325 -0.828127201  0.027755587
## [123,]  2.89765149  0.413641060  0.857421825 -0.127336502
## [124,]  1.33323218 -0.481811219  0.005428364  0.139959148
## [125,]  1.70073390  1.013921867 -0.298450613 -0.061643734
## [126,]  1.95432671  1.007777596  0.419984722 -0.218338351
## [127,]  1.17510363 -0.316394472 -0.129937757  0.125420444
```

```
## [128,]  1.02095055  0.064346029 -0.337715967 -0.008654401
## [129,]  1.78834992 -0.187361215 -0.270658006  0.031087648
## [130,]  1.86364755  0.562290726  0.715634119 -0.208215164
## [131,]  2.43595373  0.259284433  0.727816146 -0.017923365
## [132,]  2.30492772  2.626323468  0.493473808 -0.211675709
## [133,]  1.86270322 -0.178549495 -0.354148712  0.100009882
## [134,]  1.11414774 -0.292922623  0.183488392 -0.186343697
## [135,]  1.20247330 -0.811315271  0.164723757 -0.489483470
## [136,]  2.79877045  0.856803329  0.542906499  0.295881050
## [137,]  1.57625591  1.068581107 -0.945853819  0.035605759
## [138,]  1.34629210  0.422430611 -0.180875478 -0.215421288
## [139,]  0.92482492  0.017223100 -0.416826193  0.005238409
## [140,]  1.85204505  0.676128174  0.012672115  0.195195239
## [141,]  2.01481043  0.613885637 -0.428332842  0.247538313
## [142,]  1.90178409  0.689575494 -0.130075005  0.469696647
## [143,]  1.15788161 -0.698869862 -0.530160149 -0.040520754
## [144,]  2.04055823  0.867520601 -0.338144000  0.045187126
## [145,]  1.99814710  1.049168747 -0.632413436  0.214045204
## [146,]  1.87050329  0.386966082 -0.256273852  0.389256845
## [147,]  1.56458048 -0.896686809  0.026371352  0.220192100
## [148,]  1.52117050  0.269069144 -0.180178380  0.119171137
## [149,]  1.37278779  1.011254419 -0.933395241  0.026128648
## [150,]  0.96065603 -0.024331668 -0.528248807 -0.163078032
```

```
(myEigen <- eigen(cor(data)))
```

```
## eigen() decomposition
## $values
## [1] 2.91849782 0.91403047 0.14675688 0.02071484
##
## $vectors
##            [,1]        [,2]       [,3]       [,4]
## [1,]  0.5210659 -0.37741762  0.7195664  0.2612863
## [2,] -0.2693474 -0.92329566 -0.2443818 -0.1235096
## [3,]  0.5804131 -0.02449161 -0.1421264 -0.8014492
## [4,]  0.5648565 -0.06694199 -0.6342727  0.5235971
```

```
myEigen$values[1]/sum(myEigen$values)
```

```
## [1] 0.7296245
```

Notice that the eigenvalue is the same as squaring the standard deviation!

```
(myPCA$loadings)
```

```
##
## Loadings:
##              Comp.1 Comp.2 Comp.3 Comp.4
## Sepal.Length  0.521  0.377  0.720  0.261
## Sepal.Width  -0.269  0.923 -0.244 -0.124
## Petal.Length  0.580         -0.142 -0.801
## Petal.Width   0.565         -0.634  0.524
```

```
##
##                 Comp.1 Comp.2 Comp.3 Comp.4
## SS loadings      1.00   1.00   1.00   1.00
## Proportion Var   0.25   0.25   0.25   0.25
## Cumulative Var   0.25   0.50   0.75   1.00
```

```
(myPCA$sdev)
```

```
##    Comp.1    Comp.2    Comp.3    Comp.4
## 1.7083611 0.9560494 0.3830886 0.1439265
```

```
(myPCA$sdev[1] ^ 2)
```

```
##   Comp.1
## 2.918498
```

```
myEigen$values / sum(myEigen$values)
```

```
## [1] 0.729624454 0.228507618 0.036689219 0.005178709
```

```
sum(myEigen$values[1:2]) / sum(myEigen$values)
```

```
## [1] 0.9581321
```

Eigenshoes

1. We take all the images of all available shoes, centered and clean.
2. We then recombine the images by weighting pixels and their associated colors.
3. The weighting is done so that the dominant (most important) characteristics (design features) that describe the shoes are placed in the first image.
4. The second image has the second-most dominant features and so on.
5. Each shoe image can be reconstructed from these images by reverse weighting.
6. A small subset of these images can reconstruct the design of the original shoes with high accuracy.

The predominant shoe features emerge with very few images...

Principal components are nothing more than Eigenvectors, at the end of the day. The Eigenvalues are nothing more than the variance!

```
var(myPCA$scores[,1])
```

```
## [1] 2.938085
```

```
myEigen
```

```
## eigen() decomposition
## $values
## [1] 2.91849782 0.91403047 0.14675688 0.02071484
```

```
##
## $vectors
##            [,1]        [,2]       [,3]       [,4]
## [1,]  0.5210659 -0.37741762  0.7195664  0.2612863
## [2,] -0.2693474 -0.92329566 -0.2443818 -0.1235096
## [3,]  0.5804131 -0.02449161 -0.1421264 -0.8014492
## [4,]  0.5648565 -0.06694199 -0.6342727  0.5235971
```

$$Var(Eigenvectors) = \lambda$$

```
# Prepare to merge

master <- read.csv("")
master$filename <- master$Profile_2500x1200_JPG
master1$filename <- master1$Profile_2500x1200_JPG

var <- mypca$sdev^2
sum(var[1:150]) / sum(var) # Eigenvalues divided by the variability
myvar <- mypca$scores[1:150,] # Captures most of the variability of the pictures!

mypixels <- data.frame(t(myvar))

mypixels$filename <- files

myfile <- read.csv("")
myfile$filename <- myfile$Profile_2500x1200_JPG


# Merge and save

total2 <- merge(mypixels,
                myfile,
                by = "filename",
                all.x = TRUE)

#rm(mupixels)
#rm(master)
#total$X <- NULL
#write.csv(total, ".csv")


# Analyze best and worst-selling products

setwd("~/focm/wk4")
m1 <- read.csv(".csv")
m2 = read.csv("")

total2 <- rbind(m1, m2)
myagg7 <- aggregate(x~Style.Number, data = total2, FUN = sum)
myagg7 <- myagg7[order(myagg7$x, decreasing = TRUE), ]
mymerge <- merge(myagg7, myfile,
                 by.x = "Style.Number",
                 by.y = "Style.Number", all.y = TRUE)
```

```
# TRANSFORM IMAGES

# GLOBALS
height <- 1200
width <- 2500
scale <- 20

newdata <- im
dim(newdata) <- c(length(files), height*width*3/scale^2)

mypca <- princomp(t(as.matrix(newdata)), scores = TRUE, cor = TRUE)

# Generate Eigenshoes

mypca2 <- t(mypca$scores)
dim(mypca2) <- c(length(files),
                 height / scale,
                 width / scale,
                 3)
par(mfrow = c(5,5))
par(mai = c(0.001, 0.001, 0.001, 0.001))
for (i in 1:25) { # Plot first 25 eigenshoes only
  plot_jpeg(writeJPEG(mypca2[i,,,],
                      bg = white))
  }
```