3.0-Principal-Component-Analysis

JD Santos

11/1/2020

Problem 1

For this problem we'll be using a dataset of police application fields:

```
police <- read.csv("https://bit.ly/police_applications")
```

a) Perform principal component analysis using the correlation matrix. You do not need to do data cleaning or fixing the direction of variables in this data.

To perform principal component anlysis, use the `princomp()` function

- Use `cor = T` to use the correlation matrix (correlation = True)
- Run `summary()` with `loading = T` To view the components and the variable eigenvectors

```
police.pca <- princomp(police, cor = T)
summary(police.pca, loading =T)
```

```
## Importance of components:
##                           Comp.1    Comp.2     Comp.3     Comp.4     Comp.5
## Standard deviation     2.2874533 1.5681077 1.14008912 1.11087426 1.07502934
## Proportion of Variance 0.3488295 0.1639308 0.08665355 0.08226944 0.07704587
## Cumulative Proportion  0.3488295 0.5127603 0.59941382 0.68168326 0.75872913
##                           Comp.6     Comp.7     Comp.8     Comp.9    Comp.10
## Standard deviation     0.9286329 0.83740056 0.75504829 0.62158268 0.60463975
## Proportion of Variance 0.0574906 0.04674931 0.03800653 0.02575767 0.02437261
## Cumulative Proportion  0.8162197 0.86296905 0.90097558 0.92673324 0.95110586
##                           Comp.11    Comp.12     Comp.13    Comp.14     Comp.15
## Standard deviation     0.57163001 0.43703530 0.365630031 0.20815295 0.196566621
## Proportion of Variance 0.02178406 0.01273332 0.008912355 0.00288851 0.002575896
## Cumulative Proportion  0.97288992 0.98562324 0.994535594 0.99742410 1.000000000
##
## Loadings:
##        Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9 Comp.10
## REACT           0.161  0.280          0.793                 0.238  0.153
## HEIGHT  0.309 -0.195  0.410          0.121 -0.161  0.166                 0.206
## WEIGHT  0.416                                              0.218 -0.174
## SHLDR   0.299 -0.208  0.238        -0.279          0.308  0.164  0.473 -0.351
## PELVIC  0.293 -0.201         0.435          0.103 -0.117 -0.223  0.185  0.457
```

```
## CHEST   0.361        -0.144  0.151 -0.151        -0.131  0.485 -0.368 -0.116
## THIGH   0.284  0.303 -0.249 -0.187                0.340 -0.353         0.238
## PULSE  -0.118  0.383  0.443        -0.134  0.239  0.314  0.279 -0.222
## DIAST          0.264         0.706         0.228        -0.361        -0.421
## CHNUP  -0.292 -0.235  0.223  0.205 -0.150  0.337         0.134 -0.268  0.470
## BREATH  0.253         0.464 -0.189        -0.205 -0.426 -0.358 -0.422 -0.175
## RECVR          0.454  0.281 -0.117 -0.426 -0.133        -0.155         0.148
## SPEED         -0.482        -0.208         0.272  0.502 -0.274 -0.318 -0.229
## ENDUR  -0.205                0.317        -0.759  0.391        -0.270  0.111
## FAT     0.368  0.218 -0.240                0.118  0.166        -0.243  0.182
##        Comp.11 Comp.12 Comp.13 Comp.14 Comp.15
## REACT   0.290   0.261
## HEIGHT         -0.743          -0.143
## WEIGHT                          0.832   0.182
## SHLDR           0.319  -0.294          -0.234
## PELVIC -0.272   0.326   0.373  -0.103   0.157
## CHEST   0.249                  -0.487   0.319
## THIGH           0.114  -0.511  -0.109   0.365
## PULSE  -0.572
## DIAST   0.134  -0.175
## CHNUP   0.315          -0.431          -0.139
## BREATH          0.280  -0.186
## RECVR   0.547           0.355
## SPEED   0.151           0.341           0.135
## ENDUR           0.173
## FAT                     0.158          -0.766
```

b) What percentage of the total variance is covered by the first two principal components?

If we refer to the “Importance of components” output above, we can find this answer in the “cumulative proportion” section. This reports that the cumulative proportion of the second component is 51.23%

c) Report the loading coefficients (eigenvector of the correlation matrix) of the first two principal components.

We can select the first two loadings from our police.pca by specifying loadings:

```
police.pca$loadings[,1:2]
```

```
##             Comp.1        Comp.2
## REACT   0.05074485  0.1610889135
## HEIGHT  0.30938720 -0.1948652676
## WEIGHT  0.41586867 -0.0326183906
## SHLDR   0.29939671 -0.2082727762
## PELVIC  0.29314755 -0.2013196634
## CHEST   0.36054275 -0.0005222451
## THIGH   0.28379796  0.3034561646
## PULSE  -0.11804080  0.3828513742
## DIAST  -0.03411639  0.2639467180
## CHNUP  -0.29172354 -0.2346251379
## BREATH  0.25261897 -0.0262139810
## RECVR  -0.02075408  0.4537337307
## SPEED  -0.03228068 -0.4821396135
## ENDUR  -0.20471081 -0.0352845349
## FAT     0.36777130  0.2178618386
```

d) Describe what information we can extract from the first two principal components? Explain. (You need to interpret the loading of the first two PCs)

We would expect a candidate scoring high in PC1 would be tall and large in multiple dimensions (chest, thighs, weight, etc). They would struggle with chin-ups, and have relatively low endurance. Candidates scoring high in PC2are smaller, but have higher recovery and pulse rate in general

Problem 2

These questions use per-state crime data:

```
crime <- read.csv("https://rb.gy/wu8kvo", row.names = "STATE")
head(crime)
```

```
##            MURDER RAPE ROBBERY ASSAULT BURGLARY LARCENY  AUTO
## ALABAMA      14.2 25.2    96.8   278.3   1135.5  1881.9 280.7
## ALASKA       10.8 51.6    96.8   284.0   1331.7  3369.8 753.3
## ARIZONA       9.5 34.2   138.2   312.3   2346.1  4467.4 439.5
## ARKANSAS      8.8 27.6    83.2   203.4    972.6  1862.1 183.4
## CALIFORNIA   11.5 49.4   287.0   358.0   2139.4  3499.8 663.5
## COLORADO      6.3 42.0   170.7   292.9   1935.2  3903.2 477.1
```

a) Perform the principal components using the correlation matrix. You do not need to do data cleaning or fixing the direction of variables in this data.

As before, we will save the principal component analysis into an object and print the results with `summary()`

```
crime.pca <- princomp(crime, cor = T)
summary(crime.pca, loading = T)
```

```
## Importance of components:
##                           Comp.1    Comp.2    Comp.3     Comp.4     Comp.5
## Standard deviation     2.0285363 1.1129788 0.8519487 0.56252293 0.50791186
## Proportion of Variance 0.5878514 0.1769603 0.1036881 0.04520458 0.03685349
## Cumulative Proportion  0.5878514 0.7648116 0.8684997 0.91370429 0.95055778
##                            Comp.6     Comp.7
## Standard deviation     0.47121064 0.35221592
## Proportion of Variance 0.03171992 0.01772229
## Cumulative Proportion  0.98227771 1.00000000
##
## Loadings:
##          Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
## MURDER    0.300  0.629  0.178  0.232  0.538  0.259  0.268
## RAPE      0.432  0.169 -0.244         0.188 -0.773 -0.296
## ROBBERY   0.397         0.496  0.558 -0.520 -0.114
## ASSAULT   0.397  0.344        -0.630 -0.507  0.172  0.192
## BURGLARY  0.440 -0.203 -0.210         0.101  0.536 -0.648
## LARCENY   0.357 -0.402 -0.539  0.235                0.602
## AUTO      0.295 -0.502  0.568 -0.419  0.370         0.147
```

b) What percentage of the total variance is covered by the first two principal components?

Per the output above, the cumulative proportion of the first two components is 76.5%.

c) Report the loading coefficients (eigenvector of the correlation matrix) of the first two principal components.

The first two loadings can be pulled from our `crime.pca` variable, here we will store it in `pc1_2` :

```
pc1_2 <- crime.pca$loadings[,1:2]
pc1_2
```

```
##             Comp.1      Comp.2
## MURDER   0.3002792  0.62917444
## RAPE     0.4317594  0.16943512
## ROBBERY  0.3968755 -0.04224698
## ASSAULT  0.3966517  0.34352815
## BURGLARY 0.4401572 -0.20334059
## LARCENY  0.3573595 -0.40231912
## AUTO     0.2951768 -0.50242093
```

d) Describe what information we can extract from the first two principal components? Explain. (You need to interpret the loading of the first two PCs)

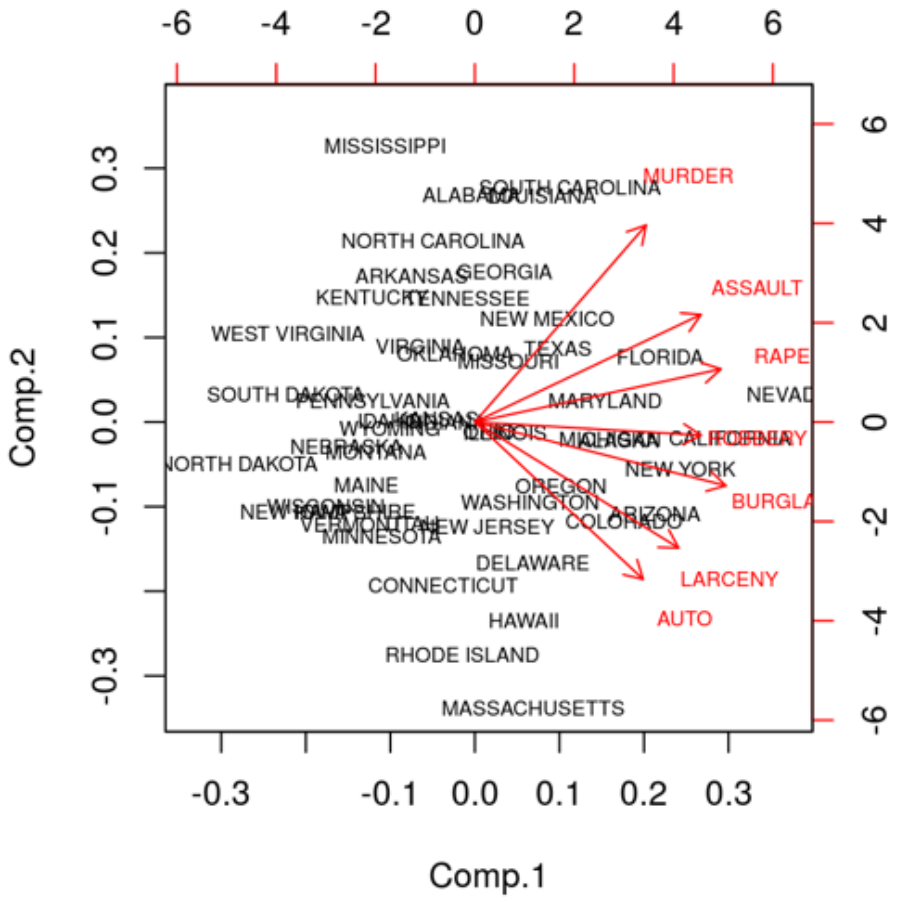
A state in PC1 scores highly in almost every type of crime, with Auto crime being the lowest. In PC2, there are strong scores for murders and somewhat for assault, but many of the other types of crime are actually negative associated like larceny and auto

e) Construct the biplot graph of the crime data. Interpret the resulting biplot graph for "MISSISSIPPI," "NEVADA," and "HAWAII." (You can validate your conclusions by looking at the actual standardized (scaled) data values.)

A biplot graph can be created with the aptly named `biplot()` function. From this graph we can see:

- Mississippi scores the highest of all states in PC2, we would it expect its crime indicators to align to the features described about PC2 above
    - This can be confirmed looking at the z-scores of the scaled crime data, murder is extremely high compared to the rest
    - We also see negative larceny and auto scored
- Nevada is the state with the highest PC1 score
    - Again we'd expect the features described above, with strong scores across all crime categories
- Hawaii scores very low on PC1, and negative in PC2
    - From this I'd assume crime in general is low in Hawaii
    - Checking the scaled data, we see that most crimes are low with the most prevalent being larceny

```
biplot(crime.pca, col=c("black", "red"), cex = 0.6)
```




```
options(digits = 2)
scale(crime)
```

```
##                MURDER    RAPE   ROBBERY ASSAULT BURGLARY LARCENY      AUTO
## ALABAMA         1.747 -0.050 -3.1e-01   0.668  -0.3617  -1.087 -0.50067
## ALASKA          0.868  2.404 -3.1e-01   0.725   0.0920   0.962  1.94304
## ARIZONA         0.532  0.787  1.6e-01   1.007   2.4377   2.474  0.32045
## ARKANSAS        0.351  0.173 -4.6e-01  -0.079  -0.7384  -1.115 -1.00378
## CALIFORNIA      1.049  2.200  1.8e+00   1.463   1.9597   1.141  1.47871
## COLORADO       -0.296  1.512  5.3e-01   0.814   1.4875   1.697  0.51488
## CONNECTICUT    -0.839 -0.830  6.1e-02  -0.793   0.1251  -0.070  1.11520
## DELAWARE       -0.373 -0.078  3.7e-01  -0.171   0.9034   1.387  0.46265
## FLORIDA         0.713  1.289  7.2e-01   2.372   1.3134   1.611 -0.13509
## GEORGIA         1.101  0.499  1.9e-01   0.451   0.1369  -0.690 -0.41173
## HAWAII         -0.063 -0.022  4.4e-02  -1.468   1.4327   1.721  0.57848
## IDAHO          -0.503 -0.589 -9.6e-01  -0.387  -0.5575  -0.099 -0.72353
## ILLINOIS        0.635 -0.366  9.9e-01  -0.023  -0.4784   0.217  0.78117
## INDIANA        -0.011  0.071 -1.0e-02  -0.577  -0.4757  -0.238 -0.00065
## IOWA           -1.330 -1.407 -9.4e-01  -1.212  -1.1086   0.019 -0.81505
```

```
## KANSAS         -0.218 -0.347 -2.6e-01  -0.307  -0.0497   0.094 -0.68888
## KENTUCKY        0.687 -0.617 -4.9e-01  -0.878  -0.9705  -1.390 -0.68319
## LOUISIANA       2.083  0.480  2.1e-01   1.239  -0.2923  -0.277 -0.20593
## MAINE          -1.304 -1.137 -9.7e-01  -0.412  -0.0897  -0.442 -0.67544
## MARYLAND        0.144  0.843  1.9e+00   1.472   0.2500   0.698  0.26358
## MASSACHUSETTS  -1.123 -0.459  5.1e-01   0.202   0.5557  -0.496  3.94310
## MICHIGAN        0.480  1.224  1.6e+00   0.631   0.5337   0.672  0.86856
## MINNESOTA      -1.227 -0.579 -4.3e-01  -1.252  -0.3635  -0.154 -0.17801
## MISSISSIPPI     1.773 -0.570 -6.6e-01  -0.221  -0.8702  -1.972 -1.20544
## MISSOURI        0.558  0.238  7.3e-01   0.221   0.0610  -0.340  0.00452
## MONTANA        -0.529 -0.840 -9.6e-01  -0.544  -1.1261   0.140 -0.35330
## NEBRASKA       -0.917 -0.710 -6.7e-01  -0.984  -1.2300  -0.489 -0.66406
## NEVADA          2.161  2.172  2.3e+00   1.433   2.6851   2.123  0.93940
## NEW HAMPSHIRE  -1.098 -1.397 -1.1e+00  -1.350  -0.5786  -0.451 -0.43500
## NEW JERSEY     -0.477 -0.440  6.4e-01  -0.261   0.3327   0.142  0.69275
## NEW MEXICO      0.351  1.242 -1.6e-01   1.318   0.2932   0.465 -0.61029
## NEW YORK        0.842  0.341  3.9e+00   1.075   1.0084   0.153  1.90426
## NORTH CAROLINA  0.816 -0.812 -7.1e-01   1.067  -0.3187  -0.873 -0.95880
## NORTH DAKOTA   -1.692 -1.555 -1.3e+00  -1.671  -1.9558  -1.141 -1.20389
## OHIO            0.092  0.146  7.5e-01  -0.301  -0.1755   0.035  0.11828
## OKLAHOMA        0.299  0.322 -5.7e-01  -0.063  -0.0086  -0.611 -0.26229
## OREGON         -0.658  1.317  9.1e-05   0.754   0.7966   1.150  0.05881
## PENNSYLVANIA   -0.477 -0.626  7.0e-02  -0.831  -0.9583  -1.443 -0.22920
## RHODE ISLAND   -0.994 -1.416 -4.3e-01  -0.103   0.4569   0.238  2.14005
## SOUTH CAROLINA  1.152  0.675 -2.1e-01   2.733   0.7439  -0.453 -0.68475
## SOUTH DAKOTA   -1.408 -1.137 -1.2e+00  -0.555  -1.6682  -1.332 -1.18941
## TENNESSEE       0.687  0.369  2.5e-01  -0.074  -0.0745  -1.233 -0.32848
## TEXAS           1.514  0.750  3.2e-01  -0.031   0.7196   0.437  0.10380
## UTAH           -1.020 -0.505 -6.3e-01  -0.638  -0.2782   0.459 -0.22248
## VERMONT        -1.563 -0.914 -1.1e+00  -1.098   0.1302  -0.648 -0.58081
## VIRGINIA        0.402 -0.226 -3.6e-01  -0.455  -0.7069  -0.207 -0.77989
## WASHINGTON     -0.813  1.289 -2.0e-01   0.135   0.7254   0.986 -0.08907
## WEST VIRGINIA  -0.373 -1.165 -9.3e-01  -1.201  -1.6060  -1.832 -1.10772
## WISCONSIN      -1.201 -1.193 -8.1e-01  -1.472  -1.0290  -0.079 -0.81091
## WYOMING        -0.529 -0.356 -9.6e-01  -0.373  -1.1106   0.139 -0.49394
## attr(,"scaled:center")
##   MURDER     RAPE  ROBBERY  ASSAULT BURGLARY  LARCENY     AUTO
##      7.4     25.7    124.1    211.3   1291.9   2671.3    377.5
## attr(,"scaled:scale")
##   MURDER     RAPE  ROBBERY  ASSAULT BURGLARY  LARCENY     AUTO
##      3.9     10.8     88.3    100.3    432.5    725.9    193.4
```

Problem 3

These questions use a husband/wife survey dataset involving a rating three questions:

1. What is the level of passionate love you feel for your partner?
2. What is the level of passionate love your partner feels for you?
3. What is the level of companionship love you feel for your partner?

Install CCA library

```
install.packages("CCA")
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.0'
## (as 'lib' is unspecified)
```

```
#hide
library(CCA)
```

Read in our survey CSV, save it as scaled data (z-scores), and create two sets for husband and wife responses:

```
love <- read.csv("https://bit.ly/3onLanp", header = T)
love <- scale(love)
options(digits = 3)
# Creating two sets for correlations
X <- love[, 1:4]  # Husband's responses
Y <- love[, 5:8]  # Wife's responses
```

a) Find the linear combination of the four husband responses and the linear combination of the four wife responses, maximizing the two derived variables' correlation. (Hint: use X coefficients and write U_1 as a linear combination of X variables, then use Y coefficients and write V_1 as a linear combination of Y variables)

Using the `cc()` function from CCA, we can get the canonical correlation between our husband and wife responses:

```
cca <- cc(X,Y)
```

Find the linear combination of the husband responses:

```
a <- cca$xcoef
round(a, 2)
```

```
##     [,1]  [,2]  [,3]  [,4]
## h1  0.19 -0.80 -1.05  0.08
## h2 -0.66  0.43  0.60  0.91
## h3 -1.83 -0.61  0.40 -2.02
## h4  1.81 -0.29  0.09  2.30
```

```
a1 <- a[,1]/min(a[,1])
round(a1, 3)
```

```
##     h1     h2     h3     h4
## -0.106  0.360  1.000 -0.985
```

$U_1 = -.11X_1 + .36X_2 + X_3 - 0.98X_4$

Find the linear combination of the wife's responses:

```
b <- cca$ycoef
round(b, 2)
```

```
##     [,1]  [,2]  [,3]  [,4]
## w1 -0.59  0.31 -0.74 -0.17
## w2 -0.46 -0.42 -0.08  0.87
## w3 -0.85  0.84  1.23  0.29
## w4  0.42 -1.21 -0.69 -1.07
```

```
b1 <- b[,1]/min(b[,1])
round(b1, 3)
```

```
##     w1     w2     w3     w4
##  0.694  0.542  1.000 -0.487
```

$V_1 = 0.69X_1 + .54X_2 + X_3 - 0.49X_4$

b) Find the correlation between U_1, V_1.

We can find the correlations from the CC function:

```
round(cca$cor, 2)
```

```
## [1] 0.57 0.42 0.23 0.09
```

c) What does the husband linear combination (U_1) measure? To answer, based on the coefficient of X in U_1, ask yourself, "What does the husband linear combination measure?"

The husband's U_1 measures the score for their reported "level of compassionate love."

d) Repeat C. for the wives linear combination (V_1).

The wife's V_1 measures the score for their reported "level of compassionate love."