SM2_Portfolio_1

Henry Bourne

2023-01-23

Principal Component analysis

Task 1

First let's look at the structure of the data:

```
USA <- USArrests[,-3]
summary(USA)
```

```
##      Murder          Assault           Rape
##  Min.   : 0.800   Min.   : 45.0   Min.   : 7.30
##  1st Qu.: 4.075   1st Qu.:109.0   1st Qu.:15.07
##  Median : 7.250   Median :159.0   Median :20.10
##  Mean   : 7.788   Mean   :170.8   Mean   :21.23
##  3rd Qu.:11.250   3rd Qu.:249.0   3rd Qu.:26.18
##  Max.   :17.400   Max.   :337.0   Max.   :46.00
```

We have four columns and would like to perform PCA on the data minus the UrbanPop feature. First we will carry out PCA using the covariance matrix, S, manually:

```
S.USA <- cov(USA)
ev.USA <- eigen(S.USA)
ev.USA
```

```
## eigen() decomposition
## $values
## [1] 6996.480738   48.658639    6.725962
##
## $vectors
##             [,1]        [,2]        [,3]
## [1,] -0.04180743  0.02555358  0.99879886
## [2,] -0.99630506 -0.07612980 -0.03975532
## [3,] -0.07502247  0.99677042 -0.02864195
```

Notice that the eigenvectors in the matrix are already sorted by the size of their eigenvalues and so have decreasing sample variance. Thus the matrix of our principal components is exactly the matrix of eigenvalues above.

Now we will carry out PCA using the correlation matrix, R, and the help of the *prcomp* command.

```
pc_USArrests <- prcomp(~Murder + Assault + Rape, USArrests, scale. = TRUE, retx=TRUE); pc_USArrests # W
```

```
## Standard deviations (1, .., p=3):
## [1] 1.5357670 0.6767949 0.4282154
##
## Rotation (n x k) = (3 x 3):
```

```
##                PC1        PC2        PC3
## Murder  -0.5826006  0.5339532 -0.6127565
## Assault -0.6079818  0.2140236  0.7645600
## Rape    -0.5393836 -0.8179779 -0.1999436
```

```
pc_USArrests$x
```

```
##                         PC1         PC2          PC3
## Alabama        -1.198027832  0.83381177 -0.162178476
## Alaska         -2.308747325 -1.52396221  0.038335742
## Arizona        -1.503330652 -0.49830384  0.878223112
## Arkansas       -0.175989446  0.32473260  0.071111741
## California     -2.045235843 -1.27257704  0.381539326
## Colorado       -1.263413283 -1.42640632 -0.083693139
## Connecticut     1.627064626  0.17860374  0.290256038
## Delaware        0.074812801  0.41561083  0.998446677
## Florida        -2.830731325  0.42331809  0.208151641
## Georgia        -1.842343065  0.88277323 -1.080609032
## Hawaii          1.302403647 -0.53528688 -0.772523404
## Idaho           1.469223571 -0.15225639  0.414302742
## Illinois       -1.079579292  0.27941102  0.291234322
## Indiana         0.513394603 -0.20015992 -0.442228830
## Iowa            2.156636865 -0.11239443 -0.054668600
## Kansas          0.832079409 -0.08014103 -0.191017094
## Kentucky        0.478831591  0.50650575 -0.730308649
## Louisiana      -1.644731071  1.04957018 -0.373768062
## Maine           2.174592271  0.25034567  0.281818786
## Maryland       -1.790861674  0.18886129  0.551385569
## Massachusetts   0.895952687 -0.04050899  0.382293578
## Michigan       -1.989964308 -0.46614937 -0.129836314
## Minnesota       1.765715718 -0.32440018 -0.055072348
## Mississippi    -1.517623726  1.60645578 -0.271636024
## Missouri       -0.616205380 -0.44134841 -0.252834582
## Montana         0.967991307  0.04418005 -0.211907462
## Nebraska        1.240695366 -0.19093752 -0.039096727
## Nevada         -2.609154480 -1.41350501 -0.404109160
## New Hampshire   2.266374551  0.03511068  0.006998662
## New Jersey      0.277745503  0.13462220 -0.001387439
## New Mexico     -1.942431655 -0.21292600  0.307911561
## New York       -1.330622591  0.19467099  0.193797219
## North Carolina -1.614416593  1.51406566  0.901426577
## North Dakota    2.654501432  0.03705123  0.126761976
## Ohio            0.425915739 -0.20485611 -0.400616435
## Oklahoma        0.374013508 -0.08879455  0.012150589
## Oregon          0.007484513 -1.08883723  0.126183190
## Pennsylvania    1.036129757  0.20425023 -0.249615405
## Rhode Island    1.308026751  0.59975203  0.923110514
## South Carolina -1.747107574  0.97782271  0.035740543
## South Dakota    1.637375231  0.02980135 -0.036558286
## Tennessee      -1.176095386  0.21275256 -0.724219698
## Texas          -1.123432710  0.30710594 -0.504726135
## Utah            0.887958106 -0.83848257  0.144173194
## Vermont         2.220758807 -0.12420650 -0.125927856
## Virginia        0.043078167  0.09584025 -0.224223254
## Washington      0.408525962 -0.96439783  0.190536108
```

```
## West Virginia   1.621260055  0.55554584 -0.275017445
## Wisconsin       2.153811966 -0.02739623 -0.127792014
## Wyoming         0.527690701  0.34566289  0.169682461
```

The **sdev** component tells us the standard deviations of the principal components which corresponds to the squareroot of the eigenvalues of the covariance matrix, $(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \sqrt{\lambda_3})^T = \text{diag}(\Lambda)$. The **rotation** component contains the matrix of the principal components, ie. the eigenvectors of the correlation matrix, R, $\mathbf{A} = [v_1 \quad v_2 \quad v_3]$ where $\mathbf{A}$ is the matrix of principal components (the eigenvectors of R) and v_i is the i-th eigenvector. Finally $\mathbf{x}$ contains the data transformed by the principal component matrix (ie. our now uncorrelated data), in the notes: $\mathbf{Y} = \mathbf{XA}$.

The correlation matrix can be interpreted as the sample covariance matrix of the scaled data. So wether we should use the covariance matrix or correlation matrix depends on the variances of the predictor variables. Recalling the variances:

```
diag(S.USA)
```

```
##     Murder    Assault       Rape
##   18.97047 6945.16571   87.72916
```

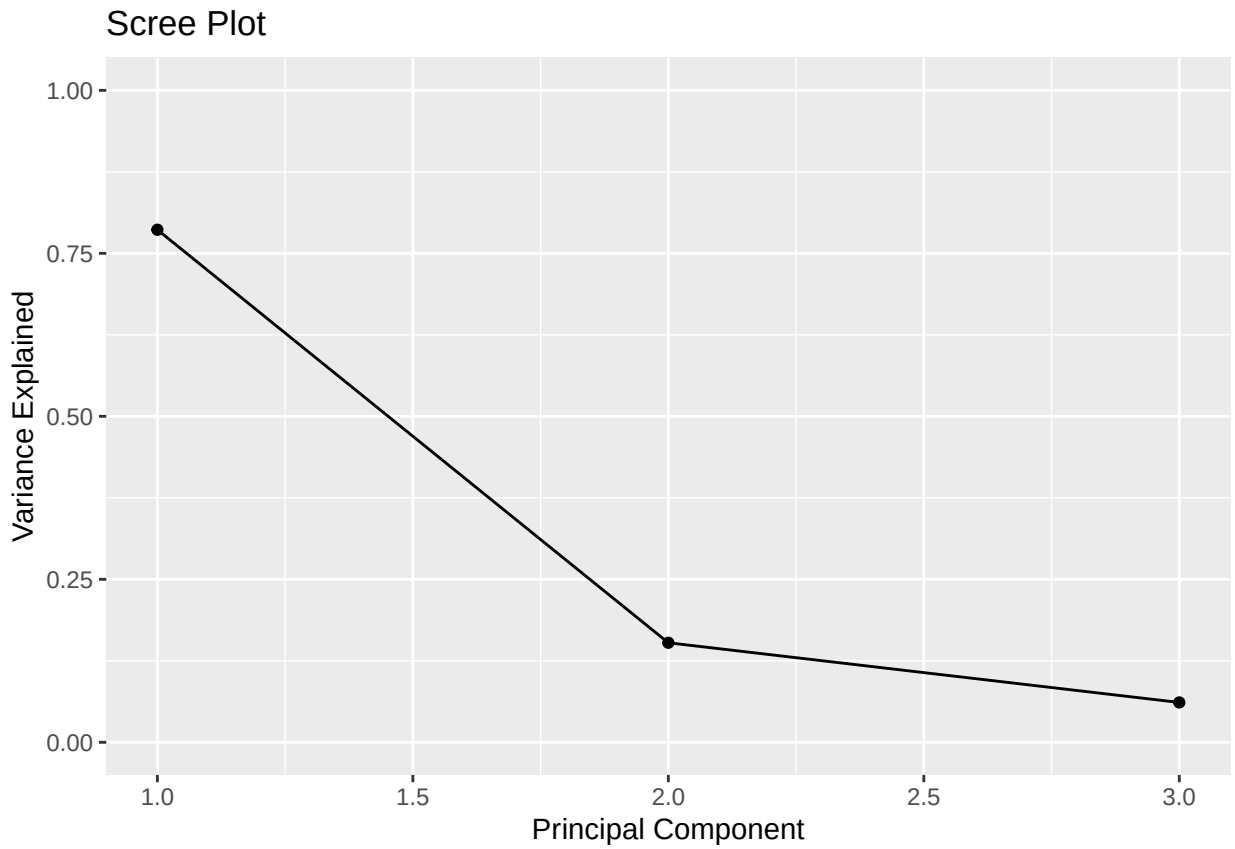
We see that the variances of the predictor variables greatly differ by orders of magnitude, therefore it is sensible to work with the scaled data, ie. the correlation matrix R.

Now we will plot a **scree plot** which is a plot of the variance explained by each principal component, ie. the percentage of the variance accounted for by the principal component.

```
var_prcnt = pc_USArrests$sdev^2 / sum(pc_USArrests$sdev^2)

qplot(c(1:3), var_prcnt) +
  geom_line() +
  xlab("Principal Component") +
  ylab("Variance Explained") +
  ggtitle("Scree Plot") +
  ylim(0, 1)
```

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
```



From the scree plot we can see the first principal component accounts for a very large percentage of the variance with the 3rd principal component accounting for less than 12% of the variance.

We will now compute the Kaiser's criterion:

```
max(which(pc_USArrests$sdev > sum(pc_USArrests$sdev)/length(pc_USArrests$sdev)))
```

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

and now the number of PC's to keep according to Horn's parallel analysis:

```
M <-1000
n <- nrow(USA)
p <- ncol(USA)
lambdas <- matrix(NA, M, p)
for(i in 1:M){
  M <- matrix( rnorm(p*n,mean=0,sd=1), n, p)  # genrate matrix with N(0,1) entries
  R <- cor(M) # find correlation matix
  S <- diag(pc_USArrests$sdev) %*% R %*% diag(pc_USArrests$sdev) # scale
  lambdas[i,] <- eigen(S)$values # find the eigenvalues
}
mean_lambda <- colMeans(lambdas) # find the mean of the eigenvalues

# find the largest index of eigenvalues larger than the mean eigenvalue of our M standard normal matric
max(which(pc_USArrests$sdev > mean_lambda))
```

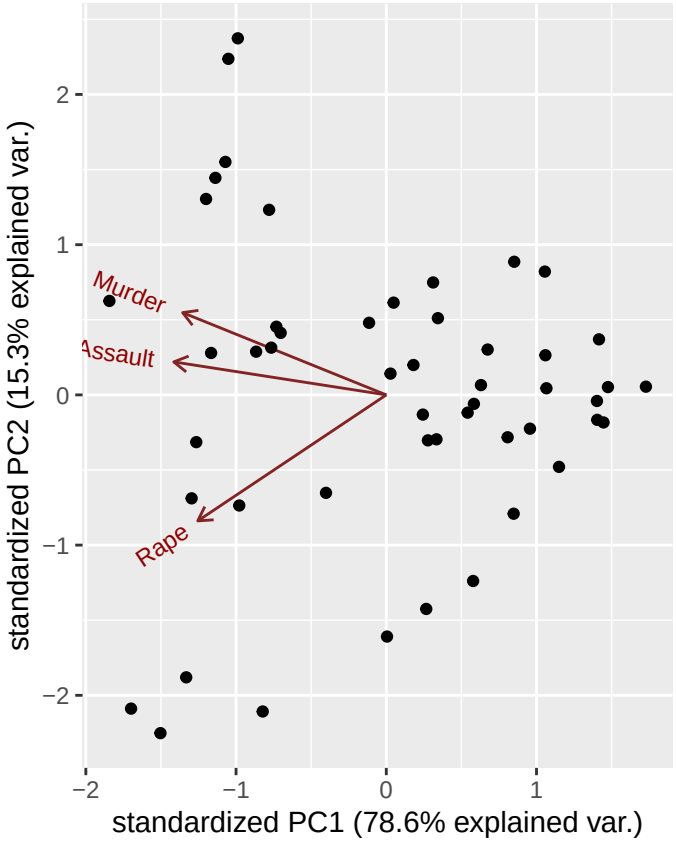
```
## [1] 3
```

Kaiser's criterion tells us to only keep the first principal component, ie. let $q = 1$, whereas Horn's parallel analysis suggests not throwing away any of the principal components, ie. $q = 3$. Horn's parallel analysis takes

into account the sampling error that arises from the fact that we don't have infinite observations (only n) unlike Kaiser's criterion, therefore we will select $q = 3$. Looking at the scree plot we can back up this decision as throwing away the second two principal components would amount to loosing close to 25% of the varaince.

Now we will produce a biplot:

```
ggbiplot(pc_USArrests)
```



The black dots are the datapoints transformed by our PC's, here we plot the first component versus the second. The red arrows tell us which predictor variables contribute to which principal components. Here we see that higher values in murder and assault contribute to a larger value of PC2 for example. From the plot we see that all the features have negative values for the first component which means the first component is an average of all three features and suggests that all three features are correlated with each other. For the second component we see that larger values in murder and assault lead to a higher value and rape to a lower value, ie. it contrasts rape against the other features.

Task 2

We will be working with the iris dataset:

```
summary(iris)
```

```
##    Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
```

```
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

Ideally we would like to use the covariance matrix as this will preserve variance, however, if the predictor variables are scaled differently this could lead to one predictor variable accounting for a large percentage of the variance. In this case we would prefer to standardize the data and use the correlation matrix. Let us asses the variances of the predictor variables:

```
diag(cov(iris[,-5]))
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##    0.6856935    0.1899794    3.1162779    0.5810063
```

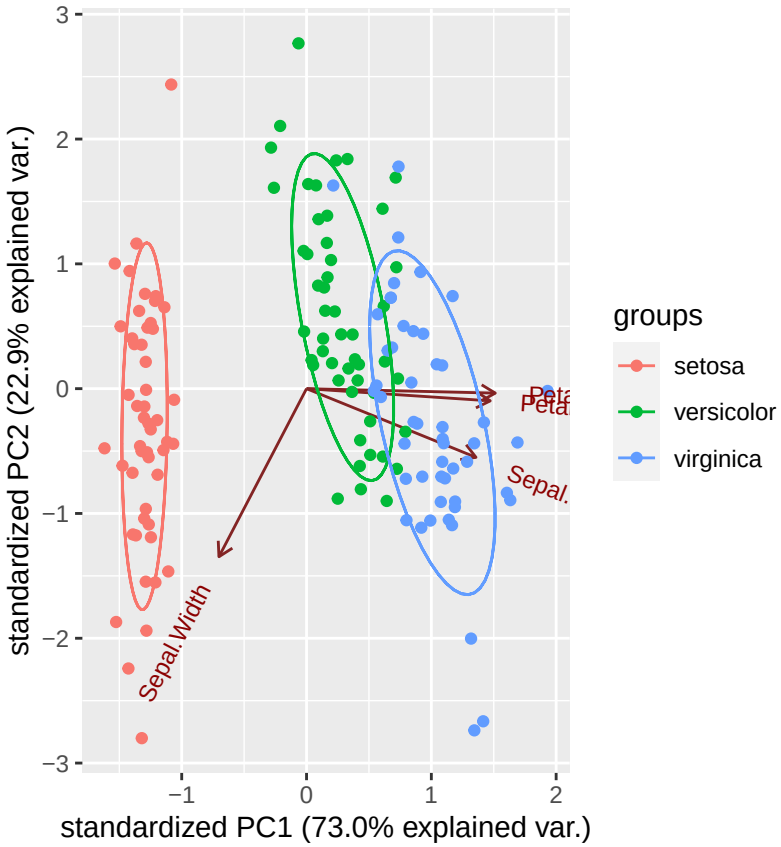
We see that the variances do differ and even though they all use the same measurement on the same scale the variances are very different, for example Petal.length has a variance of 13 times Sepal.Width so we will use the correlation matrix:

```
pc_iris <- prcomp(~Sepal.Length + Sepal.Width + Petal.Length + Petal.Width , iris, scale. = TRUE, retx='
```

```
## Standard deviations (1, .., p=4):
## [1] 1.7083611 0.9560494 0.3830886 0.1439265
##
## Rotation (n x k) = (4 x 4):
##                     PC1         PC2        PC3        PC4
## Sepal.Length  0.5210659 -0.37741762  0.7195664  0.2612863
## Sepal.Width  -0.2693474 -0.92329566 -0.2443818 -0.1235096
## Petal.Length  0.5804131 -0.02449161 -0.1421264 -0.8014492
## Petal.Width   0.5648565 -0.06694199 -0.6342727  0.5235971
```

Let's now plot the two dimensional reduction of observations and color the data-points according to their species:

```
ggbiplot(pc_iris, ellipse=TRUE, groups = iris$Species)
```



From the plot we see that using our first two principal components has resulted in a two-dimensional reduction of the data-set where the points are clustered according to group, ie. all the points belonging to the setosa species appear together in the feature space and the same can be said for versicolor and virginica, although the between group scatterness between these two is much lower (ie. they appear much sloer together in feature space).

Task 3