Principal Component Analysis

Dean D'souza

2016-07-19

## Solutions

* **Apply both methods of PCA (princomp and preProcess) to the USArrests data supplied in Moodle**

#Setting up the data  
dat<-read.csv("~/PrinAn2/USArrests.csv", row.names=1) #importing into dat for convenience  
# head(dat) #Commented out as not necessary

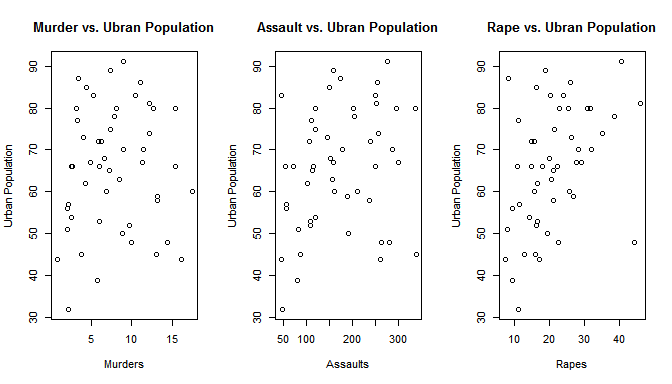
1. **First, run a correlation matrix on the 4 numeric columns, and explain which you think should be removed if you were trying to predict UrbanPop from Murder, Assault, and Rape**

cor(dat)

## Murder Assault UrbanPop Rape  
## Murder 1.00000000 0.8018733 0.06957262 0.5635788  
## Assault 0.80187331 1.0000000 0.25887170 0.6652412  
## UrbanPop 0.06957262 0.2588717 1.00000000 0.4113412  
## Rape 0.56357883 0.6652412 0.41134124 1.0000000

As we can see from the above correlation matrix, 'UrbanPop' is closely related to 'Rape' covariate. However we can also see that 'Murder' and 'Assault' covariates, while less correlated to UrbanPop, are closely correlated to each other. As 'Murder' seems to be less correlated to 'UrbanPop', in my opinion, we can remove 'Murder', keeping 'Assault' and 'Rape' in our model. (We might also choose to exclude 'Assault' instead as it has more correlation to 'Rape' as compared to 'Murder', which could possibly further reduce correlation among covariates)

1. **Next, create scatter plots of Murder, Assault, and Rape versus UrbanPop**



1. **Run PCA once on the columns Murder, Assault, and Rape using the first method described in this lecture**

* We perfrom PCA using the princomp() function as follows:

USArrests\_pca1<-princomp(~Murder+Assault+Rape+UrbanPop,data=dat)  
attributes(USArrests\_pca1)

## $names  
## [1] "sdev" "loadings" "center" "scale" "n.obs" "scores"   
## [7] "call"   
##   
## $class  
## [1] "princomp"

USArrests\_pca1$scores

## Comp.1 Comp.2 Comp.3 Comp.4  
## Alabama -64.802164 -11.4480074 -2.49493284 2.4079009  
## Alaska -92.827450 -17.9829427 20.12657487 -4.0940470  
## Arizona -124.068216 8.8304030 -1.68744836 -4.3536852  
## Arkansas -18.340035 -16.7039114 0.21018936 -0.5209936  
## California -107.422953 22.5200698 6.74587299 -2.8118259  
## Colorado -34.975986 13.7195840 12.27936280 -1.7214637  
## Connecticut 60.887282 12.9325302 -8.42065719 -0.6999023  
## Delaware -66.731025 1.3537978 -11.28095735 -3.7279812  
## Florida -165.244370 6.2746901 -2.99793315 1.2476807  
## Georgia -40.535177 -7.2902396 3.60952946 7.3436728  
## Hawaii 123.536106 24.2912079 3.72444284 3.4728494  
## Idaho 51.797002 -9.4691910 -1.52006356 -3.3478283  
## Illinois -78.992097 12.8970605 -5.88326477 0.3676407  
## Indiana 57.550961 2.8462647 3.73816049 1.6494302  
## Iowa 115.586790 -3.3421305 -0.65402935 -0.8694960  
## Kansas 55.789694 3.1572339 0.38436416 0.6527917  
## Kentucky 62.383181 -10.6732715 2.23708903 3.8762164  
## Louisiana -78.277631 -4.2949175 -3.82786965 4.4835590  
## Maine 89.261044 -11.4878272 -4.69240562 -2.1161995  
## Maryland -129.330136 -5.0070315 -2.34717282 -1.9283242  
## Massachusetts 21.266283 19.4501790 -7.50714835 -1.0348189  
## Michigan -85.451527 5.9045567 6.46434210 0.4990479  
## Minnesota 98.954816 5.2096006 0.00657376 -0.7318957  
## Mississippi -86.856358 -27.4284196 -5.00343624 3.8797577  
## Missouri -7.986289 5.2756414 5.50057972 0.6794055  
## Montana 62.483635 -9.5105021 1.83835536 0.2459426  
## Nebraska 69.096544 -0.2111959 0.46802086 -0.6565664  
## Nevada -83.613578 15.1021839 15.88869482 0.3341962  
## New Hampshire 114.777355 -4.7345584 -2.28238693 -0.9359106  
## New Jersey 10.815725 23.1373389 -6.31015739 1.6124273  
## New Mexico -114.868163 -0.3364531 2.26126996 -1.3812478  
## New York -84.294231 15.9239655 -4.72125960 0.8920194  
## North Carolina -164.325514 -31.0966153 -11.69616350 -2.1111927  
## North Dakota 127.495597 -16.1350394 -1.31182982 -2.3009639  
## Ohio 50.086822 12.2793244 1.65733077 2.0291157  
## Oklahoma 19.693723 3.3701310 -0.45314329 -0.1803457  
## Oregon 11.150240 3.8660682 8.12998050 -2.9140109  
## Pennsylvania 64.689142 8.9115466 -3.20646858 1.8749353  
## Rhode Island -3.063973 18.3739704 -17.47001970 -2.3082597  
## South Carolina -107.281069 -23.5361159 -2.03279501 1.2517463  
## South Dakota 86.106720 -16.5978586 1.31437998 -1.2522874  
## Tennessee -17.506264 -6.5065756 6.10012753 3.9228558  
## Texas -31.291122 12.9849566 -0.39340922 4.2420040  
## Utah 49.913397 17.6484577 1.78816852 -1.8677052  
## Vermont 124.714469 -27.3135591 4.80277765 -2.0049857  
## Virginia 14.817448 -1.7526150 1.04538813 1.1738408  
## Washington 25.075839 9.9679669 4.78112764 -2.6910819  
## West Virginia 91.544647 -22.9528778 -0.40198344 0.7368781  
## Wisconsin 118.176328 5.5075792 -2.71132077 0.2049724  
## Wyoming 10.434539 -5.9244529 -3.79444682 -0.5178674

USArrests\_pca1

## Call:  
## princomp(formula = ~Murder + Assault + Rape + UrbanPop, data = dat)  
##   
## Standard deviations:  
## Comp.1 Comp.2 Comp.3 Comp.4   
## 82.890847 14.069560 6.424204 2.457837   
##   
## 4 variables and 50 observations.

summary(USArrests\_pca1)

## Importance of components:  
## Comp.1 Comp.2 Comp.3 Comp.4  
## Standard deviation 82.8908472 14.06956001 6.424204055 2.4578367034  
## Proportion of Variance 0.9655342 0.02781734 0.005799535 0.0008489079  
## Cumulative Proportion 0.9655342 0.99335156 0.999151092 1.0000000000

From this summary data we can conclude that component 1 is sufficient to explain more than 95% of the data.

1. **Run PCA again on the columns Murder, Assault, and Rape using the second method from the caret package**

* We perform PCA using preProcess from the 'caret' package as follows ('caret' and other required packages were loaded earlier in the 'knitr' setup portion):

USArrests\_pca2<-preProcess(dat,method = c("BoxCox","center","scale","pca"),thresh = 0.95)  
attributes(USArrests\_pca2)

## $names  
## [1] "dim" "bc" "yj" "et" "mean"   
## [6] "std" "ranges" "rotation" "method" "thresh"   
## [11] "pcaComp" "numComp" "ica" "wildcards" "k"   
## [16] "knnSummary" "bagImp" "median" "data"   
##   
## $class  
## [1] "preProcess"

summary(USArrests\_pca2)

## Length Class Mode   
## dim 2 -none- numeric   
## bc 4 -none- list   
## yj 0 -none- NULL   
## et 0 -none- NULL   
## mean 4 -none- numeric   
## std 4 -none- numeric   
## ranges 0 -none- NULL   
## rotation 12 -none- numeric   
## method 5 -none- list   
## thresh 1 -none- numeric   
## pcaComp 0 -none- NULL   
## numComp 1 -none- numeric   
## ica 0 -none- NULL   
## wildcards 2 -none- list   
## k 1 -none- numeric   
## knnSummary 1 -none- function  
## bagImp 0 -none- NULL   
## median 0 -none- NULL   
## data 0 -none- NULL

1. **Be sure to create the scores from the PCA object when using the caret package method**

* We create the scores as follows:

USArrests\_num\_pca<-predict(USArrests\_pca2,dat)  
head(USArrests\_num\_pca)

## PC1 PC2 PC3  
## Alabama -1.03453112 1.0848115 -0.33595844  
## Alaska -1.67318523 1.3179300 1.37135111  
## Arizona -1.75849483 -0.6756439 -0.03729575  
## Arkansas -0.06792624 1.1731575 0.05737966  
## California -2.38210391 -1.4475095 0.30790267  
## Colorado -1.50855460 -0.7761294 0.83838786

1. **Explain why BoxCox, center, and scale functions were used in the second method**

In the second method for PCA we passed the 'BoxCox' parameter/fucntion to the preProcess() function as this parameter specifies to the function to transform the data to be more normally distributed prior to PCA. We passed the 'center' parameter/function to specify that the mean should be subtracted from each value prior to PCA. The 'scale' parameter/function was passed to specify that each value should be divided by the standard deviation prior to PCA. These methods are called so that we can pre process the data to be more suitable for PCA.