APPLIED STATISTICS

Principal Components Analysis (PCA)

Dr Tao Zou

Research School of Finance, Actuarial Studies & Statistics
The Australian National University

Last Updated: Wed Oct 25 09:24:51 2017

Overview

- Motivating Example
- Linear Combination and PCA
- PCA Usage
- When is it appropriate to use PCA?

References

1. **F.L. Ramsey and D.W. Schafer** (2012)
   Chapter 17 of *The Statistical Sleuth*
2. **J. Zhang** (2009)
   Chapter 4 of *Data Mining and Its Application*
3. The slides are made by **R Markdown**.
   http://rmarkdown.rstudio.com

Example: Test Score Data

To illustrate PCA we will be using a dataset "testscores.txt". This dataset contains the results of qualifying examinations for 25 graduate students in mathematics at a U.S university.

```
rm(list=ls())
setwd('~/Desktop/Research/AppliedStat2017/L15')
testscores=read.table("testscores.txt",header=T)
head(testscores)
```

```
##   diffgeom complex algebra reals statistics
## 1       36      58      43    36         37
## 2       62      54      50    46         52
## 3       31      42      41    40         29
## 4       76      78      69    66         81
## 5       46      56      52    56         40
## 6       12      42      38    38         28
```

The students sat for examinations in differential geometry, complex analysis, algebra, real analysis, and statistics.

The differential geometry and complex analysis examinations where closed book, while the remaining exams were open book.

Multivariate Data

We can call the observations of the tuple (diffgeom, complex, algebra, reals, statistics) multivariate data.

There are a lot of statistical tools to deal with the multivariate data. For instance, PCA, factor models, and cluster analysis. This course only instroduces PCA.

Combine Scores

It might be possible to reduce these five original vectors of test scores into one or two vectors that account for most of the information in the original dataset. This would be even more desirable if we had data on a larger number of different examinations.

One example is to use the mean test score, which is a linear combination of the five original scores with equal weights.

Do we have other methods to combine the five scores to produce an overall score?

PCA provides an answer to seek the linear combination of the original variables which contains the maximal variance (variation).

Principal Components Analysis (PCA)

PCA is potentially a way to select several linear combinations of the multivariate data that capture most of the variation information of the data.

This is most useful if relatively few linear combinations can explain most of the variation, and if the linear combinations can lend themselves to some useful interpretation.

5 variables → 1 average score

1 000   2

100 000 000   3

Linear Combinations of Variables and Principal Component Variables

S scores

A linear combination Z of variables $X_1, X_2, \cdots, X_k$ is given by:

$$Z = C_0 + C_1 X_1 + \cdots + C_k X_k.$$

In PCA, the original set of variables $X_1, \cdots, X_k$ is re-expressed in terms of a set of an equal number of principal component variables $Z_1, \cdots, Z_k$, where

linear combinations should satisfy

$$\begin{aligned} Z_1 &= C_{10} + C_{11} X_1 + \cdots + C_{1k} X_k; \\ Z_2 &= C_{20} + C_{21} X_1 + \cdots + C_{2k} X_k; \\ &\cdots \\ Z_k &= C_{k0} + C_{k1} X_1 + \cdots + C_{kk} X_k. \end{aligned}$$

Z_1 Z_2

We need **Requirement 1**: the principal component variables Z_{j_1} and Z_{j_2} are not correlated for $j_1 \neq j_2$. However, X_{j_1} and X_{j_2} are correlated for $j_1 \neq j_2$.

X_1, X_2

Linear Combinations of Variables and Principal Component Variables (Con'd)

We also need **Requirement 2**: the first principal component Z_1 is the linear combination of X_1, X_2, $\cdots$, X_k that exhibits the maximum variance by choosing $C_{10} \cdots C_{1k}$.

select the values of $C_{10} \cdots C_{1k}$
such that we maximise $Var(Z_1)$

By doing that, we are accounting for as much of the variation information contained in X_1, X_2, $\cdots$, X_k as possible, such that Z_1 has the most of the variation information.

select the values of $C_{20} \cdots C_{2k}$
such that we maximise $Var(Z_2)$

constraint

Requirement 3: the second principal component Z_2 is the linear combination of X_1, X_2, $\cdots$, X_k that has the maximum variance subject to the constraint that the correlation between Z_2 and Z_1 is zero.

Requirement 4: the third principal component Z_3 is the linear combination of X_1, X_2, $\cdots$, X_k that has the maximum variance $\boxed{\text{subject to}}$ the constraint that the correlation between Z_3 and Z_1 and the correlation between Z_3 and Z_2 are both zeros.

...

Linear Combinations of Variables and Principal Component Variables (Con'd)

We can keep working on the above procedures until we compute all the

$$\begin{pmatrix} C_{10} & C_{11} & \cdots & C_{1k} \\ \vdots & \vdots & \vdots & \vdots \\ C_{k0} & C_{k1} & \cdots & C_{kk} \end{pmatrix}$$

such that the above requirements are satisfied. The above figures are called loadings of principal components (PCs).

Example: Test Score Data (Con'd)

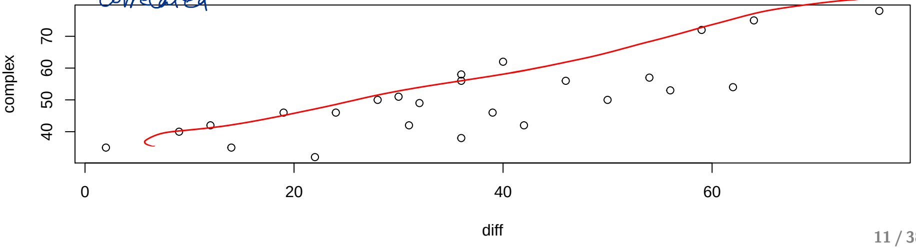
In this example we will find the principal components of the first two columns of the test score data.

```
head(testscores[,1:2])
```

```
##   diffgeom complex
## 1       36      58
## 2       62      54
## 3       31      42
## 4       76      78
## 5       46      56
## 6       12      42
```

```
X1=testscores[,1]; X2=testscores[,2];k=2
plot(X1,X2,xlab="diff",ylab="complex")
```

Correlated



Example: Test Score Data (Con'd)

To perform the PCA analysis the following R commands are used:

"lm" "glm" $k=2$

```
testscores.pca = princomp(testscores[, 1:k])
summary(testscores.pca, loadings = T)
```

True

sd of z_1 sd of z_2

```
## Importance of components:
##                           Comp.1    Comp.2
## Standard deviation     20.9086650 6.12944765
## Proportion of Variance  0.9208621 0.07913793
## Cumulative Proportion   0.9208621 1.00000000
##
## Loadings:
##          Comp.1 Comp.2
## diffgeom -0.862  0.506
## complex  -0.506 -0.862
```

(z_1) z_1, z_2

Example: Test Score Data (Con'd)

This output gives us the two principal components.

Since we have two variables, we can only have two principal components.

The output also gives us the percentage of the total variation that is explained by each of the principal components.

92% of the variation in the differential geometry and complex analysis scores is accounted for by the first principal component.

The second principal component accounts for the remaining 8% of the variation.

Example: Test Score Data (Con'd)

Suppose X_1 = diffgeom, X_2 =complex.

Then the first and second principal components are obtained by:

$0.862\,\bar{X}_1 + 0.506\,\bar{X}_2 \to C_{10}$ (idea)

$$Z_1 = -0.862(X_1 - \bar{X}_1) - 0.506(X_2 - \bar{X}_2);$$
$$Z_2 = 0.506(X_1 - \bar{X}_1) - 0.862(X_2 - \bar{X}_2),$$

Compare them to page 12

where $\bar{X}_1$ and $\bar{X}_2$ are the sample mean of X_1 and X_2. The first and second principal components can also be obtained directly from R

```
Z1=testscores.pca$scores[,1]
Z2=testscores.pca$scores[,2]
cor(Z1, Z2)
```

```
## [1] -5.866225e-16
```

$\bar{Z}_1 = 0$

$\bar{Z}_2 = 0$

Requirement for PCA

```
cor(X1, X2)
```

```
## [1] 0.8058591
```

Sample Cov of Z_1 and Z_2

$\frac{1}{n-1}\sum_{i=1}^{n}(Z_{1i} - \bar{Z}_1)(Z_{2i} - \bar{Z}_2)$

```
var(Z1)/(var(X1) + var(X2))
```

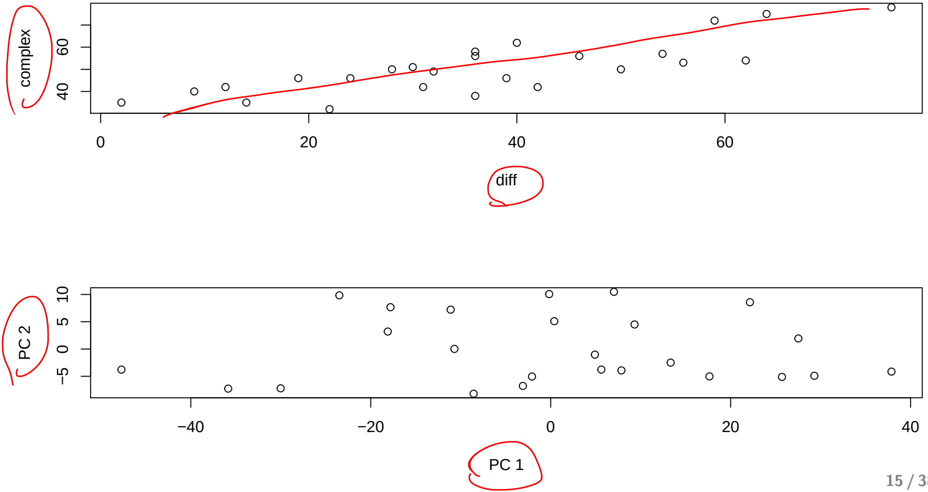
```
## [1] 0.9208621
```

L35

Example: Test Score Data (Con'd)

We can see that the two principal components are uncorrelated, while the original variables have a correlation of 0.81.

```
par(mfrow=c(2,1))
plot(testscores[,1],testscores[,2],xlab="diff",ylab="complex")
plot(Z1,Z2,xlab="PC 1",ylab="PC 2")
```



Example: Test Score Data (Con'd)

Recall that

$$\begin{cases} Z_1 = C_{10} + C_{11}X_1 + C_{12}X_2 \text{ and} \\ Z_2 = C_{20} + C_{21}X_1 + C_{22}X_2, \end{cases} \leftarrow \quad \text{page 14}$$

by the definition of the PCs.

Based on R, we can obtain the values of $C_{j_1 j_2}$, where

$$\begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} =$$

```r
t(testscores.pca$loadings[,1:k])
```

```
##          diffgeom    complex
## Comp.1 -0.8624793 -0.5060923
## Comp.2  0.5060923 -0.8624793
```

$$= \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$

Example: Test Score Data (Con'd)

We also have

$$\begin{pmatrix} \bar{X}_1 \\ \bar{X}_2 \end{pmatrix} =$$

```
testscores.pca$center
```

```
## diffgeom  complex
##    36.76    50.60
```

Hence,

$$\begin{pmatrix} C_{10} \\ C_{20} \end{pmatrix} =$$

```
-t(testscores.pca$loadings[,1:k])%*%testscores.pca$center
```

```
##               [,1]
## Comp.1 57.31301
## Comp.2 25.03750
```

Example: Test Score Data (Con'd)

Hence

$$\begin{pmatrix} C_{10} & C_{11} & C_{12} \\ C_{20} & C_{21} & C_{22} \end{pmatrix} =$$

```
cbind(-t(testscores.pca$loadings[,1:k])
      %*%testscores.pca$center,
      t(testscores.pca$loadings[,1:k]))
```

```
##                   diffgeom    complex
## Comp.1 57.31301 -0.8624793 -0.5060923
## Comp.2 25.03750  0.5060923 -0.8624793
```

By using the above loadings, we can compute the PCs:

use the information of loadings

```
Z=t(cbind(-t(testscores.pca$loadings[,1:k])
      %*%testscores.pca$center,
      t(testscores.pca$loadings[,1:k]))%*%t(cbind(1,X1,X2)))
```

Example: Test Score Data (Con'd)

Compare

computed by formula

```
head(Z[,1])
```

```
## [1]  -3.089599 -23.489692   9.320275 -47.710618 -10.702207  25.707382
```

```
head(Z[,2])
```

new z_1, z_2

```
## [1] -6.76697707  9.84134048  4.50223032 -3.77287054  0.01890475 -5.11352375
```

new values of X_1 X_2

```
head(testscores.pca$scores[,1])
```

old values of X_1 X_2

z_1

```
##          1          2          3          4          5          6
##  -3.089599 -23.489692   9.320275 -47.710618 -10.702207  25.707382
```

```
head(testscores.pca$scores[,2])
```

z_2

```
##           1           2           3           4           5           6
## -6.76697707  9.84134048  4.50223032 -3.77287054  0.01890475 -5.11352375
```

Example: Test Score Data (Con'd)

We will now use PCA on the full test score data.

```
testscores.pca=princomp(testscores)
summary(testscores.pca,loadings=T)
```

```
## Importance of components:
##                           Comp.1     Comp.2     Comp.3     Comp.4
## Standard deviation     28.4896795 9.03547104 6.60095491 6.13358179
## Proportion of Variance  0.8212222 0.08260135 0.04408584 0.03806395
## Cumulative Proportion   0.8212222 0.90382353 0.94790936 0.98597332
##                            Comp.5
## Standard deviation     3.72335754
## Proportion of Variance 0.01402668
## Cumulative Proportion  1.00000000
##
## Loadings:
##            Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
## diffgeom   -0.598  0.675  0.185  0.386
## complex    -0.361  0.245 -0.249 -0.829 -0.247
## algebra    -0.302 -0.214 -0.211 -0.135  0.894
## reals      -0.389 -0.338 -0.700  0.375 -0.321
## statistics -0.519 -0.570  0.607        -0.179
```

Example: Test Score Data (Con'd)

The first principal component explains 82% of the variance, and the first two principal components contain 90% of the variance.

In this example we might consider retaining only the first two principal components. This would mean we have only two variables instead of the original five.

These first two principal components give the "best" two dimensional view of the data.

Example: Test Score Data (Con'd)

Looking at the loadings ($k = 5$ in this case)

$$\begin{pmatrix} C_{10} & C_{11} & \cdots & C_{1k} \\ \vdots & \vdots & \vdots & \vdots \\ C_{k0} & C_{k1} & \cdots & C_{kk} \end{pmatrix} =$$

```r
k=5
cbind(-t(testscores.pca$loadings[,1:k])
      %*%testscores.pca$center,
      t(testscores.pca$loadings[,1:k]))
```

$z_1 = 96 - 0.6 X_1 - 0.4 X_2 - \cdots - 0.5 X_5$

X_1 X_2 X_2 X_4 X_5

```
##                  diffgeom    complex    algebra      reals  statistics
## Comp.1  96.20039 -0.59827824 -0.3607532 -0.3021774 -0.3890403 -0.51889947
## Comp.2  14.18709  0.67454038  0.2450733 -0.2140882 -0.3384022 -0.56972322
## Comp.3  22.12546  0.18525556 -0.2490064 -0.2114109 -0.6999921  0.60744765
## Comp.4  20.44037  0.38597894 -0.8287185 -0.1348456  0.3753787 -0.07178665
## Comp.5 -12.45426  0.06131111 -0.2470174  0.8944144 -0.3212995 -0.17892129
```

A reasonable interpretation for the first principal component is the average score of the five examinations. The second principal component contrasts the two closed book exams with the three open book exams.

Example: Test Score Data (Con'd)

Similarly by using the above loadings, we can compute the PCs

```
Z=t(cbind(-t(testscores.pca$loadings[,1:k])
      %*%testscores.pca$center,
      t(testscores.pca$loadings[,1:k]))
    %*%t(cbind(1,testscores)))
```

Example: Test Score Data (Con'd)

Compare

```r
head(Z[,1])
```

```
##         1          2          3          4          5          6
##   7.540322 -20.361037  19.503154 -65.965273  -9.778056  33.073953
```

```r
head(Z[,2])
```

```
##         1         2         3         4         5         6
## 10.216765 13.346034  6.555244  1.313665  6.068014 -4.372231
```

```r
head(testscores.pca$scores[,1])
```

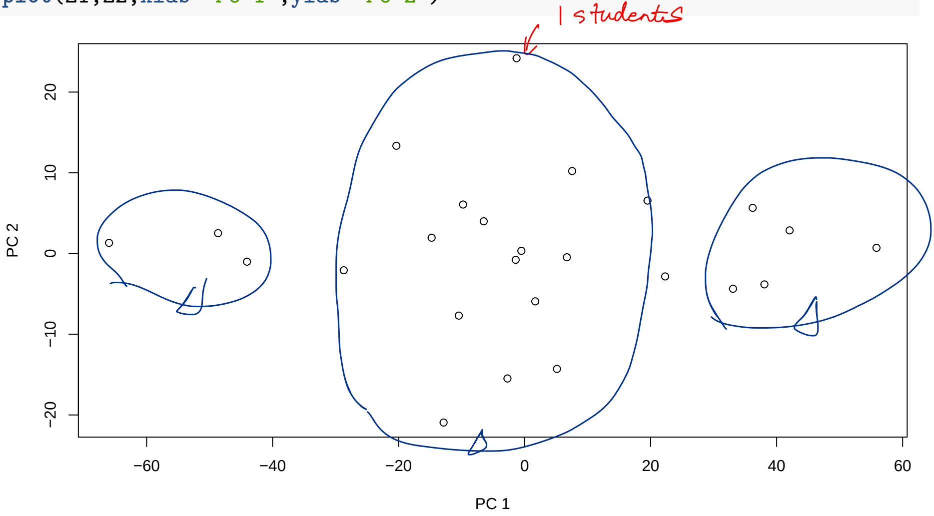
```
##         1          2          3          4          5          6
##   7.540322 -20.361037  19.503154 -65.965273  -9.778056  33.073953
```

```r
head(testscores.pca$scores[,2])
```

```
##         1         2         3         4         5         6
## 10.216765 13.346034  6.555244  1.313665  6.068014 -4.372231
```

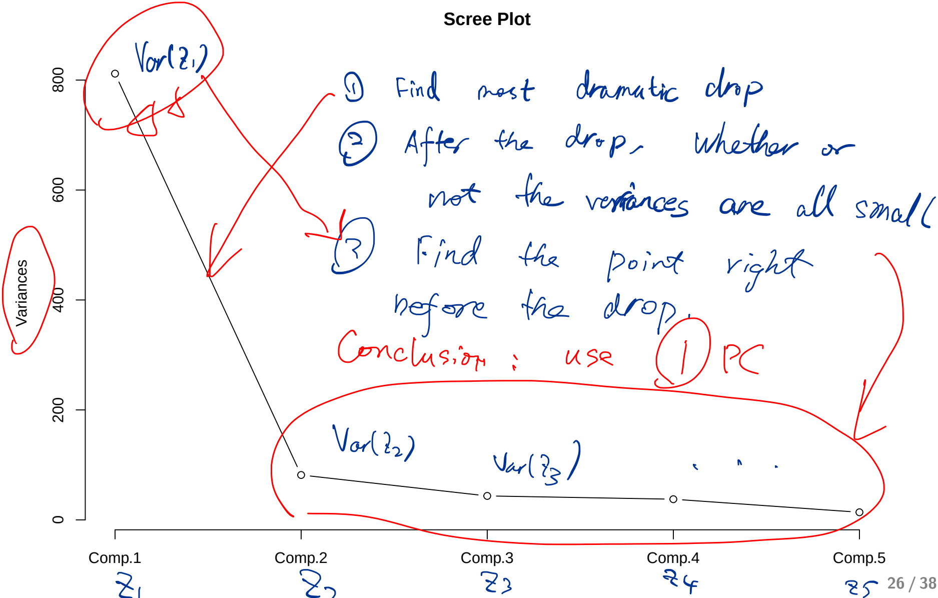
Example: Test Score Data (Con'd)

```
Z1=testscores.pca$scores[,1]
Z2=testscores.pca$scores[,2]
plot(Z1,Z2,xlab="PC 1",ylab="PC 2")
```



Determining the Number of Principal Components

```
screeplot(testscores.pca,type='lines',main='Scree Plot')
```



① Find most dramatic drop

② After the drop, whether or not the variances are all small

③ Find the point right before the drop.

Conclusion: use 1 PC

PCA on explanatory variables prior to Regression

PCA is sometimes recommended as a way of avoiding problems of multicollinearity.

on $X_1 \cdots X_k$

In multiple linear regression, PCA can be used to select a small number of uncorrelated variables for use in the regression model.

Example: SAT Data (Con'd)

```
library(Sleuth3)
SATdata=case1201
head(SATdata)
```

```
##          State  SAT Takers Income Years Public Expend Rank
## 1         Iowa 1088      3    326 16.79   87.8  25.60 89.7
## 2 SouthDakota 1075      2    264 16.07   86.2  19.95 90.6
## 3 NorthDakota 1068      3    317 16.57   88.3  20.62 89.8
## 4       Kansas 1045      5    338 16.30   83.9  27.14 86.3
## 5     Nebraska 1045      5    293 17.25   83.6  21.05 88.5
## 6      Montana 1033      8    263 15.91   93.7  29.48 86.4
```

6 , $X_1 \ldots X_6$

```
SATdata=SATdata[-29,] #removing Alaska
n=length(SATdata[,1])
n
```

```
## [1] 49
```

```
#Randomly choose the training data and test data
set.seed(1)
TestIndex=sample(1:n,floor(n*0.1),replace=F)
TestIndex
```

draw 4 balls from balls labeled by 1, 2, 3, ... 49 without replacement

```
## [1] 14 18 27 42
```

$49 \times 0.1 = 4.9$

floor(4.9) = 4

```
SATdataTest=SATdata[TestIndex,]
SATdataTraining=SATdata[-TestIndex,]
YTraining<-SATdataTraining[,2]
XTraining<-SATdataTraining[,-c(1,2)]
```

Example: SAT Data (Con'd)

```r
fit=lm(YTraining~.,data=XTraining)
summary(fit)
```

```
##
## Call:
## lm(formula = YTraining ~ ., data = XTraining)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -50.553 -13.803   0.426  14.014  51.252
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -241.08294  208.14437  -1.158 0.253990
## Takers         0.08336    0.69408   0.120 0.905040
## Income         0.20985    0.15995   1.312 0.197390
## Years         17.31698    6.36423   2.721 0.009765 **
## Public        -0.33927    0.56556  -0.600 0.552148
## Expend         3.68170    0.92470   3.981 0.000298 ***
## Rank           9.87277    2.08782   4.729 3.09e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.66 on 38 degrees of freedom
## Multiple R-squared:  0.9023, Adjusted R-squared:  0.8869
## F-statistic: 58.52 on 6 and 38 DF,  p-value: < 2.2e-16
```

```r
library(car)
vif(fit)
```

```
##    Takers    Income     Years    Public    Expend      Rank
## 17.530307  3.168006  1.494717  2.288297  1.504407 14.290468
```

Example: SAT Data (Con'd)

One way to solve the multicollinearity problem is to use PCA.

$X_1 \cdots X_6$

```
pca = princomp(XTraining)
summary(pca,loadings=T)
```

```
## Importance of components:
##                            Comp.1     Comp.2     Comp.3      Comp.4
## Standard deviation     44.5887106 15.9482569 8.15348132 4.232637337
## Proportion of Variance  0.8532103  0.1091523 0.02852939 0.007688266
## Cumulative Proportion   0.8532103  0.9623626 0.99089198 0.998580248
##                             Comp.5       Comp.6
## Standard deviation     1.725883075 0.5741415323
## Proportion of Variance 0.001278289 0.0001414634
## Cumulative Proportion  0.999858537 1.0000000000
##
## Loadings:
##        Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6
## Takers  0.397  0.828 -0.210  0.149  0.300
## Income -0.907  0.367 -0.194
## Years                                     0.994
## Public        -0.288 -0.926  0.207
## Expend               -0.247 -0.962
## Rank          -0.296                0.946
```

Example: SAT Data (Con'd)

Looking at the loadings ($k = 6$ in this case)

$$\begin{pmatrix} C_{10} & C_{11} & \cdots & C_{1k} \\ \vdots & \vdots & \vdots & \vdots \\ C_{k0} & C_{k1} & \cdots & C_{kk} \end{pmatrix} =$$

```r
k=6
round(cbind(-t(pca$loadings[,1:k])
      %*%pca$center,
      t(pca$loadings[,1:k])),4)
```

```
##                  Takers  Income   Years  Public  Expend    Rank
## Comp.1 253.8407  0.3972 -0.9073 -0.0036  0.0945  0.0206 -0.0987
## Comp.2 -83.8376  0.8281  0.3668  0.0068 -0.2883  0.0950 -0.2957
## Comp.3 142.4120 -0.2101 -0.1936  0.0232 -0.9257 -0.2465 -0.0046
## Comp.4 -16.6089  0.1491  0.0666 -0.0651  0.2070 -0.9623 -0.0121
## Comp.5 -84.8807  0.2996  0.0200  0.0896 -0.0781  0.0130  0.9463
## Comp.6 -11.5686 -0.0166  0.0013  0.9935  0.0445 -0.0591 -0.0844
```

Any reasonable interpretations for the first principal component and the second principal component?

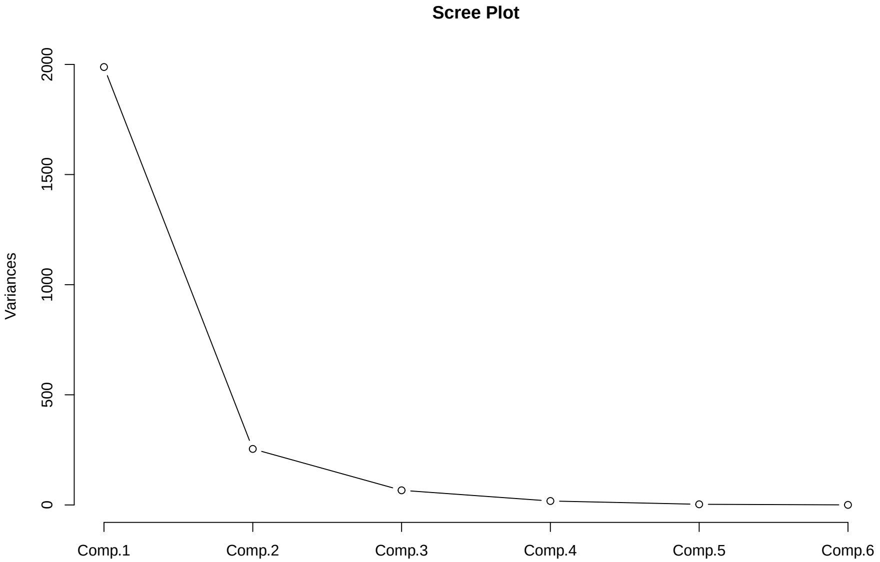
Example: SAT Data (Con'd)

Similarly by using the above loadings, we can compute the PCs of the training dataset.

```
ZTraining=t(cbind(-t(pca$loadings[,1:k])
        %*%pca$center,
        t(pca$loadings[,1:k]))
     %*%t(cbind(1,XTraining)))
```

Example: SAT Data (Con'd)

```
screeplot(pca,type='lines',main='Scree Plot')
```



Example: SAT Data (Con'd)

We might decide to run a regression of Y on the first two principal components, which account for 96% of the total variance.

```
ZTraining.pca2=data.frame(ZTraining[,1:2])
fit.pca2=lm(YTraining~.,data=ZTraining.pca2)
summary(fit.pca2)
```

```
##
## Call:
## lm(formula = YTraining ~ ., data = ZTraining.pca2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -90.781 -22.925  -1.853  23.448  68.606
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 947.9778     5.7406 165.135  < 2e-16 ***
## Comp.1       -1.1509     0.1287  -8.939 2.86e-11 ***
## Comp.2       -2.2085     0.3600  -6.136 2.53e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38.51 on 42 degrees of freedom
## Multiple R-squared:  0.7368, Adjusted R-squared:  0.7242
## F-statistic: 58.78 on 2 and 42 DF,  p-value: 6.713e-13
```

```
vif(fit.pca2)
```

```
## Comp.1 Comp.2
##      1      1
```

Example: SAT Data (Con'd)

By using the loadings, we can also compute the PCs of the test dataset:

```
YTest<-SATdataTest[,2]
XTest<-SATdataTest[,-c(1,2)]
ZTest=t(cbind(-t(pca$loadings[,1:k])
      %*%pca$center,
      t(pca$loadings[,1:k]))
    %*%t(cbind(1,XTest)))
```

z_1, z_2 for test dataset

Compare the mean squared prediction error (MSPE) for the model with the multicollinearity problem and the model constructed by the two principle components.

$X_1 \ldots X_6$

```
YPred=predict(fit,XTest)
MSPE=mean((YTest-YPred)^2)
MSPE
```

```
## [1] 121.3127
```

```
ZTest.pca2=data.frame(ZTest[,1:2])
YPred.pca2=predict(fit.pca2,ZTest.pca2)
MSPE.pca2=mean((YTest-YPred.pca2)^2)
MSPE.pca2
```

```
## [1] 215.0987
```

Example: SAT Data (Con’d)

Try to use all of the principal components:

```
ZTraining.pca6=data.frame(ZTraining)
fit.pca6=lm(YTraining~.,data=ZTraining.pca6)
summary(fit.pca6)
```

```
##
## Call:
## lm(formula = YTraining ~ ., data = ZTraining.pca6)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -50.553 -13.803   0.426  14.014  51.252
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 947.97778    3.67590 257.890  < 2e-16 ***
## Comp.1       -1.15091    0.08244 -13.961  < 2e-16 ***
## Comp.2       -2.20854    0.23049  -9.582 1.10e-11 ***
## Comp.3       -0.29594    0.45084  -0.656    0.516
## Comp.4       -4.83419    0.86846  -5.566 2.24e-06 ***
## Comp.5       10.99797    2.12986   5.164 7.95e-06 ***
## Comp.6       16.13864    6.40242   2.521    0.016 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.66 on 38 degrees of freedom
## Multiple R-squared:  0.9023, Adjusted R-squared:  0.8869
## F-statistic: 58.52 on 6 and 38 DF,  p-value: < 2.2e-16
```

Example: SAT Data (Con'd)

```
vif(fit.pca6)
```

```
## Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6
##      1      1      1      1      1      1
```

Compare MSPE for the model constructed by all of the principle components:

```
ZTest.pca6=data.frame(ZTest)
YPred.pca6=predict(fit.pca6,ZTest.pca6)
MSPE.pca6=mean((YTest-YPred.pca6)^2)
MSPE.pca6
```

```
## [1] 121.3127
```

When is it appropriate to use PCA?

Typically, PCA is used when we have a large number of correlated variables.

In such situations PCA may be able to reduce a large set of variables to a small set that still contains most of the variation information in the large set.

Another advantage of PCA is that the principal components are uncorrelated, so we can talk about one principal component without referring to the others.

One disadvantage of PCA is that the principal components are often difficult to interpret. In such situations it may not be desirable to use the principal components in future analyses such as regression. Also it cannot provide better prediction in regression.

/ L36