Principal Component Analysis II

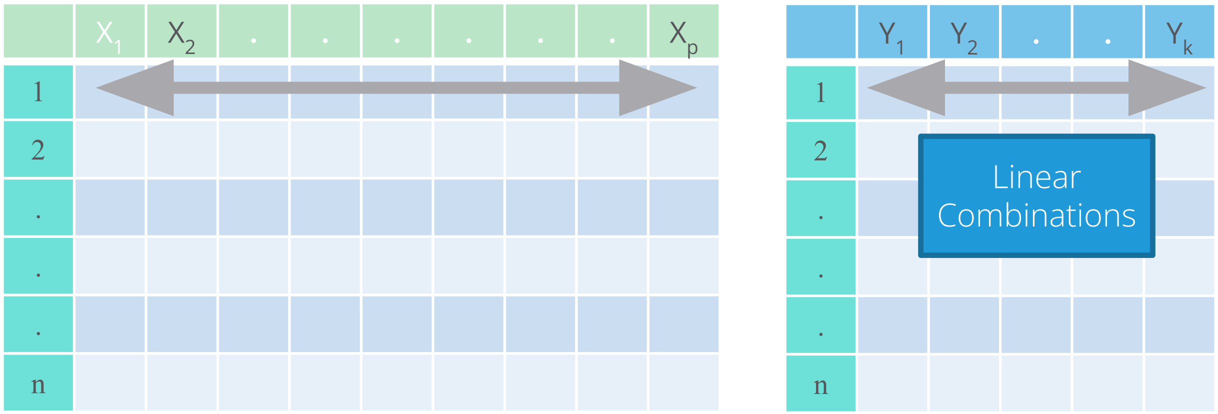
How to Manage Data Dimensionality

Without Losing Information

Contents

Data Reduction

- Summarization of data with p variables by a smaller set of (k) derived variables.
- These k derived variables are **linear combinations of original p variables.**



- In short, **n * p** matrix is **reduced to n * k** matrix.

Case Study – Athletics Records

Background

- Data on national athletics records for various countries is available.

Objective

- To achieve data reduction and obtain score for each country which can be used to rank countries based on athletics records.

Available Information

- Data Source: Applied Multivariate Statistical Analysis by Richard A. Johnson , Dean W. Wichern
- Sample size is 55 countries athletics.
- Records for 8 different athletics events – 100 meters to Marathon

Data Snapshot

Athleticsdata

Variables

Observations

Country	100m_s	200m_s	400m_s	800m_min	1500m_min	5000m_min	10000m_min	Marathon_min
Argentina	10.39	20.81	46.84	1.81	3.7	14.04	29.36	137.72
Australia	10.31	20.06	44.84	1.74	3.57	13.28	27.66	128.3

Column	Description	Type	Measurement	Possible Values
Country	Country Name	Categorical	-	-
100m_s	Time for 100 meter running	Continuous	Seconds	Positive Values
200m_s	Time for 200 meter running	Continuous	Seconds	Positive Values
400m_s	Time for 400 meter running	Continuous	Seconds	Positive Values
800m_min	Time for 800 meter running	Continuous	Minutes	Positive Values
1500m_min	Time for 1500 meter running	Continuous	Minutes	Positive Values
5000m_min	Time for 5000 meter running	Continuous	Minutes	Positive Values
10000m_min	Time for 10000 meter running	Continuous	Minutes	Positive Values
Marathon_min	Time for Marathon running	Continuous	Minutes	Positive Values

PCA in R

```
#Import the data
data<-read.csv("Athleticsdata.csv", header=TRUE)
```

```
athletics<-subset(data,select=c(-Country))

pc<-princomp(formula=~.,data=athletics,cor=T)

summary(pc)
```

- **subset()** is used to remove the variable “Country” from the data.
- **princomp()** from base R performs PCA on the given numeric data matrix.
- **formula=** contains the numeric variables.
  ~. ensures all numeric variables are taken.
- **cor=T** indicates that calculations should be done using the Correlation Matrix. It is equivalent to standardization.

PCA in R

```
# Output:
```

```
Importance of components:
                          Comp.1    Comp.2     Comp.3    Comp.4     Comp.5      Comp.6     Comp.7      Comp.8
Standard deviation     2.5740680 0.9355011 0.39820722 0.3521954 0.28286280 0.260301726 0.21484785 0.149909664
Proportion of Variance 0.8282283 0.1093953 0.01982112 0.0155052 0.01000142 0.008469624 0.00576995 0.002809113
Cumulative Proportion  0.8282283 0.9376236 0.95744470 0.9729499 0.98295131 0.991420937 0.99719089 1.000000000
```

Interpretation:

- The summary function on object pc gives **std. deviation, proportion of variance and cumulative proportion**.

- First Principal Component explains 83% of the variation. Note that 8 PC's are derived using 8 variables but first PC explains most of the variation.

PCA in R –Matrix of Loadings

```
# Component Loadings
pc$loadings
```

- **loadings** are coefficients in linear combinations
- The first column under Comp.1 gives coefficients for first principal component

```
# Output:
```

```
> pc$loadings

Loadings:
              Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
X100m_s        0.318  0.565  0.326  0.129  0.267  0.590  0.154  0.113
X200m_s        0.337  0.462  0.369 -0.257 -0.157 -0.648 -0.128 -0.102
X400m_s        0.356  0.249 -0.561  0.650 -0.221 -0.158
X800m_min      0.369        -0.531 -0.482  0.540        -0.237
X1500m_min     0.373 -0.140 -0.155 -0.407 -0.491  0.143  0.608  0.143
X5000m_min     0.364 -0.312  0.190        -0.250  0.155 -0.593  0.543
X10000m_min    0.367 -0.307  0.182        -0.128  0.232 -0.165 -0.796
Marathon_min   0.342 -0.440  0.260  0.300  0.493 -0.329  0.393  0.160

               Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
SS loadings     1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000
Proportion Var  0.125  0.125  0.125  0.125  0.125  0.125  0.125  0.125
Cumulative Var  0.125  0.250  0.375  0.500  0.625  0.750  0.875  1.000
```

Interpretation:

- First Principal Component can be interpreted as ‘general athletics skill’ since all variables have similar loadings.

Deriving Scores Using PCA

```
# Adding PCA scores to original data as a new variable:
```

```
data$performance<-pc$score[,1]
head(data)
```

```
# Output:
```

```
> data$performance<-pc$score[,1]
> head(data)
    Country X100m_s X200m_s X400m_s X800m_min X1500m_min X5000m_min X10000m_min Marathon_min performance
1 Argentina   10.39   20.81   46.84      1.81       3.70      14.04       29.36       137.72   0.2656535
2 Australia   10.31   20.06   44.84      1.74       3.57      13.28       27.66       128.30  -2.4669681
3   Austria   10.44   20.81   46.82      1.79       3.60      13.26       27.72       135.90  -0.8134149
4   Belgium   10.34   20.68   45.04      1.73       3.60      13.22       27.45       129.95  -2.0582394
5   Bermuda   10.28   20.58   45.91      1.80       3.75      14.68       30.55       146.62   0.7471461
6    Brazil   10.22   20.43   45.21      1.73       3.66      13.62       28.62       133.13  -1.5710562
```

Interpretation:

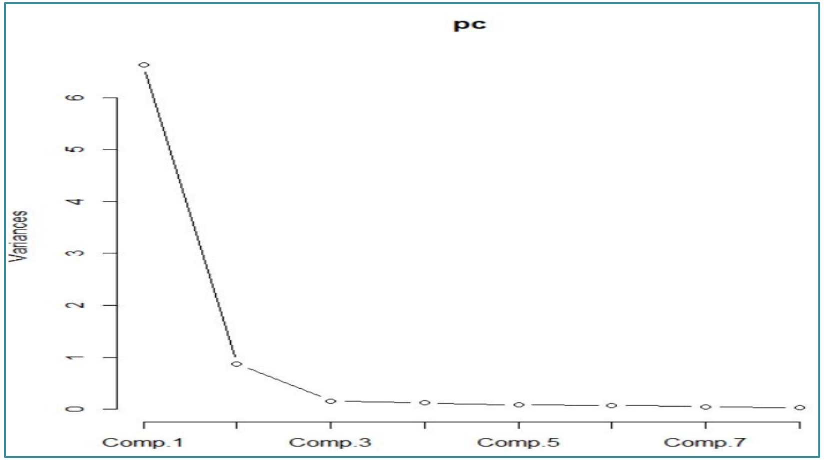
- New column ‘performance’ stores calculated scores using first Principal Component.
- Lower score implies lesser time and hence better athletics performance.

PCA in R - Scree Plot

```
# Scree Plot
plot(pc, type="lines")
```

plot() generates a scree plot

```
# Output:
```



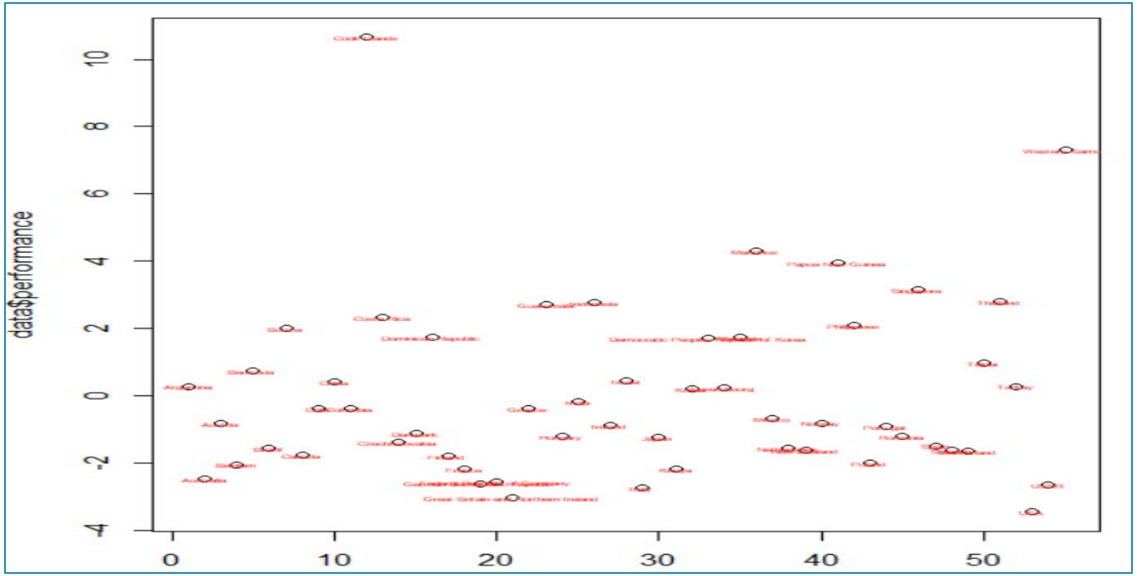
Interpretation:
First Principal Component is sufficient in explaining most of the variation.

```
# Plot of country wise performance
plot(data$performance)
text(data$performance,label=data$Country,col="red",cex=0.4)
```

text() is used to assign names to each points in the plot

PCA in R –Plot of "Performance"

```
# Output-plot of country wise performance plot:
```



Interpretation:

- Athletics from country Cook islands and Western Samoa are performing low since, their score are highest.(lower the score ,better is the performance).

Which are bottom 3 countries?

```
# head function gives countries with highest “performance”
# In our context, these are bottom 3 countries
```

```
top3<-head(data[order(-data$performance),],3)
top3
```

```
# Output :
```

	Country	X100m_s	X200m_s	X400m_s	X800m_min	X1500m_min	X5000m_min	X10000m_min	Marathon_min	performance
12	Cook Isands	12.18	23.20	52.94	2.02	4.24	16.70	35.38	164.70	10.653867
55	Western Samoa	10.82	21.86	49.00	2.02	4.24	16.28	34.71	161.83	7.297965
36	Mauritius	11.19	22.45	47.70	1.88	3.83	15.06	31.77	152.23	4.299192

Which are top 3 countries?

```
# tail function gives top 3 countries
bottom3<-tail(data[order(-data$performance),],3)
bottom3
```

```
# Output :
```

	Country	X100m_s	X200m_s	X400m_s	X800m_min	X1500m_min	X5000m_min	X10000m_min	Marathon_min	performance
29	Italy	10.01	19.72	45.26	1.73	3.60	13.23	27.52	131.08	-2.750446
21	Great Britain and Northern Ireland	10.11	20.21	44.93	1.70	3.51	13.01	27.51	129.13	-3.050287
53	USA	9.93	19.75	43.86	1.73	3.53	13.20	27.43	128.22	-3.460450

Interpretation:

- USA, Britain and Italy are the top three performing countries.
- Cook Islands, Western Samoa and Mauritius are the bottom three countries.

Principal Components Are Uncorrelated

```
# Correlation Matrix of principal components
```

```
round(cor(pc$scores))
```

round(cor()) calculates rounded correlations of the PCA scores.

```
# Output:
```

	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6	Comp.7	Comp.8
Comp.1	1	0	0	0	0	0	0	0
Comp.2	0	1	0	0	0	0	0	0
Comp.3	0	0	1	0	0	0	0	0
Comp.4	0	0	0	1	0	0	0	0
Comp.5	0	0	0	0	1	0	0	0
Comp.6	0	0	0	0	0	1	0	0
Comp.7	0	0	0	0	0	0	1	0
Comp.8	0	0	0	0	0	0	0	1

Interpretation:

- Correlation matrix shows that, principal components are uncorrelated. Diagonal 1’s are the correlation of component to itself.

Quick Recap

Data Reduction and PCA	• PCA reduces n * p matrix to n * k where k is smaller than p
PCA in R	• **`princomp()`** function in base R performs PCA. • Loadings from the summary output are used to derive new variables