Chapter 6

Principal Components and Cluster Analyses

EXAM PA LEARNING OBJECTIVES

8. Topic: Cluster and Principal Component Analyses

Learning Objectives

The candidate will be able to apply cluster and principal components analysis to enhance supervised learning.

a) Understand and apply K-means clustering.

b) Understand and apply hierarchical clustering.

c) Understand and apply principal component analysis.

Chapter overview: Exam PA, as you can tell from its name, is mostly concerned with developing models to "predict" a target variable of interest. In the final chapter of this study manual, we switch our attention to unsupervised learning methods, which ignore the target variable (if present) and look solely at the predictor variables in the dataset to extract their structural relationships. We will learn two unsupervised learning techniques, *principal components analysis* and *cluster analysis*, which are advanced data exploration tools that lend themselves to high-dimensional datasets (i.e., those with a large number of variables) and have the potential of generating useful features for prediction. In a PA project, there is often a task (e.g., Task 3 of the June 2019 PA exam and the Hospital Readmissions sample project) asking you to perform these unsupervised learning techniques and use the output you get to create a new feature for predicting the target variable.

6.1 Principal Components Analysis

6.1.1 Theoretical Foundation

Motivation. In Section 2.2, we performed exploratory data analysis with the aid of summary statistics and graphical displays. At that time, the dataset involved only a few variables, so bivariate data exploration worked well for helping us understand the relationships between different variables. In practice, however, our dataset can involve tens or thousands of variables, many of which are

correlated in subtle ways. For such a high-dimensional dataset, trying to unravel the relationships among the variables by bivariate data exploration is not only a computationally inefficient task, but also unlikely to show us the "big picture" that lies behind the data or produce useful features for constructing predictive models. More advanced data exploration tools are warranted.

Principal[i] *components analysis* (PCA) is an advanced data analytic technique that transforms a large number of possibly correlated variables into a smaller, much more manageable set of representative (hence the qualifier "principal") variables that capture most of the information (in terms of variability) in the original high-dimensional dataset. Known as *principal components* (PCs), these variables are linear combinations of the existing variables and collectively simplify the dataset, reducing its dimension and making it more amenable to data exploration and visualization. In a predictive analytic context, PCA is particularly useful for feature generation.

We now describe how PCA works with the aid of a small amount of mathematical notation in keeping with ISLR without getting bogged down in technical details.

What exactly are the PCs? Throughout this chapter, suppose for concreteness that we are given n observations, each containing the measurements of p features,[ii] $X_1, X_2, \ldots, X_p$ (the target variable, if present, plays no role here, so it is ignored). In many practical situations, p can be very large (e.g., 100 or more). The data matrix is the $n \times p$ matrix given by

$$\mathbf{X} = (x_{ij})_{\substack{i=1,\ldots,n \\ j=1,\ldots,p}} = \begin{array}{c} \\ 1 \\ 2 \\ \vdots \\ n \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & \cdots & p \end{array} \\ \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \end{array}.$$

Here for each x_{ij},

i is the index for the observations, and

j is the index for the variables or features.

Typically, the observations of the features have been centered to have mean zero, i.e., $\bar{X}_j = \sum_{i=1}^{n} X_{ij}/n = 0$ for $j = 1, 2, \ldots, p$. If not, we can use the mean-centered values $x'_{ij} = x_{ij} - \bar{x}_j$ as values of the feature variables.

Mathematically, the PCs are composite variables which are normalized linear combinations of the original set of features. For $m = 1, 2, \ldots, M$, where $M(\leq p)$ is the number of PCs to use, the mth PC is defined by

$$Z_m = \phi_{1m}X_1 + \phi_{2m}X_2 + \cdots + \phi_{pm}X_p = \sum_{j=1}^{p} \phi_{jm}X_j, \tag{6.1.1}$$

where the coefficients $\phi_{1m}, \phi_{2m}, \ldots, \phi_{pm}$ are the *loadings* of the mth PC corresponding to $X_1, X_2, \ldots, X_p$, respectively. To avoid confusion, keep in mind that:

- The PCs are constructed from the features, so the sum above is taken over j, the index for features, but not i.

[i] PCA is commonly misspelled as "principle components analysis." See, for example, the header of Section 8.1 of the PA e-learning modules. Do not make this mistake on the exam!

[ii] In this chapter, we prefer the term "features" rather than "predictors" because there is nothing to predict in an unsupervised setting.

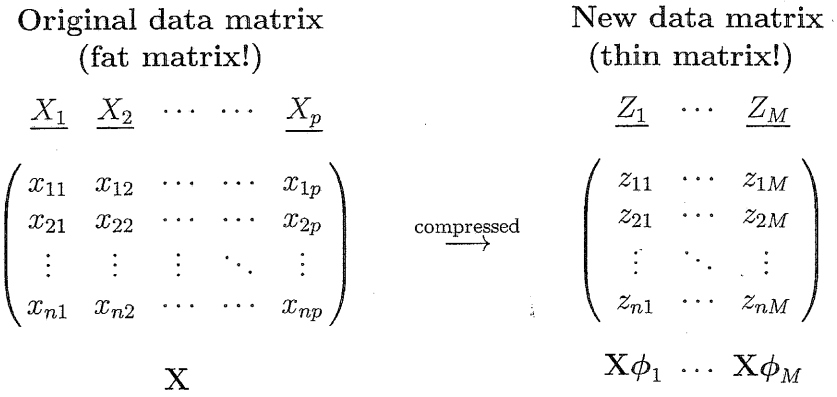


Figure 6.1.1: A pictorial illustration of how PCA achieves dimension reduction.

- The generic symbol ϕ_{jm} for PC loadings is indexed as follows:

$$\begin{aligned} j &: \text{index for the variables (features)} \\ m &: \text{index for PC.} \end{aligned}$$

It is sometimes convenient to collect the p loadings of a PC using the PC *loading vector* given by

$$\boldsymbol{\phi}_m = (\phi_{1m}, \phi_{2m}, \ldots, \phi_{pm})^\top,$$

where "$\top$" stands for transpose. Given the data matrix $\mathbf{X}$, we can compute the realizations of Z_m by

$$\boxed{z_{im} = \phi_{1m}x_{i1} + \phi_{2m}x_{i2} + \cdots + \phi_{pm}x_{ip} = \sum_{j=1}^{p} \phi_{jm}x_{ij}, \quad i = 1, \ldots, n,} \tag{6.1.2}$$

which is the same as (6.1.1) with the observation index i added and with the uppercase letters changed to lowercase letters. We call $z_{1m}, z_{2m}, \ldots, z_{nm}$ the *scores* of the mth PC. If we collect these n scores by a column vector $\mathbf{z}_m = (z_{1m}, z_{2m}, \ldots, z_{nm})^\top$, then the PC scores, loadings, and feature values are related via

$$\boxed{\mathbf{z}_m = \mathbf{X}\boldsymbol{\phi}_m.} \tag{6.1.3}$$

This matrix formula, though not given in ISLR or the PA e-learning modules, stresses the fact that the PC scores result from rotations of the original observations in a direction determined by the loading vector.

Figure 6.1.1 gives a pictorial illustration of how PCA reduces the dimension of the dataset from p features to M features.

How do we go about choosing the PC loadings? For simplicity, let's start with the first PC and select its loadings $\phi_{11}, \phi_{21}, \ldots, \phi_{p1}$. To ensure that the first PC captures as much information in the original dataset as possible, the PC loadings are such that they *maximize the sample variance*[iii] of Z_1, i.e., they solve

$$\max_{\phi_{11},\ldots,\phi_{p1}} \frac{1}{n}\sum_{i=1}^{n} z_{i1}^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{j1}x_{ij}\right)^2 \tag{6.1.4}$$

[iii] Here the sample variance of Z_1 equals its second sample moment because the features values are centered, so are the PC scores.

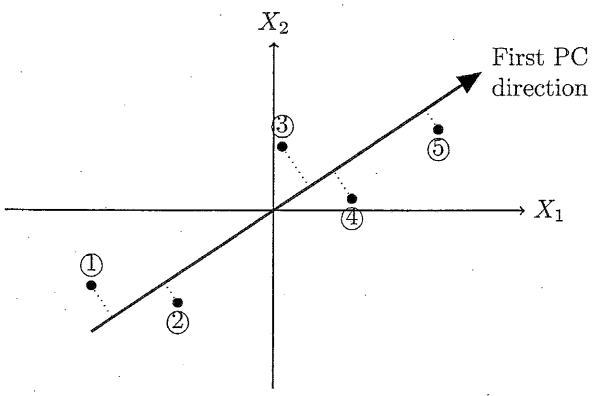


Figure 6.1.2: Geometric illustration of the first PC as the direction along which the data vary the most.

subject to the normalization constraint $\sum_{j=1}^{p} \phi_{j1}^2 = 1$, which makes the variance maximization meaningful, for otherwise we can scale up the ϕ_{j1}'s to inflate the variance arbitrarily. There is no need to worry about how this constrained maximization problem is solved in R, but it is important to keep in mind that the PC loadings are defined to maximize variance. Geometrically, the p loadings $\phi_{11}, \phi_{21}, \ldots, \phi_{p1}$ represent a line in the p-dimensional feature space along which the data vary the most in the sense that the projected points of the n observations on this line, which are the PC scores, are as *spread out* as possible (among all possible lines). Figure 6.1.2 illustrates this with a simplistic dataset with $n = 5$ and $p = 2$.

Given the first PC, subsequent PCs are defined the same way as the maximal variance linear combination, with the additional constraint that they must be *orthogonal* to (or *uncorrelated* with) the previous PCs. To put this in mathematical language, the loadings of the mth PC with $m \geq 2$ maximize

$$\frac{1}{n}\sum_{i=1}^{n} z_{im}^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{jm}x_{ij}\right)^2,$$

subject to the constraints

$$\begin{cases} \displaystyle\sum_{j=1}^{p}\phi_{jm}^2 = 1 & \text{(normalization constraint)} \\ \boldsymbol{\phi}_r^\top \boldsymbol{\phi}_m = \displaystyle\sum_{j=1}^{p}\phi_{jr}\phi_{jm} = 0, \quad r = 1, \ldots, m-1, & \text{(orthogonality constraint)} \end{cases}$$

The orthogonality constraint is imposed so that the subsequent PCs are independent of the previous PCs and serve to measure different aspects of the variables in the dataset.

In Figure 6.1.3, we have included both the first and second PCs to the same toy dataset in Figure 6.1.2. It can be seen that the two PC directions are mutually perpendicular, with the angle between the two lines being 90°.

Proportion of variance explained. Via PCA, we successfully reduce the dimension of the dataset from p variables to a much smaller set of variables (more precisely, the M PCs) that together retain most of the information measured by variance. To quantify how much information can be captured by the PCs, we can assess the proportion of variance explained by each of the M

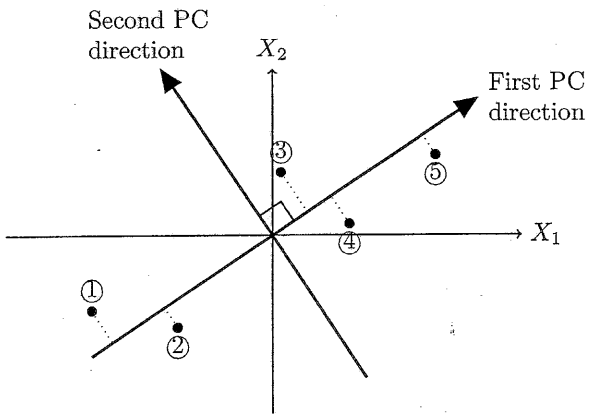


Figure 6.1.3: Geometric illustration of the mutual perpendicularity of the first and second PCs.

PCs in comparison to the total variance. Doing so allows us to address the issue of how many PCs to use in a given setting.

- *Total variance:* The total variance present in the data is the sum of the sample variance of each variable:

$$\underbrace{\sum_{j=1}^{p}}_{\substack{\text{sum over} \\ \text{all features}}} \underbrace{\frac{1}{n}\sum_{i=1}^{n} x_{ij}^2}_{\substack{\text{sample variance} \\ \text{of feature } j}},$$

  which, if the data values have been standardized, equals $\sum_{j=1}^{p}(1) = p$.

- *Variance explained by the mth PC:* As with the variables, the scores of each PC also have zero mean, so the variance explained by the mth PC equals the (sample) second moment:

$$\frac{1}{n}\sum_{i=1}^{n} z_{im}^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{jm}x_{ij}\right)^2.$$

As a result, the *proportion of variance explained* (PVE) by the mth PC is given by

$$\text{PVE}_m = \frac{\text{variance explained by } m\text{th PC}}{\text{total variance}} = \frac{\sum_{i=1}^{n} z_{im}^2}{\sum_{i=1}^{n}\sum_{j=1}^{p} x_{ij}^2} \overset{\text{(for standardized data)}}{=} \frac{\sum_{i=1}^{n} z_{im}^2}{np}. \quad (6.1.5)$$

These proportions for $m = 1, \ldots, M$ satisfy several properties:

- Each PVE_m can be easily shown to be between 0 and 1, i.e., it is indeed a proportion.

- By the definition of PCs, the PVE_m's are monotonically decreasing in m, i.e., $\text{PVE}_1 \geq \text{PVE}_2 \geq \cdots \geq \text{PVE}_M$, meaning that the proportion of variance explained decreases with the index of the PC. The first PC can explain the greatest amount of variance, followed by the second PC, third PC, and so forth. This is because subsequent PCs have more and more orthogonality constraints to comply with and therefore less flexibility with the choice of the PC loadings.

Example 6.1.1. (CAS Exam MAS-II Fall 2018 Question 41: Calculation of PVE) You are provided with the following normalized and scaled data set:

i	X_1	X_2	X_3
1	−0.577	1	−1
2	−0.577	1	1
3	−0.577	−1	1
4	1.732	−1	−1

The first principal component loading vector of the data set is $(0.707, -0.500, -0.500)$. Calculate the proportion of variance explained by the first principal component.

(A) Less than 53%

(B) At least 53% but less than 58%

(C) At least 58% but less than 63%

(D) At least 63% but less than 68%

(E) At least 68%

Solution. Given that $\phi_1 = (0.707, -0.5, -0.5)^\top$, the scores of the first PC, by (6.1.3), are

$$z_1 = \begin{pmatrix} -0.577 & 1 & -1 \\ -0.577 & 1 & 1 \\ -0.577 & -1 & 1 \\ 1.732 & -1 & -1 \end{pmatrix} \begin{pmatrix} 0.707 \\ -0.5 \\ -0.5 \end{pmatrix} = \begin{pmatrix} -0.407939 \\ -1.407939 \\ -0.407939 \\ 2.224524 \end{pmatrix}.$$

Since the dataset has been scaled, the PVE by the first PC, by (6.1.5), is

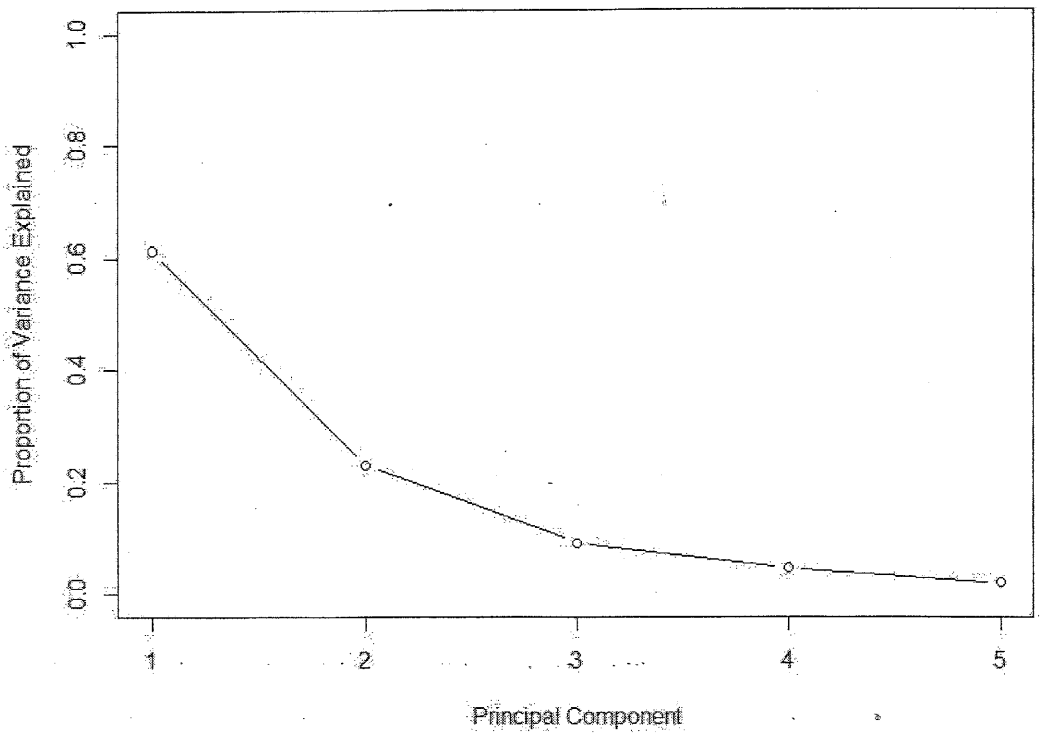
$$\frac{z_{11}^2 + z_{21}^2 + z_{31}^2 + z_{41}^2}{4(3)} = \boxed{60.54\%}. \quad \textbf{(Answer: (C))}$$

□

Choosing the number of PCs by a scree plot. Given the PVEs, a *scree plot* provides a simple visual inspection method for selecting the number of PCs to use. Such a plot depicts the PVE of each of the M PCs. Its horizontal axis shows the index of the PCs whereas the vertical axis shows the PVE. As discussed above, the PVE will decrease as we move along the horizontal axis of a scree plot. The number of PCs required is typically chosen by eyeballing the scree plot and looking for the point at which the PVE has dropped off to a sufficiently low level (which can be subjective). That point is known as an *elbow* in the scree plot. The PCs beyond the elbow have a very small PVE and can be safely dropped.

Example 6.1.2. (SOA Exam SRM Sample Question 35: Determining the number of PCs from a scree plot) Using the following scree plot, determine the minimum number

of principal components that are needed to explain at least 80% of the variance of the original dataset.



(A) One

(B) Two

(C) Three

(D) Four

(E) It cannot be determined from the information given.

Solution. From the scree plot, we can see that the first PC explains about 62% of the variance and the second PC explains about 23% of the variance. Together they explain about 85% of the variance, and hence $\boxed{\text{two}}$ PCs are sufficient. **(Answer: (B))** □

Example 6.1.3. (SOA Exam SRM Sample Question 6: True-or-false statements about PVE) Consider the following statements:

I. The proportion of variance explained by an additional principal component increases as more principal components are added.

II. The cumulative proportion of variance explained increases as more principal components are added.

III. Using all possible principal components provides the best understanding of the data.

IV. A scree plot provides a method for determining the number of principal components to use.

Determine which of the statements are correct.

(A) Statements I and II only

(B) Statements I and III only

(C) Statements I and IV only

(D) Statements II and III only

(E) Statements II and IV only

Solution. I. Incorrect. The PVE by an additional PC decreases as more PCs are added.

II. Correct. The cumulative PVE increases as more PVEs are added.

III. Incorrect. We want to use the least number of PCs required to gain the best understanding of the data. If all possible PCs are used, then it is the identical to using all of the original features.

IV. Correct. Typically, the number of PCs are chosen based on a scree plot. **(Answer: (E))** □

[IMPORTANT!] Feature generation using PCA. In the context of Exam PA, the most important application of PCA is to *create useful features* for predicting a given target variable. Once we have settled on the number of PCs to use (typically, we will use just the first few), the original variables $X_1, \ldots, X_p$ are replaced by the PCs $Z_1, \ldots, Z_M$, which capture most of the information in the dataset and serve as predictors for the target variable. These predictors are, by construction, mutually uncorrelated, so collinearity is no longer an issue. By reducing the dimension of the data and the complexity of the model, we hope to optimize the bias-variance trade-off and improve the prediction accuracy of the model. We will see how to create new features based on PCA in the case study in the next subsection.

When it comes to fitting a predictive model using the PCs as predictors, it is important to delete the original variables $X_1, \ldots, X_p$ to avoid any duplication of information. If you are fitting a GLM, the co-existence of the PCs and the original variables will result in a rank-deficient model.

Remark 1: Importance of centering and scaling. In addition to centering the features so that they all have a zero mean, it is common to scale them to produce unit standard deviation prior to performing a PCA. Whether the scaling is done can have a substantial effect on the results of the PCA. If no scaling is done, then those variables with an unusually *large variance* on their scale will receive a *large PC loading* (remember that the loadings are defined to maximize the sample variance, so it makes sense to attach a large weight to a high-variance variable) and dominate the corresponding PC, even though it may not be a variable that explains much of the underlying

pattern in the data.

Example 6.1.4. (SOA Exam SRM Sample Question 37: What are the possible reasons for having different PC loadings?) Analysts W, X, Y, and Z are each performing principal components analysis on the same data set with three variables. They use different programs with their default settings and discover that they have different factor loadings for the first principal component. Their loadings are:

	Variable 1	Variable 2	Variable 3
W	−0.549	−0.594	0.587
X	−0.549	0.594	0.587
Y	0.549	−0.594	−0.587
Z	0.140	−0.570	−0.809

Determine which of the following is/are plausible explanations for the different loadings.

I. Loadings are unique up to a sign flip and hence X's and Y's programs could make different arbitrary sign choices.

II. Z's program defaults to not scaling the variables while Y's program defaults to scaling them.

III. Loadings are unique up to a sign flip and hence W's and X's programs could make different arbitrary sign choices.

(A) None

(B) I and II only

(C) I and III only

(D) II and III only

(E) The correct answer is not given by (A), (B), (C), or (D).

Solution. I. True. Note that the PC loading vectors ϕ_m's are *unique up to a sign flip*. This means that if, for example, ϕ_1 is a PC loading vector, then $-\phi_1$ is also a PC loading vector (the negative sign will vanish after squaring in (6.1.4)). Geometrically, flipping the sign of a PC loading vector does not change the direction it represents and therefore gives rise to another valid PC loading vector.

Back to this example, the uniqueness up to a sign flip means that all three signs must be flipped. This is true for X and Y.

II. True. We can see that the PC loadings for Y and Z differ quite widely in size, which can be the result of the presence or absence of scaling.

III. False. The uniqueness up to a sign flip requires that all three signs be flipped. For W and X, only the second loading is flipped. **(Answer: (B))**

□

Remark 2: PCA with categorical features. PCA, in its current form, can only be applied to numeric variables. To perform PCA on categorical variables, they have to be explicitly binarized (with the use of the `dummyVars()` function in the `caret` package we used in Chapters 3 and 4) in advance to create the dummy variables, which are numeric, taking values of either 0 or 1. Then all of the discussions above apply equally well.

6.1.2 Simple Case Study

In this subsection we illustrate the concepts presented in Subsection 6.1.1 by means of a simple case study. After completing this case study, you should be able to:

- Perform a PCA using the `prcomp()` function in R.
- Interpret the output of a PCA and extract useful components from a `prcomp` object.
- Realize the importance of scaling on the results of a PCA.
- Use the output of a PCA to create useful features for prediction.

Data description. This case study centers on the well-known `USArrests` dataset that comes with base R. For each of the $n = 50$ US states in 1973, this dataset hosts information about $p = 4$ variables:

- `Murder`: The number of arrests for murder per 100,000 residents
- `Assault`: The number of arrests for assault per 100,000 residents
- `Rape`: The number of arrests for rape per 100,000 residents
- `UrbanPop`: The percent (from 0 to 100) of the population residing in urban areas

One would expect that the three crime-related variables are rather closely related and it may make sense to combine them into a single variable by means of PCA to reduce the dimension of the data.

TASK 1: Perform univariate exploration of the four variables

Use graphical displays and summary statistics to determine if any of these variables should be transformed and, if so, what transformation should be made. Do your recommended transformations, if any, and delete the original variables.

Before performing a PCA, let's first run CHUNK 1 to load the dataset and print a summary for each of the four variables.

```
# CHUNK 1
data(USArrests)
summary(USArrests)
```

```
##      Murder          Assault         UrbanPop          Rape
##  Min.   : 0.800   Min.   : 45.0   Min.   :32.00   Min.   : 7.30
##  1st Qu.: 4.075   1st Qu.:109.0   1st Qu.:54.50   1st Qu.:15.07
##  Median : 7.250   Median :159.0   Median :66.00   Median :20.10
##  Mean   : 7.788   Mean   :170.8   Mean   :65.54   Mean   :21.23
##  3rd Qu.:11.250   3rd Qu.:249.0   3rd Qu.:77.75   3rd Qu.:26.18
##  Max.   :17.400   Max.   :337.0   Max.   :91.00   Max.   :46.00
```

We can see that the four variables are on drastically different scales, even among the three crime-related variables. In CHUNK 2, we use the `apply()` function (introduced on page 43) to "apply" the `mean()` and `sd()` functions to return the mean and standard deviation of each variable.

```
# CHUNK 2
apply(USArrests, 2, mean)

##   Murder  Assault UrbanPop     Rape
##    7.788  170.760   65.540   21.232

apply(USArrests, 2, sd)

##    Murder   Assault  UrbanPop      Rape
##  4.355510 83.337661 14.474763  9.366385
```

That the four standard deviations differ substantially makes it imperative to scale the four variables prior to performing a PCA (which can be easily achieved by setting an option in an R function, as we will see soon).

To learn more about the overall distribution of the four variables, in CHUNK 3 we use a `for` loop to produce a histogram for each (see Figure 6.1.4). None of the four graphs indicates a particularly problematic distribution, but there is a small amount of right skewness in the distribution of `Rape`. A log transformation[iv] is applied, with the new variable called `logRape`, and the original variable deleted in CHUNK 4.

[iv] If you are interested, you may skip CHUNK 4 and see how applying or not applying the log transformation to `Rape` affects the results of PCA.

```
# CHUNK 3
library(ggplot2)

# names(USArrests) extracts the column names of the USArrests data
for (i in names(USArrests)) {
  plot <- ggplot(USArrests, aes(x = USArrests[, i])) +
    geom_histogram() +
    xlab(i)
  print(plot)
}
```

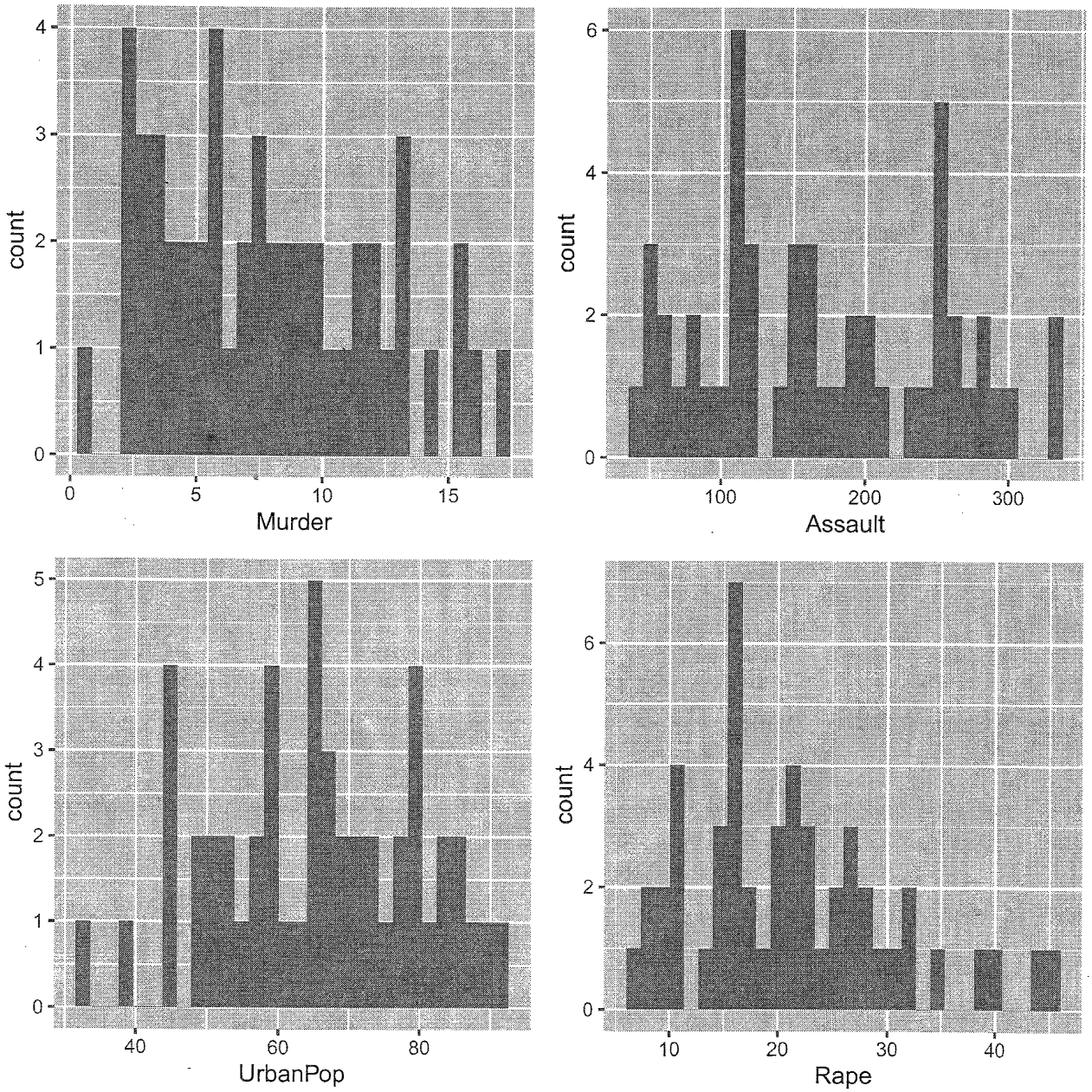


Figure 6.1.4: Histograms for the four variables in the `USArrests` dataset.

```
# CHUNK 4
USArrests$logRape <- log(USArrests$Rape)
USArrests$Rape <- NULL
summary(USArrests)

##      Murder          Assault        UrbanPop        logRape
##  Min.   : 0.800   Min.   : 45.0   Min.   :32.00   Min.   :1.988
##  1st Qu.: 4.075   1st Qu.:109.0   1st Qu.:54.50   1st Qu.:2.713
##  Median : 7.250   Median :159.0   Median :66.00   Median :3.001
##  Mean   : 7.788   Mean   :170.8   Mean   :65.54   Mean   :2.959
##  3rd Qu.:11.250   3rd Qu.:249.0   3rd Qu.:77.75   3rd Qu.:3.265
##  Max.   :17.400   Max.   :337.0   Max.   :91.00   Max.   :3.829
```

The summary shows that the mean and median of the log-transformed Rape variable are almost identical.

TASK 2: Interpret the results

Perform a principal components analysis (PCA) on the USArrests data and interpret the output, including the loadings on significant principal components.

Implementing a PCA. There are several functions in R for doing PCA. The one we will use in Exam PA is the `prcomp()` function, which is part of base R (though it is not the most versatile one). This function takes a numeric matrix or a data frame carrying the variables to which PCA is to be applied (if you want to perform PCA on only certain variables in the data frame, subset it using the square bracket `[, ]` notation; recall how this works as discussed on page 28). To incorporate mean-centering and scaling, we set `center = TRUE` and `scale. = TRUE`. By default, `center = TRUE` and `scale. = FALSE`.

In CHUNK 5, we run a PCA on the USArrests data and print a summary.

```
# CHUNK 5
PCA <- prcomp(USArrests, center = TRUE, scale. = TRUE)
summary(PCA)

## Importance of components:
##                           PC1    PC2     PC3     PC4
## Standard deviation     1.5940 0.9930 0.54644 0.41769
## Proportion of Variance 0.6352 0.2465 0.07465 0.04362
## Cumulative Proportion  0.6352 0.8817 0.95638 1.00000
```

Interpretation of PCA output. When the summary() function is applied to a prcomp object, we get the PVE and the cumulative PVE by each PC. For instance, the first PC explains 63.52% of the variability and the second PC explains 24.65%, for a cumulative PVE of 88.17%. It seems that the first two PCs together are able to explain most of the variability in the data.

Note that a `prcomp` object is a list hosting a lot of useful information that we can access using the dollar sign `$` notation. The two most important components are:[v]

`rotation` This is the $p \times M$ numeric matrix of PC loadings indexed by variables (row) and PCs (column). Each column of the matrix provides the corresponding PC loading vector ϕ_m. The matrix is called `rotation` because the PC loadings define the "rotated" coordinate system.

`x` This is the $n \times M$ numeric matrix of PC scores indexed by observations (row) and PCs (column). Despite its nomenclature, it is not the original data matrix $\mathbf{X}$, but the new data matrix whose columns are the PC scores computed in accordance with (6.1.3).

These two components are accessed in CHUNK 6.

```
# CHUNK 6
PCA$rotation

##                PC1        PC2        PC3        PC4
## Murder   -0.5377481  0.4076787 -0.2260763 -0.7025059
## Assault  -0.5725819  0.1859535 -0.4175757  0.6805893
## UrbanPop -0.2766636 -0.8817037 -0.3293236 -0.1939118
## logRape  -0.5535650 -0.1477092  0.8161287  0.0753777

PCA$x

##                   PC1         PC2         PC3          PC4
## Alabama     -1.08857173  1.08040146 -0.26485606 -0.223275549
## Alaska      -1.59517817  1.20815437  1.33114413  0.770869203
## Arizona     -1.74286818 -0.73170358 -0.10577896  0.841565601
## Arkansas     0.02590682  1.08051464  0.22523823  0.203985489
## California  -2.27067947 -1.44575752  0.17412539  0.446993002
## Colorado    -1.33306093 -0.90185347  0.80129408  0.202583949
## Connecticut  1.42803044 -1.07347295 -0.71924115 -0.017836594
## Delaware    -0.10887464 -0.35521443 -0.74485862  0.733947673
## Florida     -2.96086941  0.03372143 -0.63849855  0.003739992
## Georgia     -1.71387288  1.23179720 -0.04883230 -1.098938437
## Hawaii       0.77349867 -1.59004854  0.44126511 -0.843701602
## Idaho        1.58397955  0.20390459  0.23460177  0.525892530
## Illinois    -1.46182554 -0.71600668 -0.52965170  0.020262855
## Indiana      0.37510248 -0.17895295  0.48652752 -0.355380434
## Iowa         2.29527344 -0.08449260  0.09567755 -0.010508842
## Kansas       0.67903028 -0.29739272  0.23794778 -0.184578859
## Kentucky     0.65242081  0.92071803  0.21550950 -0.659335002
## Louisiana   -1.65881774  0.81297569 -0.54304971 -0.571442615
## Maine        2.69037404  0.45290210 -0.56662933  0.244736056
## Maryland    -1.79738288  0.40865125 -0.20272321  0.530434707
```

[v]The other three components are `sdev`, `center`, and `scale`, which are respectively the standard deviation of each PC, and the mean and standard deviation of the original variables.

```
## Massachusetts   0.40120822 -1.49624706 -0.46061752  0.080089078
## Michigan       -2.00608208 -0.11940052  0.24260927 -0.021023047
## Minnesota       1.61318004 -0.64050835  0.28371243 -0.034976506
## Mississippi    -1.07406270  2.32612244 -0.59983479 -0.351446592
## Missouri       -0.75002114 -0.26625605  0.48547695 -0.132735565
## Montana         1.08263952  0.51148535  0.39584336 -0.074934548
## Nebraska        1.16116223 -0.21345845  0.32536046  0.022538197
## Nevada         -2.46253447 -0.63142774  0.58107443 -0.110366833
## New Hampshire   2.53217873  0.02594468 -0.19441144 -0.001727117
## New Jersey     -0.28894798 -1.48337457 -0.50002622 -0.351929111
## New Mexico     -1.93999527  0.15484342  0.05844813  0.375579694
## New York       -1.74258521 -0.84945712 -0.50796826 -0.078021024
## North Carolina -1.17261108  2.16876574 -0.96121883  0.762118048
## North Dakota    3.32682130  0.69445403 -0.26900260  0.226819396
## Ohio            0.10043818 -0.76926601  0.25278616 -0.445164553
## Oklahoma        0.19045838 -0.31713206  0.17099230  0.003407881
## Oregon         -0.10276031 -0.52216855  0.93078036  0.419958030
## Pennsylvania    0.82042771 -0.59315984 -0.21001998 -0.418338494
## Rhode Island    1.14053009 -1.43552698 -1.79704434  0.306299281
## South Carolina -1.41385435  1.87836907 -0.20771937  0.078222445
## South Dakota    1.96845640  0.82246941  0.36018752  0.157951039
## Tennessee      -1.06927613  0.83463470  0.38249048 -0.588994500
## Texas          -1.43284496 -0.44488373 -0.23090006 -0.692415319
## Utah            0.42817534 -1.47971429  0.47405855  0.160435090
## Vermont         2.83895331  1.42338830  0.68851605  0.257590415
## Virginia       -0.02500370  0.16519932  0.22313739 -0.189497685
## Washington      0.12670289 -0.96661387  0.70939295  0.351774974
## West Virginia   2.27377958  1.45893004 -0.15313645 -0.162180405
## Wisconsin       2.14985617 -0.58718814 -0.19646165 -0.227634892
## Wyoming         0.55399633  0.29233150 -0.15571676  0.118589497
```

We can see, for example, that the loadings of the first PC are $\phi_{11} = -0.5377$, $\phi_{21} = -0.5726$, $\phi_{31} = -0.2767$, $\phi_{41} = -0.5536$, which means that the first PC is defined by

$$Z_1 = -0.5377(\texttt{Murder}') - 0.5726(\texttt{Assault}') - 0.2767(\texttt{UrbanPop}') - 0.5536(\texttt{logRape}'),$$

where the four variables have been centered and standardized. Of course, we have $\phi_{11}^2 + \phi_{21}^2 + \phi_{31}^2 + \phi_{41}^2 = 1$.

In case you are interested (hopefully so!), let's verify the first PC score for Alabama, the very first observation in the `USArrests` data. From the first row of the data, Alabama has `Murder` $= 13.2$, `Assault` $= 236$, `UrbanPop` $= 58$, `logRape` $= 3.054001$ (before centering and standardization), so its first PC score is (the means and standard deviations of the four variables are taken from CHUNK 2)

$$\begin{aligned} z_{11} &= -0.5377\left(\frac{13.2 - 7.7880}{4.3555}\right) - 0.5726\left(\frac{236 - 170.7600}{83.3377}\right) \\ &\quad -0.2767\left(\frac{58 - 65.5400}{14.4748}\right) - 0.5536\left(\frac{3.054001 - 2.9590}{0.4524}\right) = -1.0885, \end{aligned}$$

which is the $(1,1)$-entry of the `x` matrix (subject to some rounding error). Other PC scores in the `x` matrix are computed in a similar fashion.

Exercise 6.1.1. (First PVE) Verify that the proportion of variance explained by the first PC is 63.52%.

Solution. From CHUNK 5, the (sample) standard deviation of the first PC is 1.5940. Because we perform scaling when doing PCA, we can use the rightmost formula in (6.1.5) to calculate the proportion of variance explained by the first PC as

$$\text{PVE}_1 = \frac{\overbrace{\sum_{i=1}^{n} z_{i1}^2/n}^{\text{variance of first PC}}}{p} = \frac{\overbrace{1.5940^2}^{\text{square st. dev. to get variance}}}{4} = 0.6352.$$

□

Remark. As a matter of fact, R uses division by $n-1$ to get the sample standard deviation, but whether you use n or $n-1$ makes hardly any difference for most datasets.

Taking a closer look at the PC loadings. The results of a PCA become particularly meaningful if we take a closer look at the PC loadings, paying attention to their relative sign and magnitude. This allows us to understand how the original variables contribute to the construction of different PCs and uncover potentially interesting relationships among the variables.

Examining the PC loadings in CHUNK 6, we can see that the first PC attaches roughly the same weight to each of `Assault`, `Murder`, and `logRape`, so it can be interpreted as a measure of overall crime rates. In Task 3 below, we will use this insight to combine these three crime-related variables as a single variable without losing much information. In contrast, the second PC puts a very heavy (negative) weight on `UrbanPop` and a rather small weight on the three crime-related variables, so it is mainly a measure of urbanization level.

Exercise 6.1.2. (Interpreting the first two PCs) Consider the following output of a principal components analysis run on seven variables:

	PC1	PC2	PC3	PC4
X_1	0.30	0.65	0.08	0.59
X_2	0.42	0.16	−0.34	0.00
X_3	0.39	0.07	0.55	−0.48
X_4	0.40	0.30	−0.21	−0.52
X_5	0.42	−0.25	−0.17	0.23
X_6	0.35	−0.51	−0.40	0.07
X_7	0.34	−0.37	0.59	0.30

Interpret the first two principal components.

Solution. The loadings of the first PC share the same sign and approximately the same size, so it appears to be a simple average of all seven variables. The second PC can be considered as a *contrast* between X_1, X_2, X_3, X_4 (with positive loadings) and X_5, X_6, X_7 (negative loadings), with X_1 and X_6 being the most dominant variables. □

Biplot. As soon as the PCs have been computed, we can plot them against one another to produce a low-dimensional view of the data. In particular, plotting the first two PCs against each other allows us to visualize the data on a two-dimensional scatterplot and visually tell which observations are similar to each other. In CHUNK 7, we use the `biplot()` function to create a biplot (see Figure 6.1.5). It is called a *bi*plot because it displays the first two PC scores Z_1 and Z_2 against each other for the 50 states (identified by their names) along with the directions represented by the first two PC loading vectors. For instance, `Murder` is represented by the arrow heading to the north west and passing through $(\phi_{11}, \phi_{12}) = (-0.5377481, 0.4076787)$, where the two loadings are from CHUNK 6.

Looking at the biplot, we can see which states have a high crime rate and high level of urbanization. Florida, for example, is sitting on the far left end on the first PC axis, i.e., its first PC score is very negative (remember, the loadings of the first PC are negative, so the higher the value of `Murder`, `Assault`, and `logRape`, the more negative the PC score), and is a high-crime area. In the same vein, California has both high crime rates and a high level of urbanization because it has large negative scores for both PCs. That the directions of the three crime variables are rather similar indicates that these three variables are rather positively correlated. Moreover, `UrbanPop` is less correlated with the other three crime variables as its direction differs significantly from the directions of the three variables. Note that because the variables in the dataset have been centered, the average PC scores are zero.

By way of illustration, let's look at the biplot should we perform the PCA without scaling (not recommended!). This is done in CHUNK 8 with the biplot shown in Figure 6.1.6. Compared to Figure 6.1.5, it can be seen that in the absence of scaling, the first and second PC loading vectors place almost all of their weights respectively on `Assault` and `UrbanPop`, which is not surprising given that `Assault` has a disproportionately large standard deviation, followed by `UrbanPop`, as shown in CHUNK 2. Since it is undesirable for the results obtained to depend on an arbitrary choice of scale, it is typical to standardize the features so that all of them are on the same scale prior to performing a PCA.

TASK 3:
Use observations from principal components analysis (PCA) to generate a new feature

Generate one new feature based on your observations from Task 2 (which may also involve dropping some current variables).

[IMPORTANT!] Feature generation based on PCA. Given the results of the PCA in Task 2, how can we create new useful features? We have seen in CHUNK 6 and Figure 6.1.5 that the three crime-related variables are the dominating variables in the first PC, all sharing approximately the same weight. The first PC alone can also explain 63.52% of the variability of the data, so combining the three crime-related variables as a single feature measuring overall crime rates is a judicious move.

```
# CHUNK 7
# cex argument indicates the amount by which plotting symbols should be scaled
# cex = 0.6 means 40% smaller
# scale = 0 ensures that the arrows are scaled to represent the loadings
biplot(PCA, scale = 0, cex = 0.6)
```

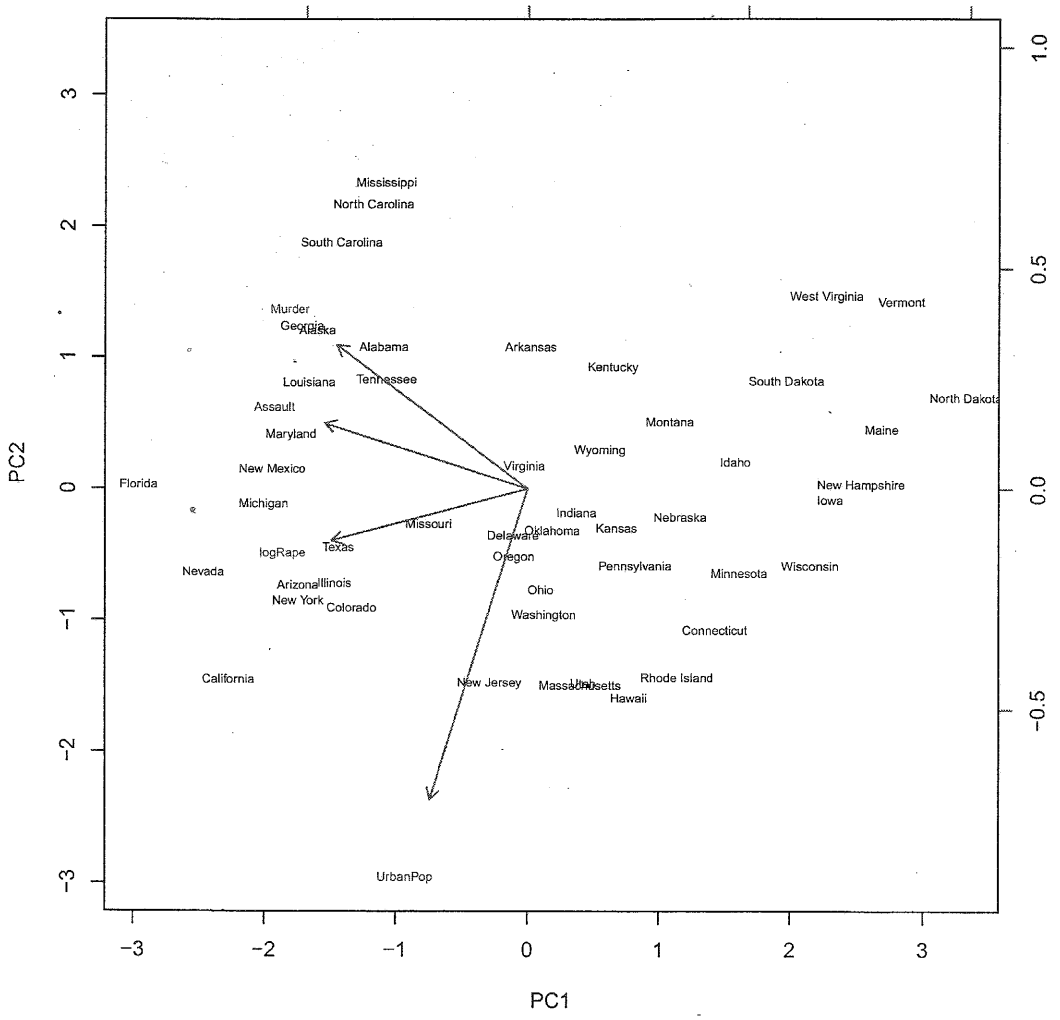


Figure 6.1.5: The biplot for the `USArrests` dataset with scaling.

```
# CHUNK 8
PCA.unscaled <- prcomp(USArrests, scale. = FALSE)
summary(PCA.unscaled)

PCA.unscaled$rotation
biplot(PCA.unscaled, scale = 0, cex = 0.6)
```

```
## Importance of components:
##                           PC1      PC2     PC3     PC4
## Standard deviation     83.4978 13.98288 2.52368 0.29143
## Proportion of Variance  0.9718  0.02725 0.00089 0.00001
## Cumulative Proportion   0.9718  0.99910 0.99999 1.00000
##                              PC1          PC2         PC3          PC4
## Murder   -0.041810210 0.047906448 -0.99725886 0.037837046
## Assault  -0.998053720 0.044131837  0.04402688 0.001669238
## UrbanPop -0.046117192 -0.997841072 -0.04561010 0.010301396
## logRape  -0.003725906 -0.008399323 -0.03815909 -0.999229430
```

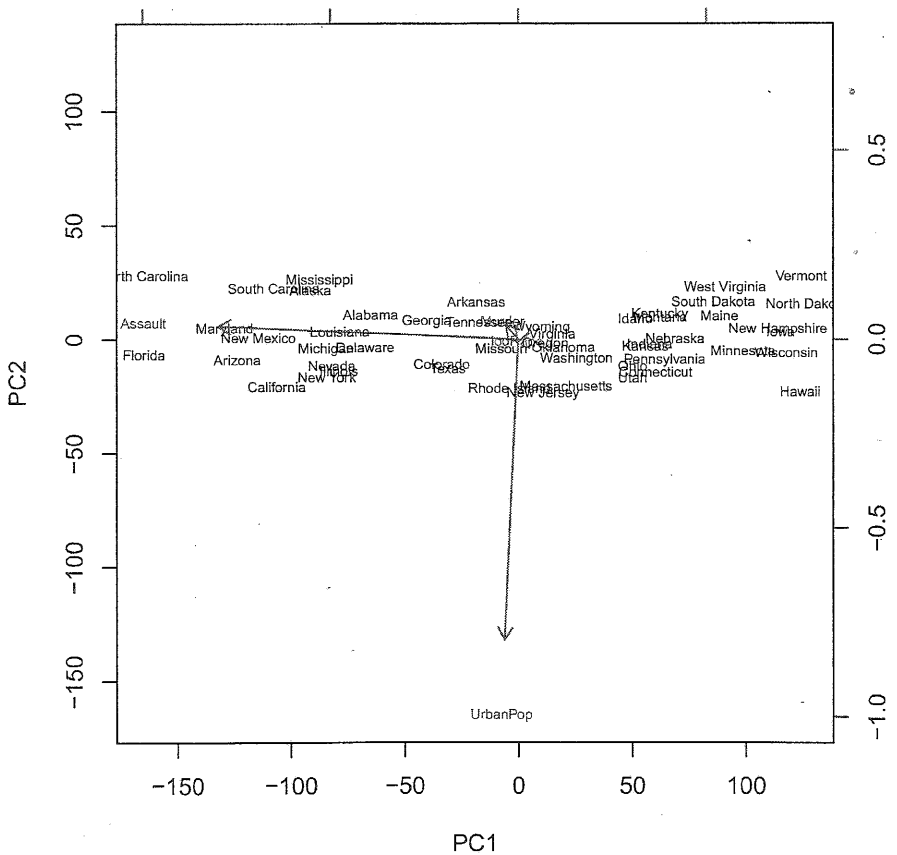


Figure 6.1.6: The biplot for the `USArrests` dataset without scaling.

There are three different ways, suggested by different sources, to combine the crime-related variables.

- *Method 1:* The most straightforward method is to simply take the scores of the first PC as a new feature. This is easily achieved in CHUNK 9, where we extract the first column of the `x` matrix, insert it into the `USArrests` data as a new feature called `crime1`, and print out the first six rows of the new dataset.

```
# CHUNK 9
USArrests.1 <- USArrests  # make a new copy of USArrests
USArrests.1$crime1 <- PCA$x[, 1]
head(USArrests.1)

##            Murder Assault UrbanPop  logRape      crime1
## Alabama      13.2     236       58 3.054001 -1.08857173
## Alaska       10.0     263       48 3.795489 -1.59517817
## Arizona       8.1     294       80 3.433987 -1.74286818
## Arkansas      8.8     190       50 2.970414  0.02590682
## California    9.0     276       91 3.703768 -2.27067947
## Colorado      7.9     204       78 3.655840 -1.33306093
```

- *Method 2 (Suggested in the June 2019 PA model solution):* When the dataset has a lot of variables (far more than 4), using the first PC entirely may result in a new feature that is not easy to interpret. In the case of the `USArrests` data, the first PC has `Murder`, `Assault`, and `logRape` as the dominating variables and is mainly a measure of overall crime rates, so we may simply drop `UrbanPop` and use the loadings of the three crime-related variables to define a new feature. In CHUNK 10, we manually create the new feature using the PC loadings of `Murder`, `Assault`, and `logRape`.

EXAM NOTE

The June 2019 PA exam model solution says that

> "[u]sing every number in the first principal component is not as beneficial as it will be difficult to explain to the client what it represents. By using only a few of the [variables] it is clearer what the new variable is measuring."

```
# CHUNK 10
USArrests.2 <- USArrests

# the scale() function will convert the USArrests data to a numeric matrix
# so we use the as.data.frame() function to change it back to a data frame
USArrests.scaled <- as.data.frame(scale(USArrests))

USArrests.2$crime2 <- PCA$rotation[1, 1] * USArrests.scaled$Murder +
```

```
  PCA$rotation[2, 1] * USArrests.scaled$Assault +
  PCA$rotation[4, 1] * USArrests.scaled$logRape

# OR
#USArrests$crime2 <- PCA$rotation[1, 1] * scale(USArrests$Murder) +
#  PCA$rotation[2, 1] * scale(USArrests$Assault) +
#  PCA$rotation[4, 1] * scale(USArrests$logRape)

head(USArrests.2)

##            Murder Assault UrbanPop  logRape     crime2
## Alabama      13.2     236       58 3.054001 -1.2326877
## Alaska       10.0     263       48 3.795489 -1.9304293
## Arizona       8.1     294       80 3.433987 -1.4664867
## Arkansas      8.8     190       50 2.970414 -0.2711172
## California    9.0     276       91 3.703768 -1.7840493
## Colorado      7.9     204       78 3.655840 -1.0949065
```

Notice that it is necessary to scale the three crime-related variables before setting up the new feature. The scaling can be done by scaling the entire `USArrests` dataset (with the new data frame called `USArrests.scaled`) or scaling each of the three crime-related variables separately using the `scale()` function.

- *Method 3 (Suggested in Module 8 of PA e-learning modules):* The third method, which is illustrated in the sample project section of Module 8 of the PA e-learning modules, is to run a PCA on only the three crime-related variables and use the first resulting PC as the new feature. By design, the first PC found this way will only involve the three crime-related variables, which are our interest, but not `UrbanPop`. This is done in CHUNK 11.

```
# CHUNK 11
USArrests.3 <- USArrests

# Run a PCA on only the three crime-related variables
PCA.3 <- prcomp(USArrests.3[, c(1, 2, 4)], center = TRUE, scale. = TRUE)
PCA.3$rotation

##                PC1        PC2        PC3
## Murder  -0.5839540  0.4974194 -0.6415385
## Assault -0.5970524  0.2722855  0.7545787
## logRape -0.5500237 -0.8236714 -0.1379833

USArrests.3$crime3 <- PCA.3$x[, 1]
head(USArrests.3)

##            Murder Assault UrbanPop  logRape     crime3
## Alabama      13.2     236       58 3.054001 -1.3085142
```

```
## Alaska       10.0      263       48 3.795489 -1.9744320
## Arizona       8.1      294       80 3.433987 -1.5022655
## Arkansas      8.8      190       50 2.970414 -0.2874132
## California    9.0      276       91 3.703768 -1.8219787
## Colorado      7.9      204       78 3.655840 -1.1004001
```

Comparing the results in CHUNKs 9, 10, and 11, we can see that the three new features possess approximately the same values. My personal recommendation is to follow the June 2019 PA exam model solution and use the second method (it was suggested by a real exam!!), although any of the three methods is acceptable so long as you provide your rationale.

Deleting the original variables. Irrespective of which feature generation method you adopt, it is of utmost importance to drop the variables that contribute to the new feature. Failure to do so will result in a duplication of information in the data and raise the issue of multicollinearity when a GLM is fitted. Using the second feature generation method as an example, we can delete the three crime-related variables in CHUNK 12.

```
# CHUNK 12
USArrests.2$Murder <- NULL
USArrests.2$Assault <- NULL
USArrests.2$logRape <- NULL

# OR
# USArrests.2[, c(1, 2, 4)] <- NULL

head(USArrests.2)

##            UrbanPop     crime2
## Alabama          58 -1.2326877
## Alaska           48 -1.9304293
## Arizona          80 -1.4664867
## Arkansas         50 -0.2711172
## California       91 -1.7840493
## Colorado         78 -1.0949065
```

After the deletion, the `USArrests.2` dataset has only two features, `UrbanPop` and `crime2`, as desired.

EXAM NOTE

The June 2019 PA exam model solution says that

> "[m]any candidates did not drop the contributing variables to the new feature, leading to a rank-deficient model in [a later task]."