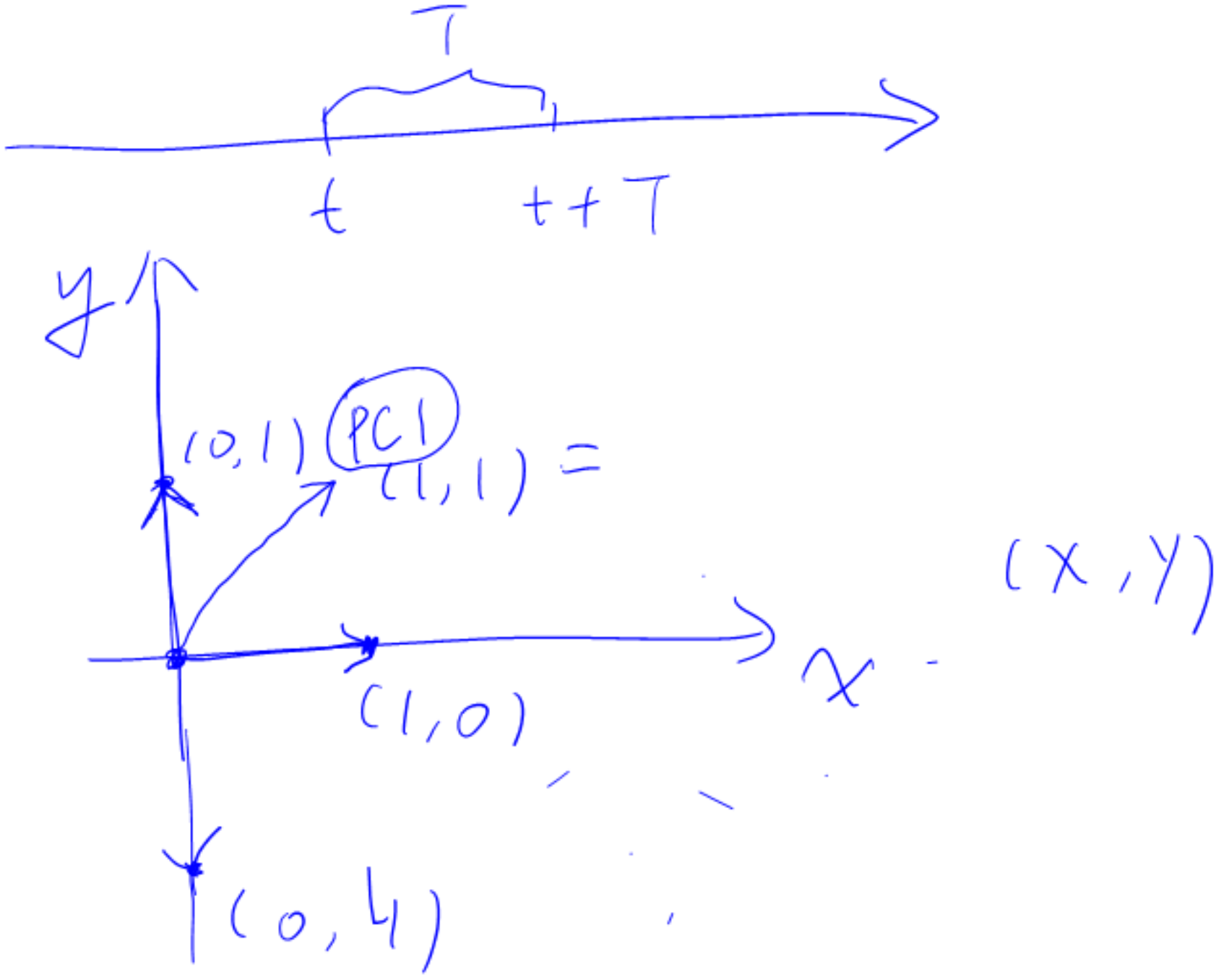

T
t
t + T
y
(0, 1)
PC1
(1, 1) =
(x, y)
x
(1, 0)
(0, 4)

Unsupervised Learning. Principle Component Analysis. (PCA)

W. Wang[1]

[1]Department of Mathematics
University of Houston

MATH 4323

Supervised Learning.

So far we've been focusing on supervised learning methods, such as K-Nearest Neighbors (KNN) and Support Vector Machines (SVM).

In supervised learning setting, we have access to both

- a set of p features $X_1, X_2, \ldots, X_p$, measured on n observations,

  and, **more importantly**,

- a response Y also measured on those same n observations.

Goal: predict/explain Y using $X_1, X_2, \ldots, X_p$.

Examples:

- Given the stock price changes in the previous 5 days, predict the price change today.
- Given the patient characteristics, predict the heart disease status.

Unsupervised Learning.

In this part of the course, we will instead focus on **unsupervised learning**, where we only have

- a set of p features $X_1, X_2, \ldots, X_p$ measured on n observations
- there is no designated response variable Y.

Goal: instead of prediction, we focus on discovering patterns & relationships among measurements $X_1, \ldots, X_p$ and among the n observations, e.g. can we discover groupings among the variables or among the observations?

Examples:

- Marketing: given demographic & spending information (predictors) on customers, proceed to group them by similarity.
- Gene Expression data: given 6830 gene expression measurements (predictors) for 64 cell lines, proceed to group the cell lines by similarity of their gene expressions.

The Challenge of Unsupervised Learning.

Supervised learning is a pretty well-understood area, due to having a

- well-developed set of tools (linear/logistic regression, KNN, SVM, and many more)
- a target response variable, that helps assessing the quality of your method via model validation approaches (validation set, cross-validation, etc).

The Challenge of Unsupervised Learning.(Continued)

In contrast, **unsupervised** learning is much more subjective:

- there's no simple goal for the analysis, such as predicting a response,
- it is often performed as part of an exploratory data analysis,
- there's no easy way to assess the quality of the solution; no universally accepted mechanism for performing cross-validation or validating results on an independent data set. (Simply because the problem is unsupervised)

PCA and Clustering.

Two of the most popular unsupervised learning techniques are

- **Principal Component Analysis (PCA)** - a tool for grouping the predictor variables $X_1, \ldots, X_p$, that's used for data visualization or data pre-processing before supervised techniques are applied.

- **Clustering** - a broad class of methods for grouping the observations, leading to discovery of unknown subgroups in data.

Usefulness of both methods - PCA & clustering - was on display for the gene expression data example in the early lectures (see next two slides).

PCA & Clustering: Gene Expression example.

Example. Presume that for our gene expression data we have

- $n = 64$ cell lines (observations),
- $p = 6830$ gene expression measurements (variables)

When we just had two variables - X_1 and X_2 - it was easy to plot the observations.

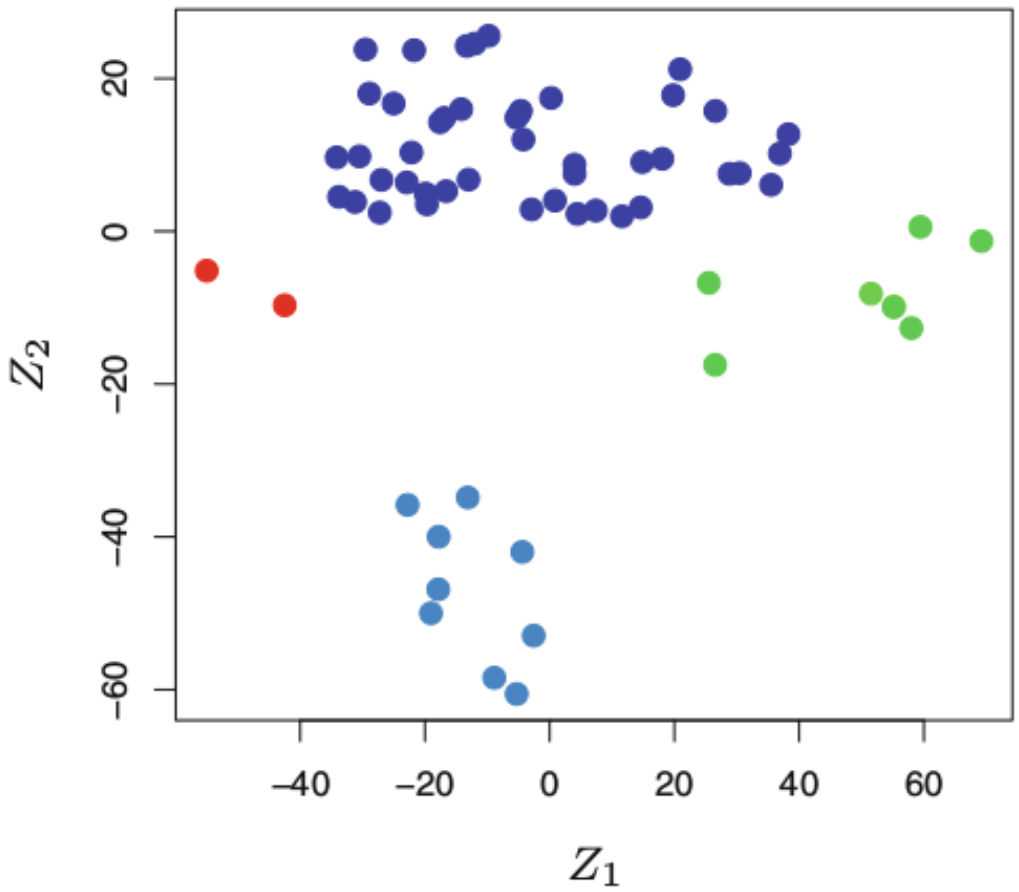
With $p = 6830$, it is virtually impossible to visualize the observations.

This issue can be addressed via **principal component analysis (PCA)**, which

- shrinks the total number p of variables down to a set of **few** principal components (PCs),
- with those PCs retaining as much information from all p variables as possible (with minimal loss of information).

PCA & Clustering: Gene Expression example.

Example (cont'd). Using just the first two PCs, Z_1 and Z_2 (which amounts to a considerable loss of information, but still):



Afterwards, we proceed to **cluster** the patients with respect to Z_1 and Z_2. Deciding on the number K of clusters is often a difficult problem, which we will address later. But here we suggest $K = 4$ clusters, each marked with separate color.

Now we could examine each cluster for types of cancer and their relationship to gene expression levels.

Motivation: Principle Component Analysis (PCA)

- Suppose we wish to visualize n observations on p features, $X_1, X_2, \cdots, X_p$. How would we do it?

  We could create two-dimensional scatterplots of the data, each of which contains the n observations' measurements on two of the features.

- Limitation:
  - There will be $\binom{p}{2}$ such scatterplots. If p is large, then it will certainly not be possible to look at all of them.
  - Moreover, most of the scatterplots will not be informative since they each contain just a small fraction of the total information present in the dataset.
  - In addition, with large p, the dispersion(covariance) matrix will be too large to study and interpret properly (too many pairwise correlation between variables to investigate).
- Clearly, a better method is required to visualize/interpret the n observations when p is large.

Principle Component Analysis (PCA)

PCA provides a tool:

- PCA produces a low-dimensional representation of a dataset. It finds a sequence of **linear combinations of the variables** that contains as much as possible of the variation.
- Each **linear combination** will correspond to a **principle components**.
- If we can obtain a low (two)-dimensional representation of the data that captures most of the information, then we can plot the observations in this low-dimensional space.
- Now, we explain the manner in which these dimensions, or principle components, are found.

Principle Component Analysis

In this section, we explain the manner in which these dimensions, or principle components, are found.

- How to perform principle component analysis
- How to decide how many principal components are needed
- How to do interpretation based on principle component scores
- Determine when a principal component analysis should be based on the variance-covariance matrix or the correlation matrix

Principle Component Analysis: linear combination

- The first principle component is the **normalized linear combination** of the features

$$Z_1 = \phi_{11} X_1 + \phi_{21} X_2 + \cdots + \phi_{p1} X_p$$

  that has the largest variance, where $\sum_{j=1}^{p} \phi_{j1}^2 = 1$.
- The elements $\phi_{11}, \cdots \phi_{p1}$ are referred as to the loadings of the first principal components; the loadings make up the principle component loading vector, $\phi_1 = (\phi_{11}, \cdots, \phi_{p1})^T$.
- We constrain the loadings so that their sum of squares is equal to one, since otherwise setting these elements to arbitrarily large in absolute value could result in an arbitrarily large variance.

Computation of the First Principal Component

- Suppose we have a n X p data set $\mathbf{X}$. Assume that each of the variables in $\mathbf{X}$ has been centered to mean zero (that is , the column means of $\mathbf{X}$ are zero).
- We then look for the linear combination of the sample feature values of the form

$$z_{i1} = \phi_{11} x_{i1} + \phi_{21} x_{i2} + \cdots + \phi_{p1} x_{ip}$$

for $i = 1, \cdots, n$ that has largest sample variance, subject to the constraint that $\sum_{j=1}^{p} \phi_{j1}^2 = 1$.

- Since each of the $x_{.j}$ has mean zero, then so does z_{i1}. Hence the variance of the z_{i1} can be written as

$$\frac{1}{n} \sum_{i=1}^{n} z_{i1}^2.$$

Computation of the First Principle Component

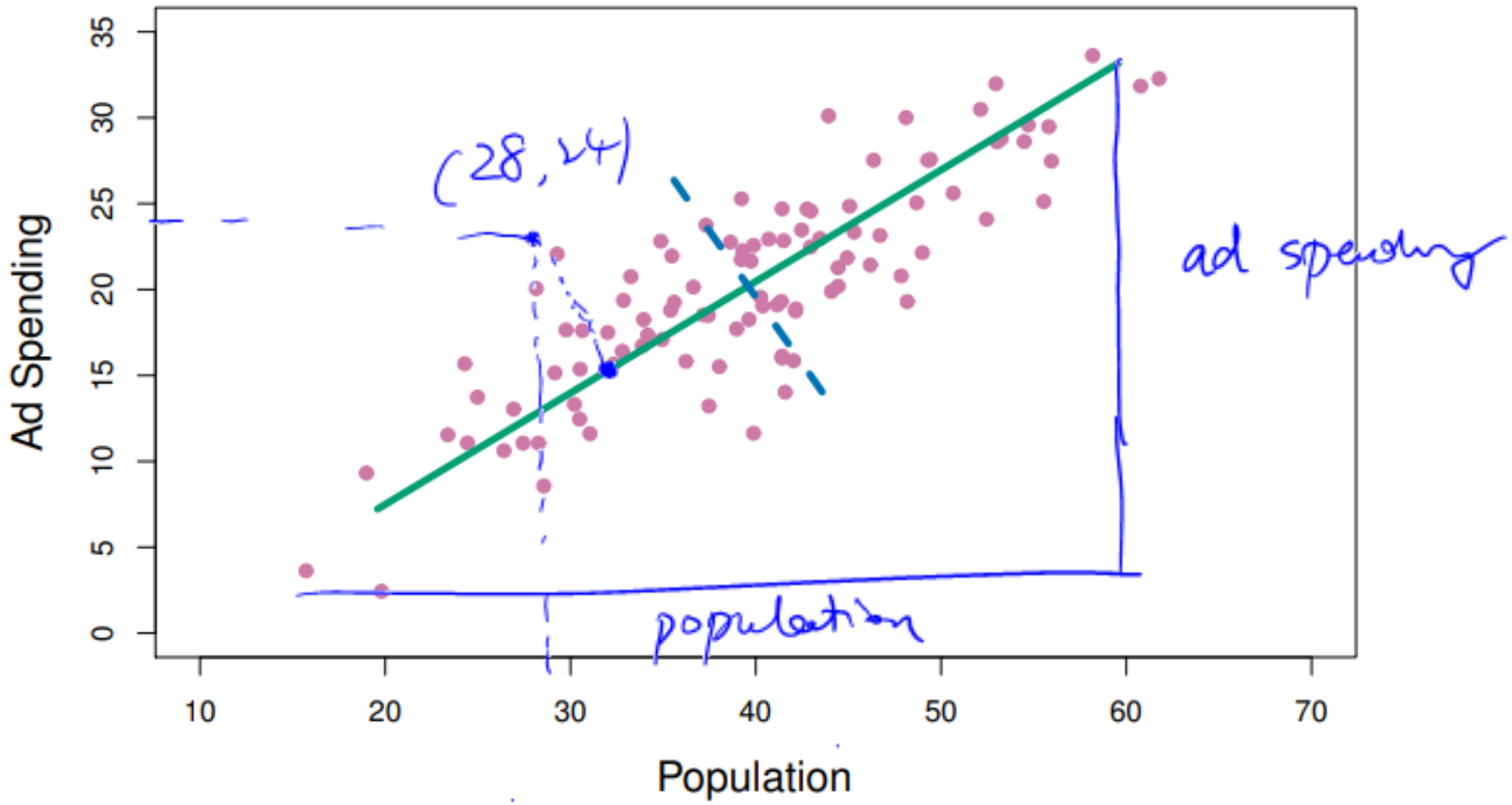
- In other words, the first principal component loading vector solves the optimization problem

$$\max_{\phi_{11},\ldots,\phi_{p1}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j1} x_{ij} \right)^2 \right\} \text{ subject to } \sum_{j=1}^{p} \phi_{j1}^2 = 1.$$

  This problem can be solved via a singular value decomposition of the matrix $\mathbf{X}$, a standard technique in linear algebra.

- We refer to $z_{11}, \cdots, z_{n1}$ as the scores of the first principle component Z_1.
- The loading vector $\phi_1 = (\phi_{11}, \cdots, \phi_{p1})^T$ defines a direction in feature space along which the data vary the most.
- If we project the n data points $x_1, \cdots, x_n$ onto this direction, the projected values are the principle component scores $z_{11}, \cdots, z_{n1}$ themselves.

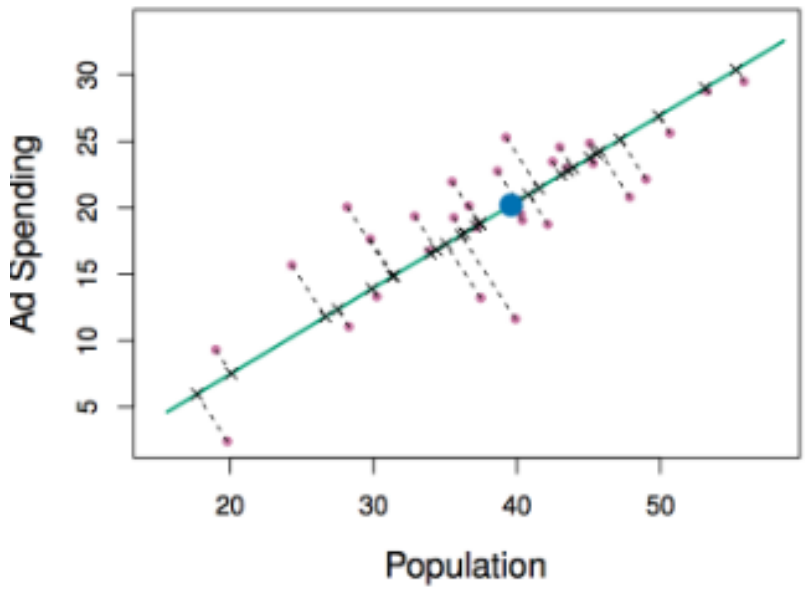
Example: PCA

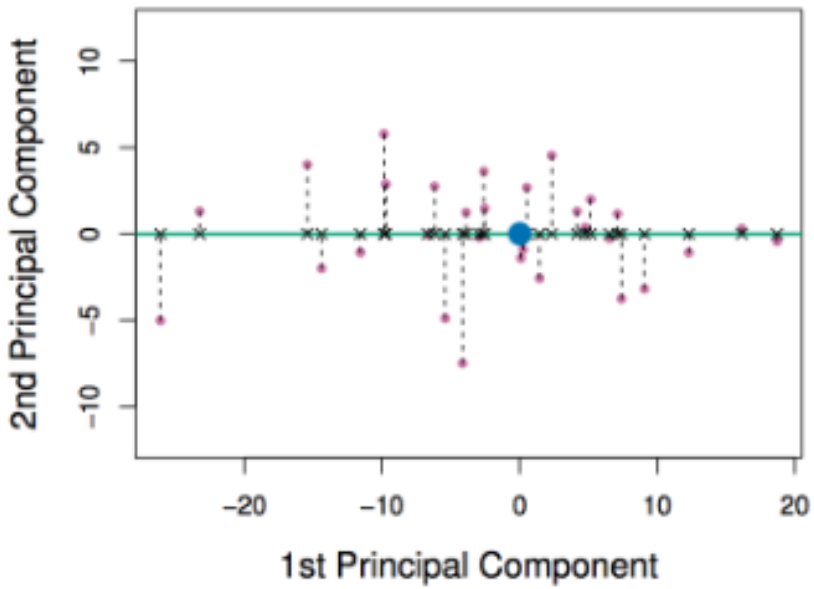


The population size (pop) and ad spending (ad) for 100 different cities are shown as purple circles. The green solid line indicates the first principal component, and the blue dashed line indicates the second principal component.

Example: Projection

The first principle component direction of the data is that along which the observations vary the most. For instance, we can see by eye that the direction of the green line is the direction along which there is the greatest variability in the data. If we projected the 100 observations onto the green line, the resulting projected observations would have the largest possible variance. (Projecting a point onto a line simply involves finding the location on the line which is closest to the point.)





Further Principle Components

- The second principle component is the linear comination of $X_1, X_2, \cdots, X_p$ that has maximal variance among all linear combinations that are **uncorrelated** with Z_1.
- The second principle component scores take the form

$$Z_{i2} = \phi_{12} x_{i1} + \phi_{22} x_{i2} + \cdots + \phi_{p2} x_{ip}$$

  where $\phi_2 = (\phi_{12}, \cdots, \phi_{p2})^T$ is the second principal component loading vector.
- Constraining Z_2 to be uncorrelated with Z_1 is equivalent to constraining the direction ϕ_2 to be orthogonal (perpendicular) to the direction ϕ_1.
- In total, there are $min(n-1, p)$ principle components.

PCA procedure in matrix form

Suppose we have a random vector X

$$\mathbf{X} = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{pmatrix}$$

with population variance-covariance matrix

$$var(\mathbf{X}) = \Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1p} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2p} \\ \vdots & \vdots & & \vdots \\ \sigma_{p1} & \sigma_{p2} & \cdots & \sigma_p^2 \end{pmatrix}$$

PCA procedure

Consider the linear combinations

$$\begin{cases} Z_1 = \phi_{11}X_1 + \phi_{12}X_2 + \cdots + \phi_{1p}X_p \\ Z_2 = \phi_{21}X_1 + \phi_{22}X_2 + \cdots + \phi_{2p}X_p \\ \vdots \\ Z_p = \phi_{p1}X_1 + \phi_{p2}X_2 + \cdots + \phi_{pp}X_p \end{cases}$$

Each of the above equation can be considered as a linear regression: Using p predictors, $X_1, X_2, \cdots, X_p$ to predict Z_i, but with no intercepts.

- Z_i is a function of random vector, so it is also a random variable.
- Denote $\phi_i = (\phi_{i1}, \phi_{i2}, \cdots, \phi_{ip})$.
- The variance of Z_i is $\phi_i^T \Sigma \phi_i$.
- The covariance between Z_i and Z_j is $\phi_i^T \Sigma \phi_j$.

PCA procedure

- The first principal component is the linear combination of X variables that has maximum variance among all linear combinations.
- The variation in the first PC accounts for as much variation in the data as possible.
- Formally speaking, we would like to select a set of $\phi_{11}, \phi_{12}, \cdots, \phi_{1p}$ that maximizes

$$var(Z_1) = \phi_1^T \Sigma \phi_1$$

- To obtain a unique answer to the above optimization task, we need to add the constraint that

$$\phi_1^T \phi_1 = \sum_{j=1}^{p} \phi_{1j}^2 = 1$$

PCA procedure

- The second principal component is the linear combination of X variables that accounts for as much of the remaining variation as possible, with the constraint that the correlation between the first component (PC1) and the second component (PC2) is 0.
- Formally speaking, we would like to select a set of $\phi_{21}, \phi_{22}, \cdots, \phi_{2p}$ that maximizes

$$var(Z_2) = \phi_2^T \Sigma \phi_2$$

- subject to:

$$\phi_2^T \phi_2 = \sum_{j=1}^{p} \phi_{2j}^2 = 1$$

$$cov(Z_1, Z_2) = \phi_1^T \Sigma \phi_2 = 0$$

PCA procedure

- All subsequent principal components have the same property: They are linear combinations that account for as much of the remaining variation as possible and they are not correlated with other principal components.

How to find the coefficients ϕ_{ij} for a principal component?

(For interested students only) The solution involves the eigenvalues and eigenvectors of the variance-covariance matrix Σ.

- Let $\lambda_1 \geq \lambda_2 \geq, \cdots, \geq \lambda_p$ be the eigen values of Σ.
- Let $\phi_1, \phi_2, \cdots, \phi_p$ be the corresponding eigenvectors. The elements of the eigenvectors are the coefficients of the principal components.
- The variance of the ith principal component equals exactly to the ith eigenvalue.
- The ith principal component explains

$$\frac{\lambda_i}{\lambda_1 + \lambda_2 + \cdots + \lambda_p}$$

proportion of the total variation.

λ_1 : variance of PC1
λ_2 : variance of PC2
λ_p : Var of PC_p

USArrest example

Let us perform PCA on the USArrests dataset, from base R package. The rows of the data set contain the 50 states, in alphabetical order.

```
> states=row.names(USArrests)
> states
```

The columns of the data set contain the four variables.

```
> names(USArrests)
[1] "Murder"   "Assault"  "UrbanPop" "Rape"
```

Grab some summary statistics first and notice that the four variables have quite different means and variances.

```
> apply(USArrests,2,mean) #  1 indicates rows
                          #  2 indicates columns
  Murder   Assault UrbanPop     Rape
   7.788  170.760   65.540   21.232
```

USArrest example

```
> apply(USArrests,2,var)
    Murder    Assault   UrbanPop       Rape
  18.97047 6945.16571  209.51878   87.72916
```

If we don't scale the variables before performing PCA, then most of the principal components that we observed would be driven by the Assault variable, since it has the largest mean and variance.

So, standardize your variables first to have mean zero and standard deviation one before performing PCA.

There are several functions in R to perform PCA.

```
> pr.out=prcomp(USArrests,scale=T)
```

USArrest example

Output of the prcomp() function:

```
> names(pr.out)
[1] "sdev"     "rotation" "center"   "scale"    "x"
```

The center and scale components correspond to the means and standard deviations of the variables that were used for scaling prior to PCA (prcomp() centers the variables to have mean zero by default).

```
> pr.out$center # mean of each column/variable
  Murder  Assault UrbanPop     Rape
   7.788  170.760   65.540   21.232
> pr.out$scale # standard deviation of each variable
   Murder   Assault  UrbanPop      Rape
 4.355510 83.337661 14.474763  9.366385
```

USArrest example

Output of the prcomp() function:

The rotation matrix provides the principal component loadings; each column of the matrix contains the corresponding principal compoment loading vector.

```
> pr.out$rotation # loadings
                PC1        PC2        PC3         PC4
Murder   -0.5358995  0.4181809 -0.3412327  0.64922780
Assault  -0.5831836  0.1879856 -0.2681484 -0.74340748
UrbanPop -0.2781909 -0.8728062 -0.3780158  0.13387773
Rape     -0.5434321 -0.1673186  0.8177779  0.08902432
```

- How many distinct principle components here? Is this expected?

USArrest example

Output of the prcomp() function:

```
> pr.out$x
```

gives us the principal component score vectors. The standard deviation of each column of this matrix is the standard deviation of each principal component.

```
> apply(pr.out$x,2,sd) # standard deviation of each pc
      PC1       PC2       PC3       PC4
1.5748783 0.9948694 0.5971291 0.4164494
> pr.out$sdev # standard deviation of each pc
[1] 1.5748783 0.9948694 0.5971291 0.4164494
```

USArrest example

Output of the prcomp() function:

Question: How much variation is explained by the first component? How about the second and the third?

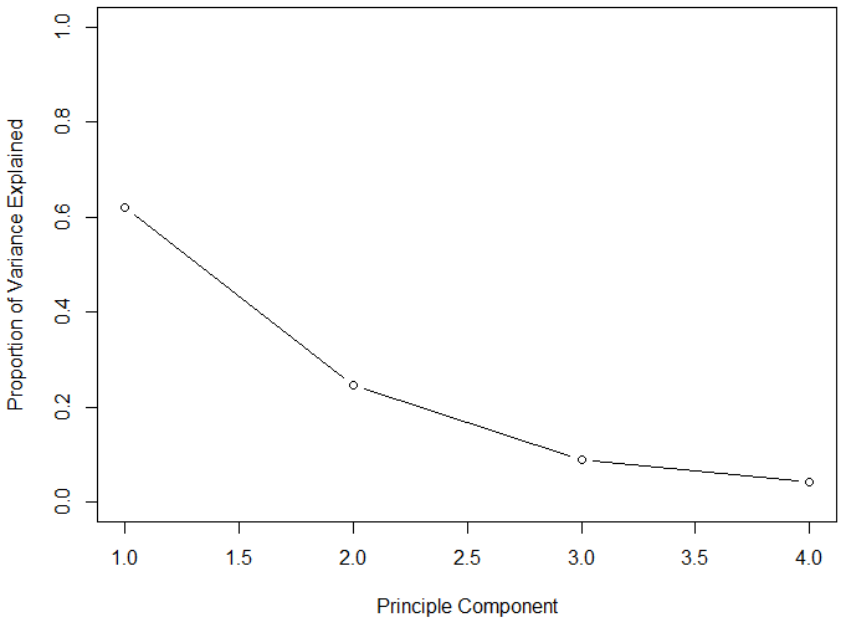
```
> pr.var=pr.out$sdev^2
> pr.var # variance explained by each pc:
         # decreasing order !!
[1] 2.4802416 0.9897652 0.3565632 0.1734301

> pr.var/sum(pr.var)
[1] 0.62006039 0.24744129 0.08914080 0.04335752
```

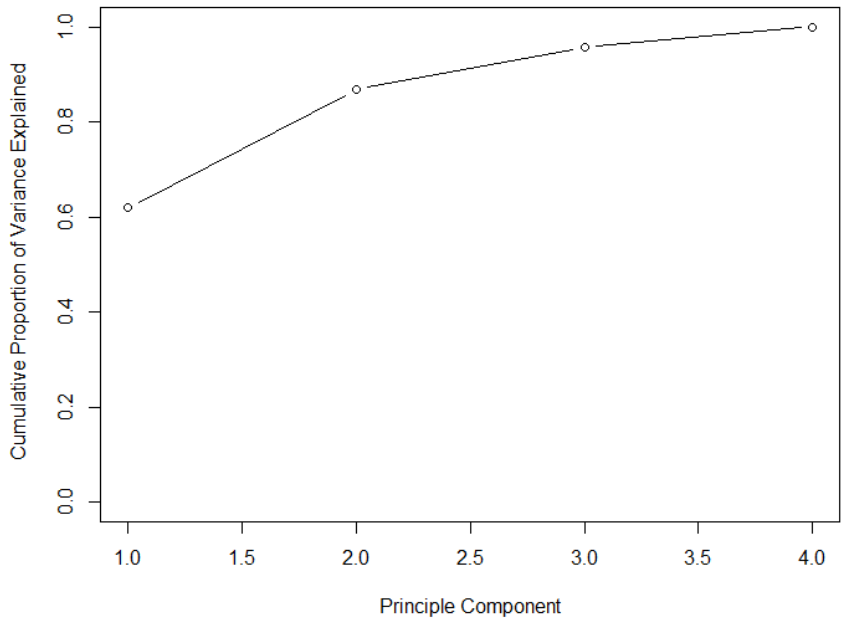
Answer: 62% of the variance is explained by the first principle component; 24.7% of the variance is explained by the second principle component, so forth.

USArrest example

Output of the prcomp() function:

Variance explained by the principle components:

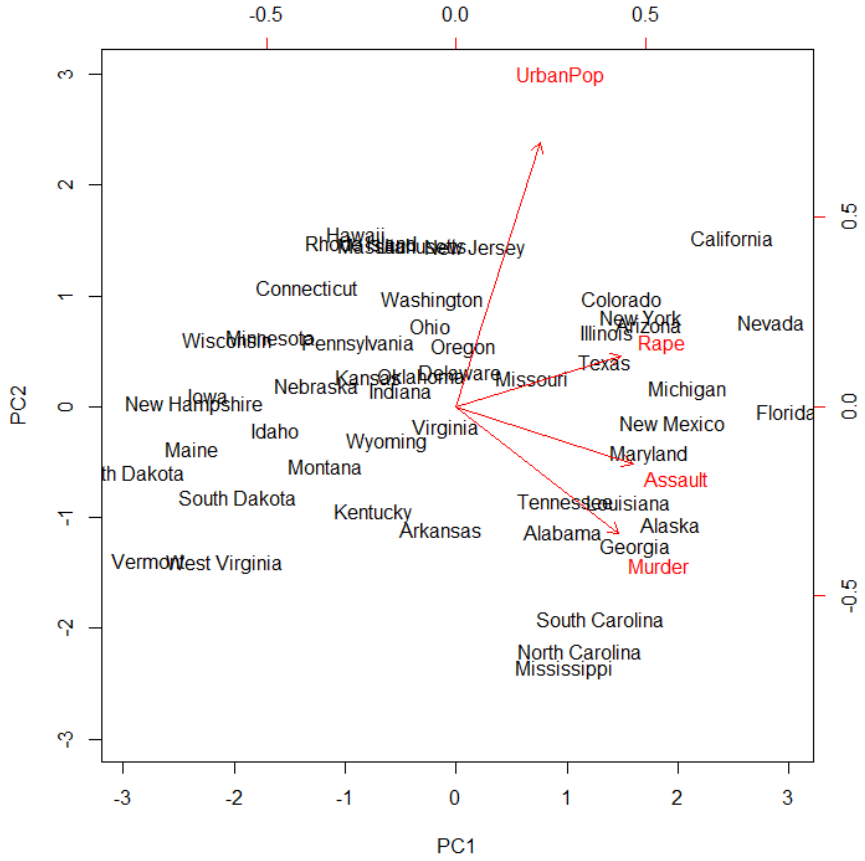




USArrest example

Output of the prcomp() function:

```
biplot(pr.out,scale = 0)
```



- Biplot plots the first two principal components of these data.
- The plot represents both the principal component scores and the loading vectors in a single display.
- Loadings can be found in slide 25.

USArrest example

- The state names represent the scores for the first two principal components. The red arrows indicate the first two principal component loading vectors.
- The crime-related variables (Murder, Assault, and Rape) are located close to each other. It is an indication that they are correlated with each other.
- The first loading vector places approximately equal weight on Assault, Murder, and Rape (Check R output), with much less weight on UrbanPop. Hence this component roughly corresponds to a measure of overall rates of serious crimes.
- The second loading vector places most of its weight on UrbanPop and much less weight on the other three features. Hence, this component roughly corresponds to the level of urbanization of the state. (California vs. Mississippi)

Other Interpretation of Principal Components

- The first principal component loading vector has a very special property:

  it defines the line in p-dimensional space that is closest to the n observations (using average squared Euclidean distance as a measure of closeness)
- The notion of principal components as the dimensions that are closest to the n observations extends beyond just the first principal component.
- For instance, the first two principal components of a data set span the plane that is closest to the n observations, in terms of average squared Euclidean distance.

Other Interpretation of Principal Components

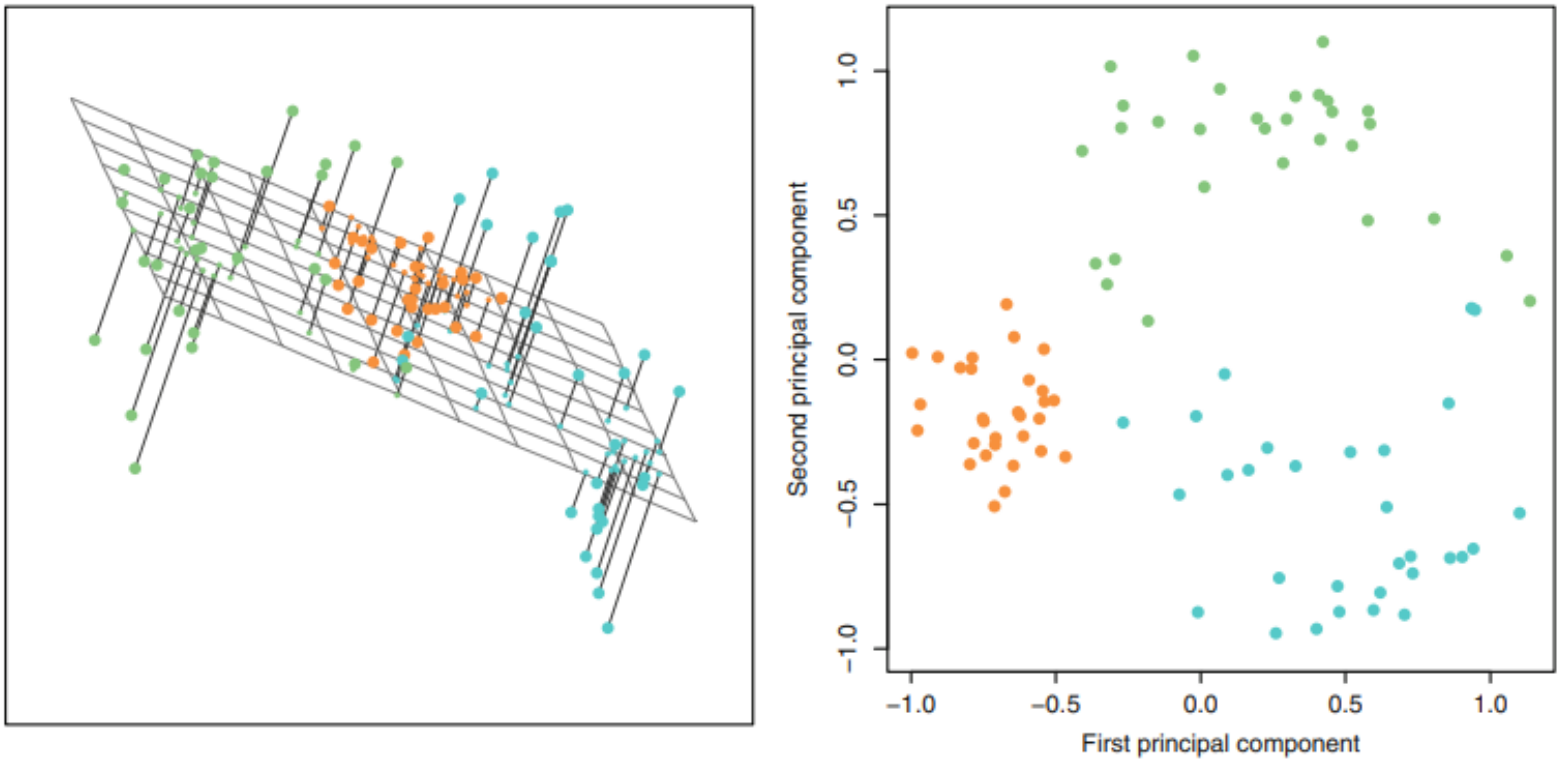


FIGURE 10.2. *Ninety observations simulated in three dimensions.* Left: *the first two principal component directions span the plane that best fits the data. It minimizes the sum of squared distances from each point to the plane.* Right: *the first two principal component score vectors give the coordinates of the projection of the 90 observations onto the plane. The variance in the plane is maximized.*

Scaling the Variables

- If the variables are in different units, scaling each to have standard deviation equal to one is recommended.
- If they are in the same units, you might or might not scale the variables.

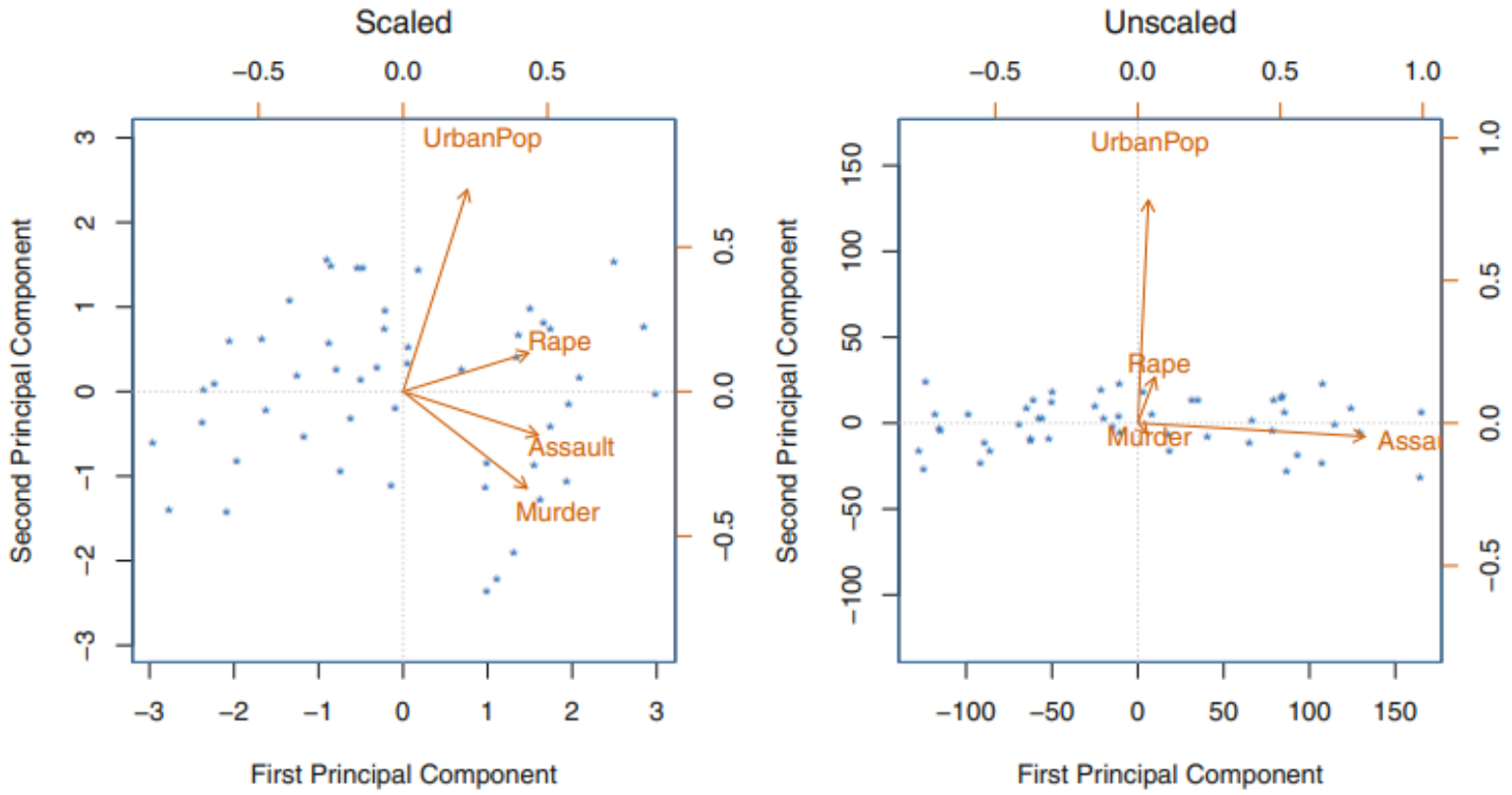
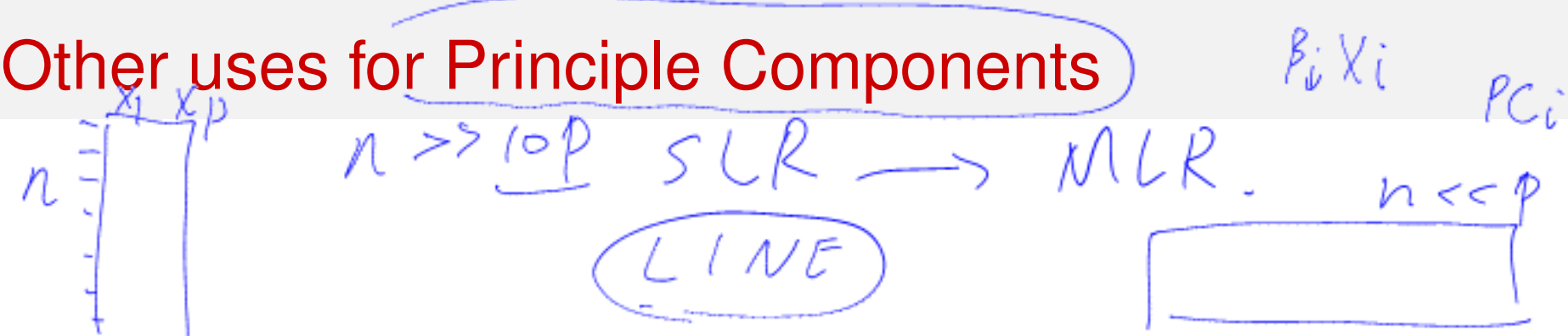


FIGURE 10.3. Two principal component biplots for the USArrests data. Left

How Many Principal Components to Use

- There is no single (or simple) answer to this question, as cross-validation is not available for this purpose. Why?
- The scree plot on the previous slide can be used as a guide: look for a point at which the proportion of variance explained by each subsequent principal component drops off. (The "elbow" in the screeplot)
- Unfortunately, there is no well-accepted objective way to decide how many principal components are enough.
- In practice, we tend to look at the first few principal components in order to find interesting patterns in the data.

Other uses for Principle Components



- We can perform regression using the principles component scores as features.
- In fact, many statistical techniques, such as regression, classification, and clustering, can be easily adapted to use the $n \times M$ matrix whose columns are the first M principal component score vectors, rather than using the full $n \times p$ data matrix.
- This can lead to less noisy results, since it is often the case that the signal (as opposed to the noise) in a dataset is concentrated in its first few principal components.