Data Mining
W4240 Sections 001, 003/004

Prof. Lauren Hannah

Columbia University, Department of Statistics

September 16, 2014

The Curse of Dimensionality

The Curse of Dimensionality: "There are multiple phenomena referred to by this name in domains such as numerical analysis, sampling, combinatorics, machine learning, data mining and databases. The common theme of these problems is that when the dimensionality increases, the volume of the space increases so fast that the available data becomes sparse. This sparsity is problematic for any method that requires statistical significance. In order to obtain a statistically sound and reliable result, the amount of data needed to support the result often grows exponentially with the dimensionality." (Wikipedia)

The Curse of Dimensionality

Problems with high dimensions:

- spurious correlation between dimensions and response (supervised)
- data are sparse in higher dimensions (need exponentially more data to make predictions)
- data can be hard to interpret/summarize (ex: sea surface temperatures, movie ratings)
- data can be hard to store if every entry is recorded (ex: movie ratings)

Dimensionality Reduction

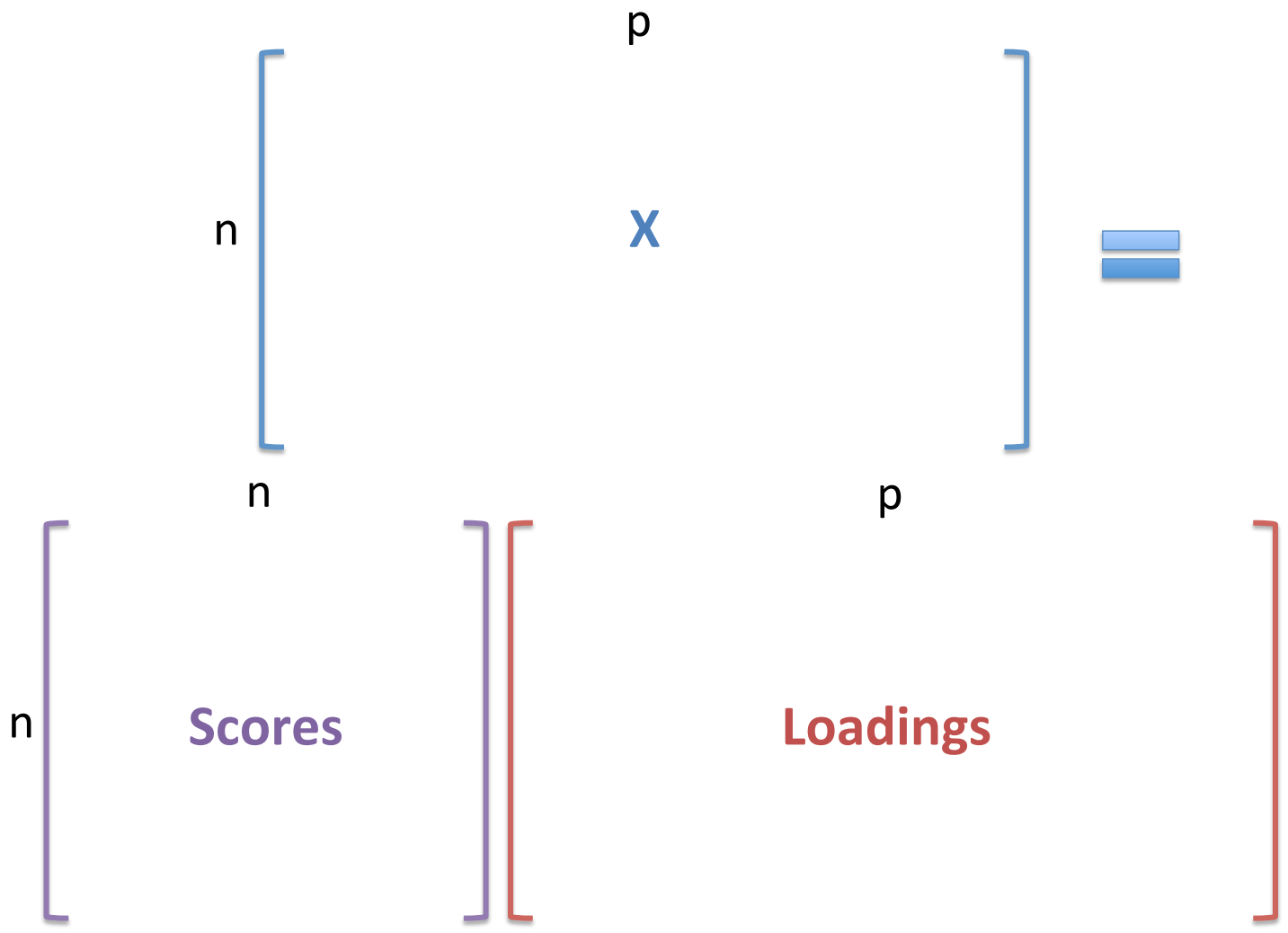
Linear feature extraction:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ \vdots \\ x_d \end{bmatrix} \rightarrow \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_K \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & \dots & \dots & \dots & w_{1d} \\ w_{21} & w_{22} & \dots & \dots & \dots & w_{2d} \\ \vdots & & & & & \vdots \\ w_{K1} & w_{K2} & \dots & \dots & \dots & w_{Kd} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ \vdots \\ x_d \end{bmatrix}$$

$$\mathbf{Y} = \mathbf{X}\mathbf{W}$$

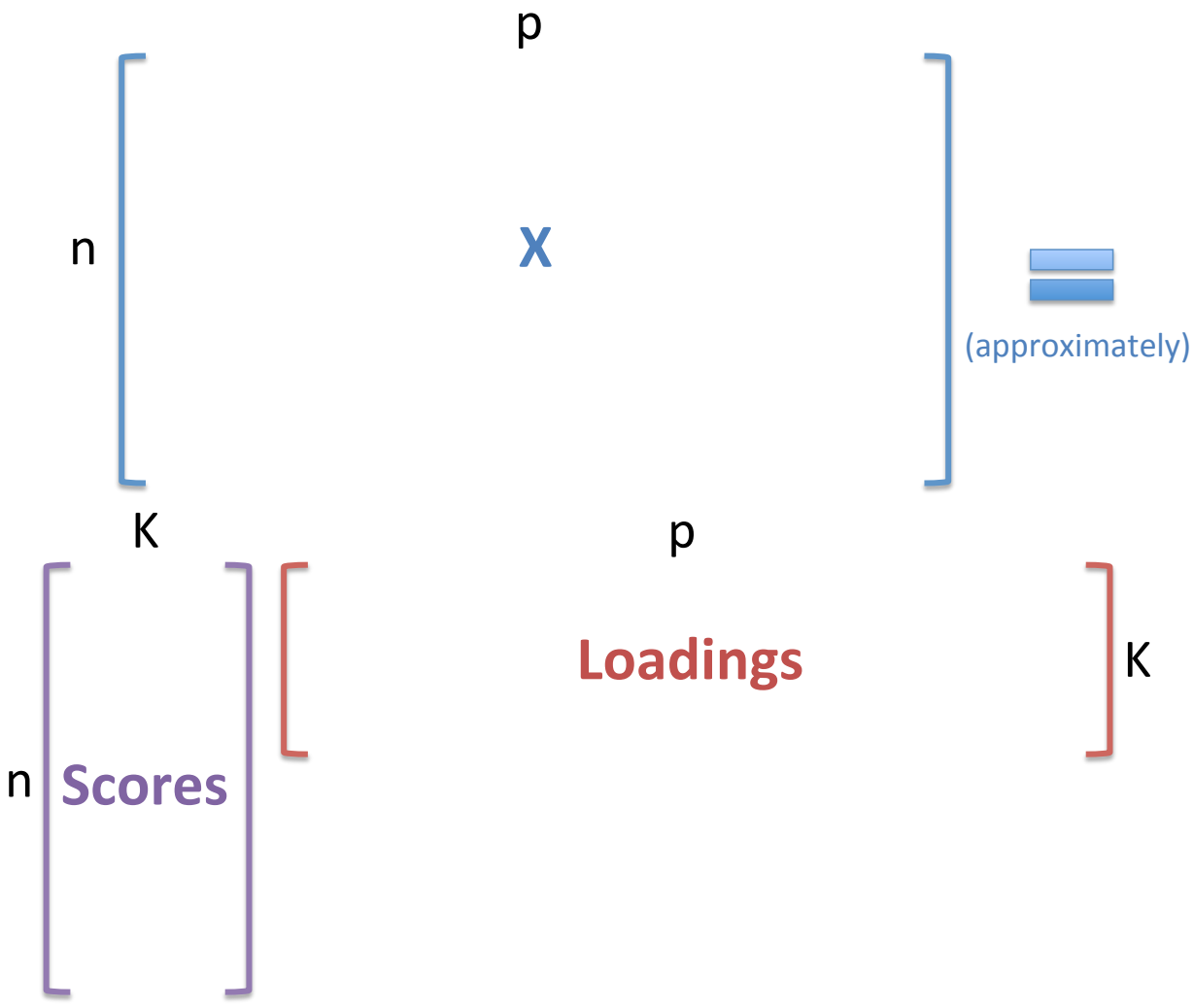
Dimensionality Reduction

Linear feature extraction can be viewed as a matrix factorization problem:

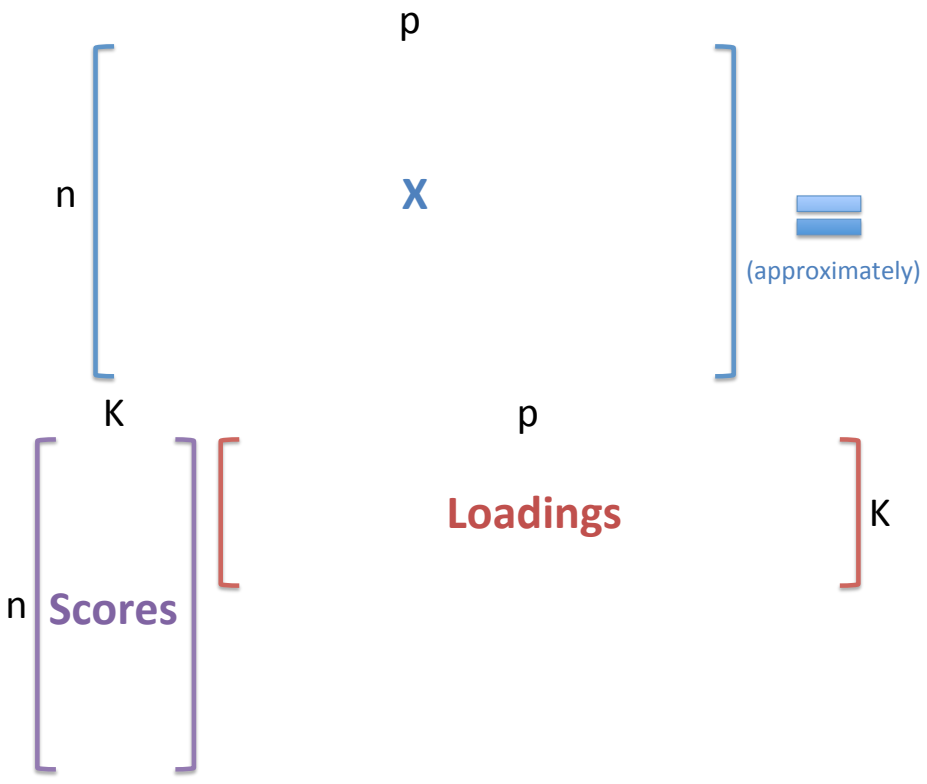


Dimensionality Reduction

Linear feature extraction can be viewed as a matrix factorization problem:

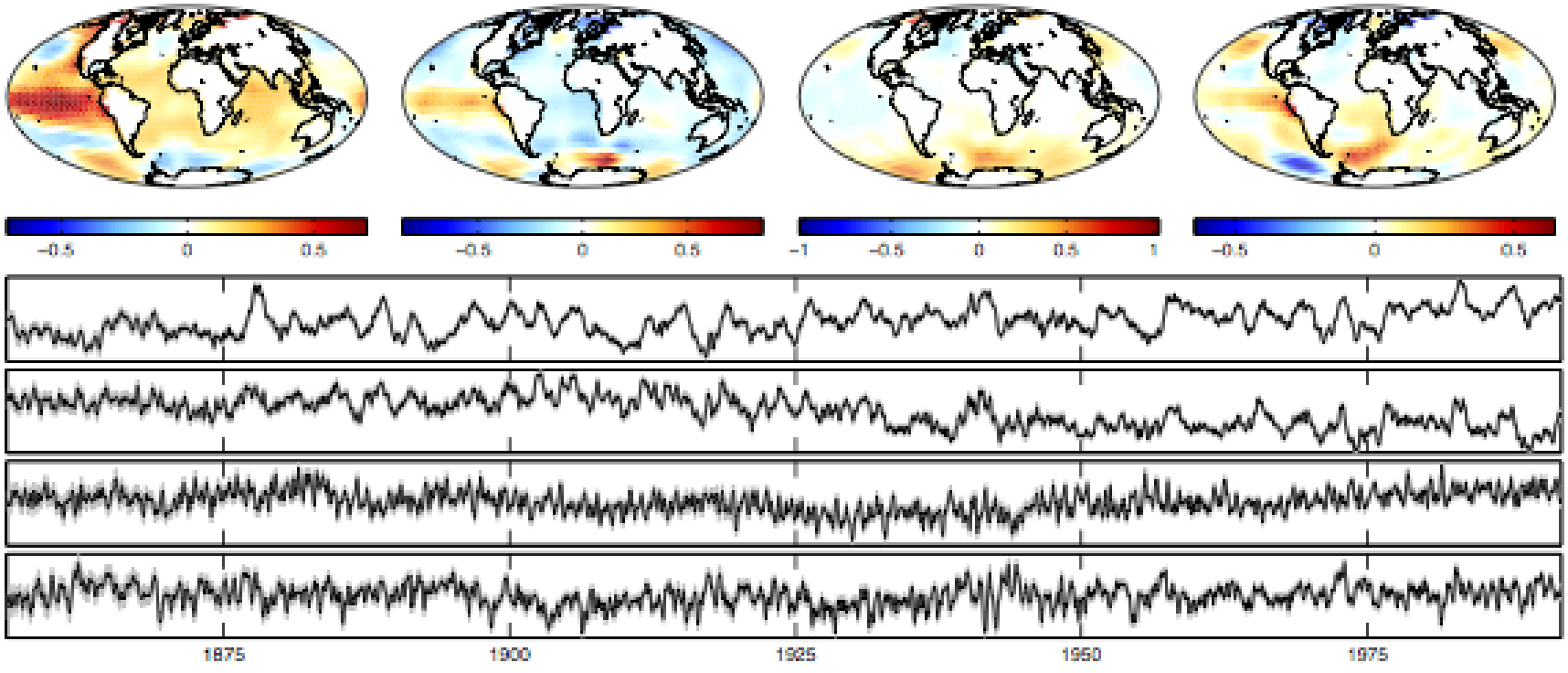


Dimensionality Reduction



- **Scores:** work in the score space for low dimensional approximately equivalent space (K dimensions instead original high dimensional data)
- **Loadings:** loadings are interpretable as building blocks for your dataset

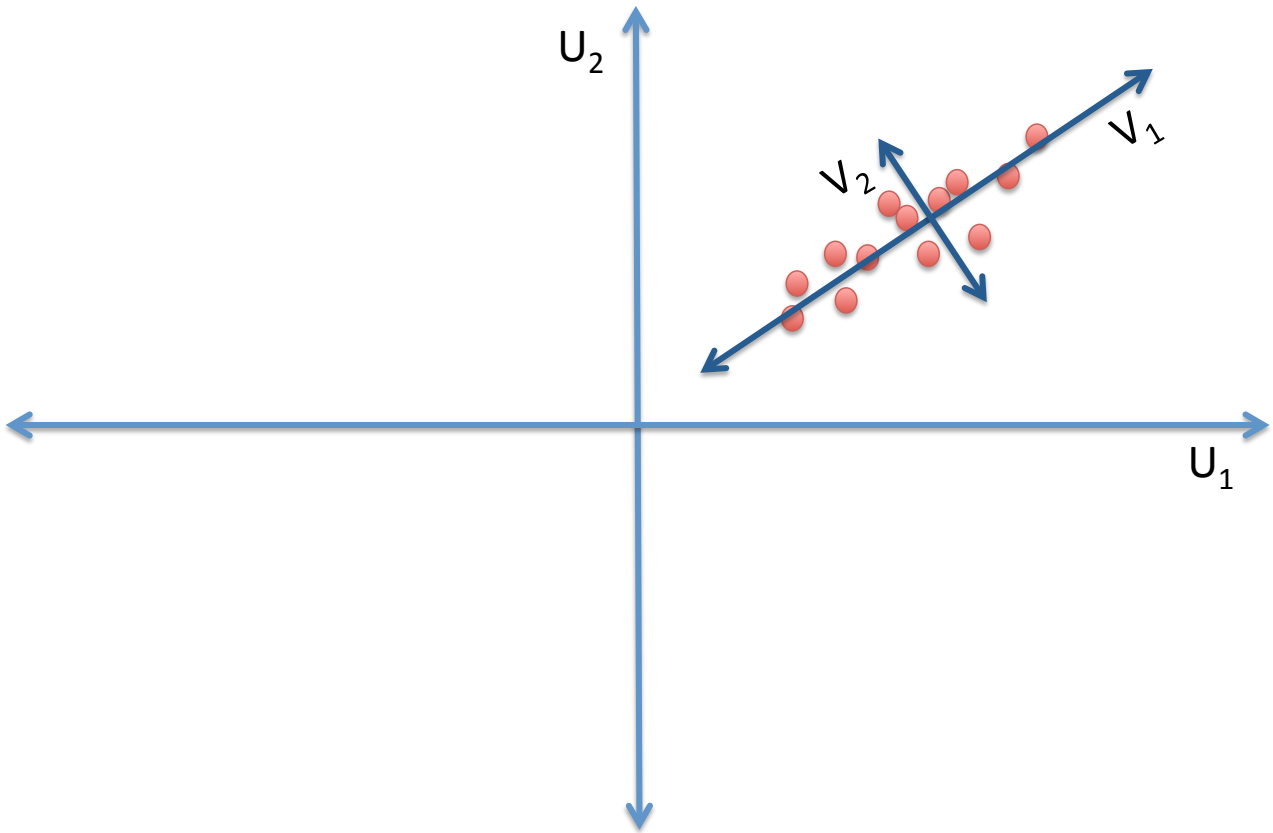
Dimensionality Reduction



From Luttinen and Ilin (2009)

Principal Components Analysis

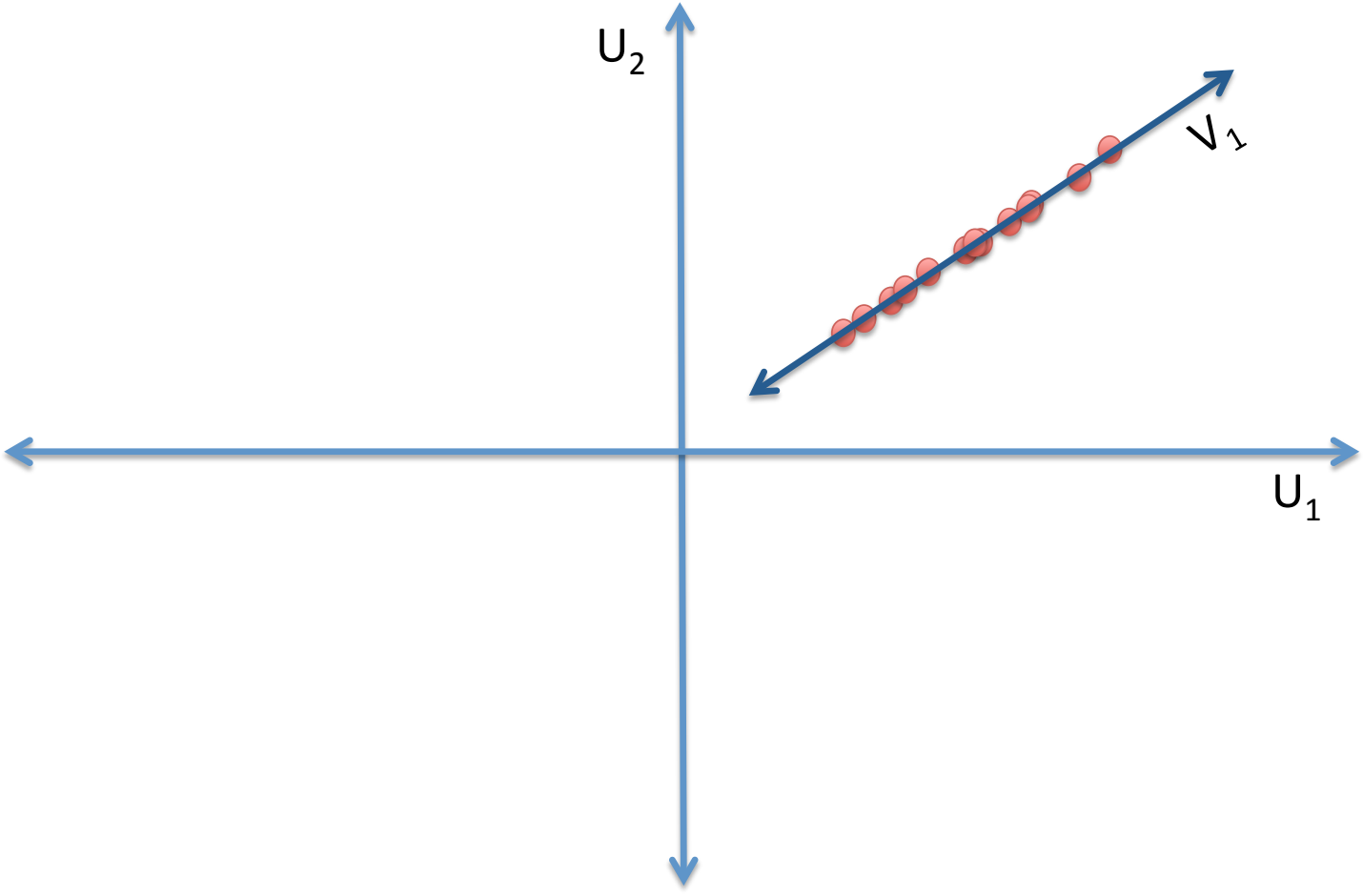
Basic idea: given dataset, want to find rotation (linear transformation) that best describes data



$y_1 = w_{11}x_1 + w_{12}x_2,\ y_2 = w_{21}x_1 + w_{22}x_2$

Principal Components Analysis

Basic idea: given linear transformation, we can throw out the less descriptive dimensions and still have a decent representation

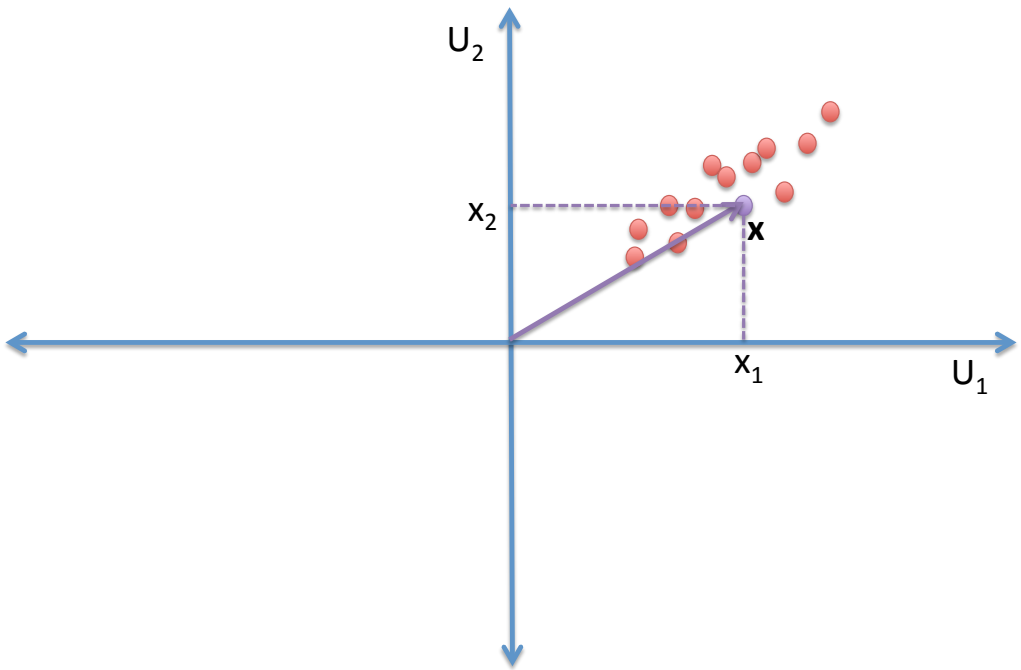


Principal Components Analysis

Mathematically, how do we do this?

Simple case: $\mathbf{x} \in R^2$, and we want a good projection $y \in R$

- make x_1, x_2 scalars
- make the axes vectors, $\mathbf{u}_1, \mathbf{u}_2$
- $\mathbf{x} = x_1\mathbf{u}_1 + x_2\mathbf{u}_2$

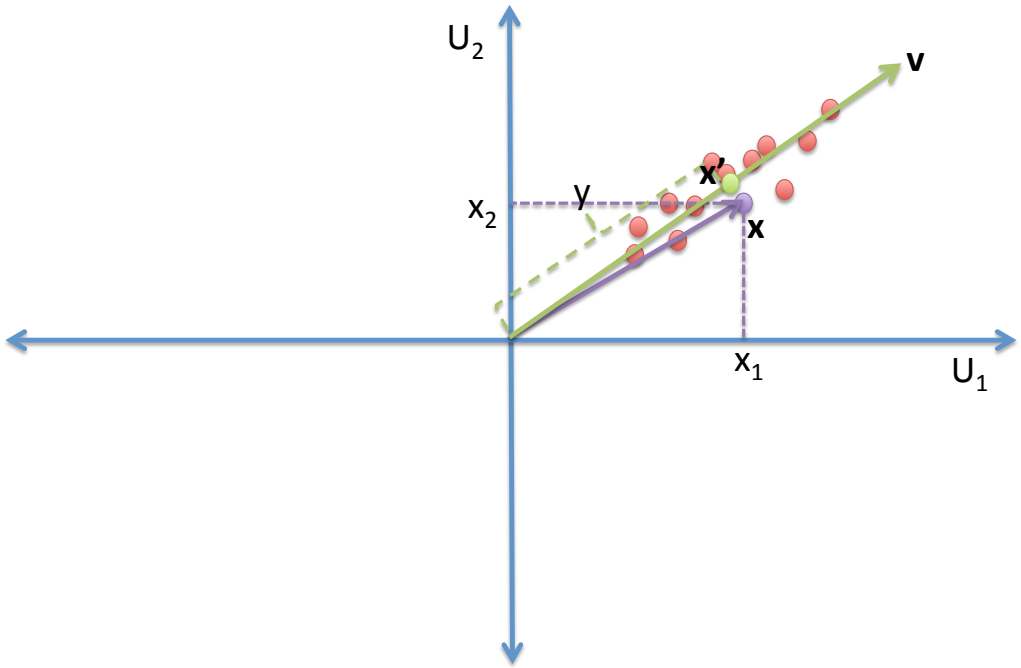


Principal Components Analysis

Mathematically, how do we do this?

Simple case: $\mathbf{x} \in R^2$, and we want a good projection $y \in R$

- now suppose we make a vector $\mathbf{v}$ (a feature)
- we can project $\mathbf{x}$ onto $\mathbf{v}$ to make $\mathbf{x}' = y\mathbf{v}$)
- two coordinates x_1, x_2 get turned into one, y



Principal Components Analysis

To find best $\mathbf{w}$:

- "closeness" is based on squared error between original points and new points:

$$\hat{\mathbf{w}}_1 = \arg\min_{\mathbf{w}_1} \left\{ \sum_{i=1}^{n} ||\mathbf{x}_i - \mathbf{x}'_i||_2^2 = \sum_{i=1}^{n} \sum_{j=1}^{d} \left(\mathbf{x}_{ij} - \mathbf{x}'_{ij}\right)^2 \right\}$$

- require that $\mathbf{w}_1$ has magnitude of 1:

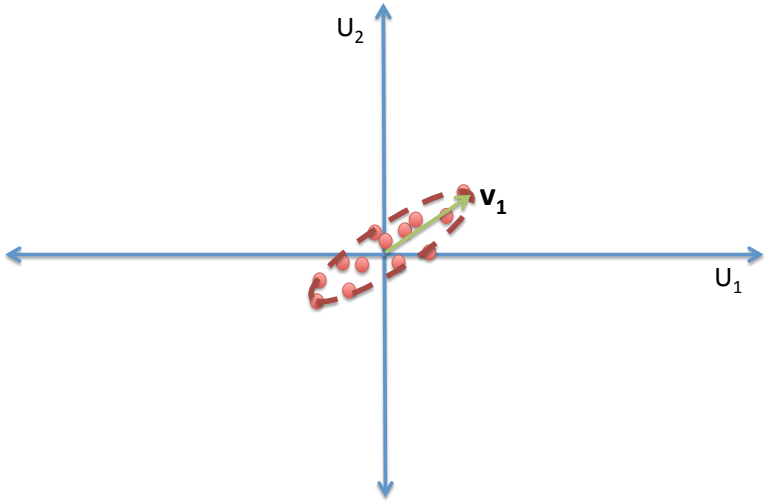
$$||\mathbf{w}_1||_2 = \sqrt{\sum_{j=1}^{d} w_{ij}^2} = 1$$

Principal Components Analysis

- objective is to minimize squared errors

$$\hat{\mathbf{w}}_1 = \arg \min_{\mathbf{w}_1 : ||\mathbf{w}_1||_2 = 1} \sum_{i=1}^{n} \sum_{j=1}^{d} \left(\mathbf{x}_{ij} - \mathbf{x}'_{ij} \right)_2^2$$

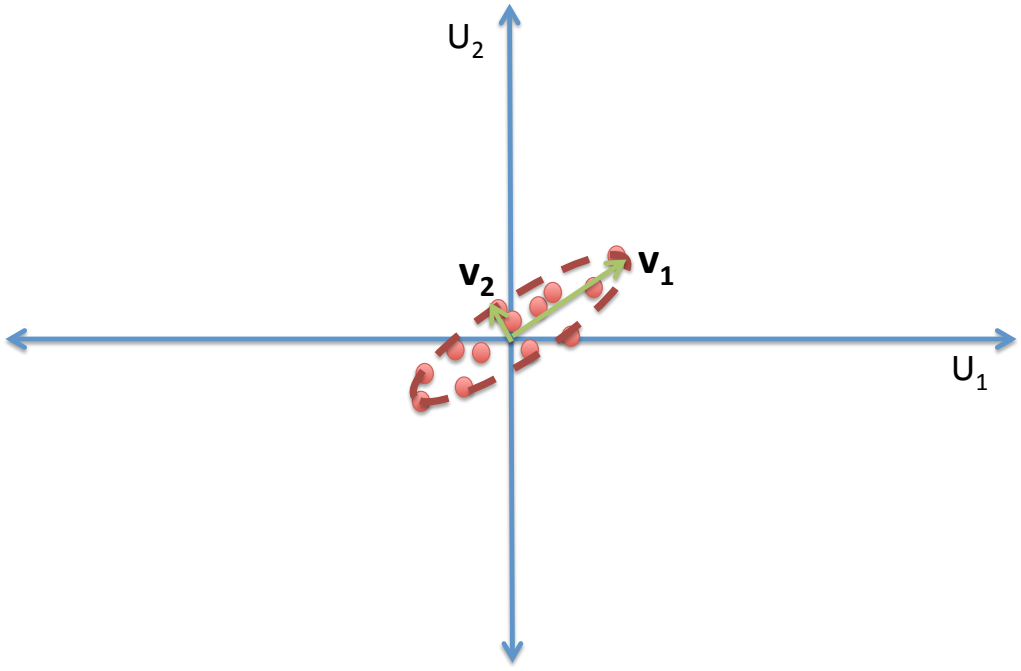
- center data
- fit a multivariate Gaussian distribution fit to the data
- Gaussian has mean 0, covariance Σ
- $\mathbf{w}_1$ is the direction of the covariance ellipse with maximum variance



Principal Components Analysis

This generalizes to multiple dimensions

- suppose we have d original dimensions and we would like K new ones
- the optimal vectors have the same direction as the K ellipse directions with maximum variance



Principal Components Analysis

Alright, so how do we find the directions of these ellipses?

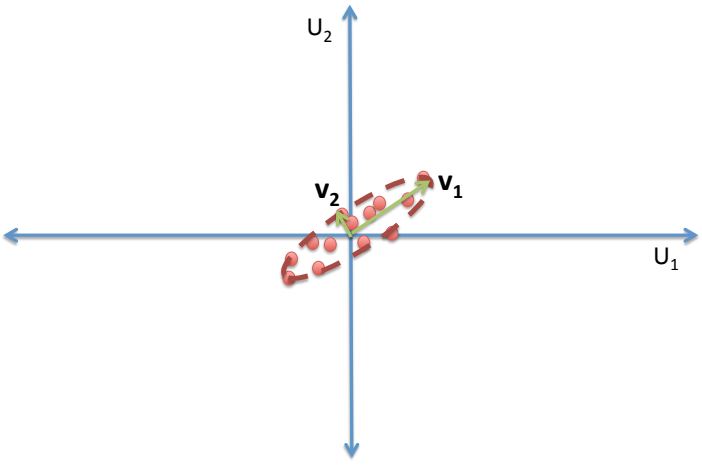
- center everything to have mean 0
- fit a multivariate Gaussian distribution to the data
- estimate the covariance matrix,

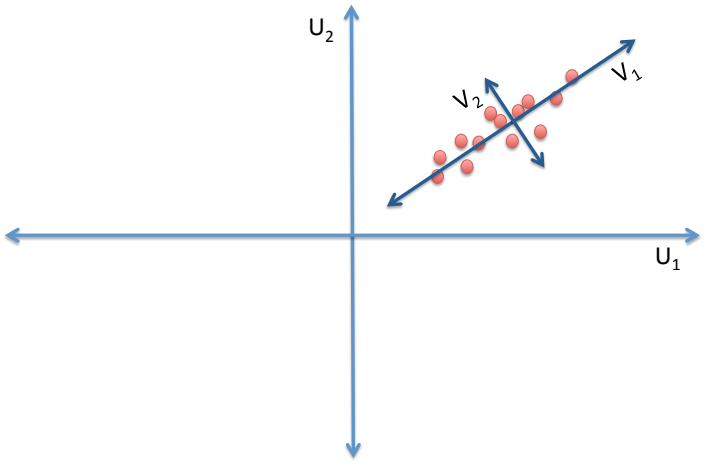
$$\hat{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T$$

- find the *eigenvalues* and *eigenvectors* of the covariance matrix
- the K most descriptive directions are the eigenvectors associated with the K largest eigenvalues

Principal Components Analysis

Why center the data? Aren't the eigenvectors the same?





Principal Components Analysis

PCA:

1. center the data
2. compute the covariance matrix $\frac{1}{n}\mathbf{X}^T\mathbf{X}$
3. compute the eigenvectors of $\frac{1}{n}\mathbf{X}^T\mathbf{X}$, denoted $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_d$, along with their eigenvalues
4. for the eigenvectors with the K largest eigenvalues, make factor loadings by setting

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_K \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & \ldots & w_{1d} \\ w_{21} & w_{22} & \ldots & w_{2d} \\ \vdots & & & \vdots \\ w_{K1} & w_{K2} & \ldots & w_{Kd} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ \vdots \\ x_d \end{bmatrix}$$

Principal Components Analysis

5. do computations in the (smaller) $\mathbf{y}$ space, the space of factor scores
6. transform results back to original space (if needed)

To transform back, we can do some linear algebra

$$\mathbf{x} = \left(\mathbf{W}^T \mathbf{y}\right) + \mu_{\mathbf{x}}$$

(If $\mathbf{y} = \mathbf{W}\mathbf{x}$, why does this work?)

Question of the Day

So **now** let's do our Question of the Day...

Find the principal components for the following dataset:

$$\mathbf{X} = \begin{bmatrix} -6 & -4 \\ -2 & 3 \\ 2 & -3 \\ 6 & 4 \end{bmatrix}$$

Question of the Day

Choosing K

1. How many principal components will I get if I run PCA?

$$\mathbf{X} = \begin{bmatrix} -6 & -4 \\ -2 & 3 \\ 2 & -3 \\ 6 & 4 \end{bmatrix}, \qquad \mathbf{Z} = \begin{bmatrix} -6 & -4 & 3 & -5 & 0 & 7 \\ -2 & 3 & 9 & 0 & -1 & 2 \\ 2 & -3 & 0 & 1 & 4 & -6 \\ 6 & 4 & -1 & -1 & -5 & 3 \end{bmatrix}$$

   This is determined by the *rank* of the data matrix.

2. How well can we reconstruct the data set if we **all** of the eigenvectors?
3. Our number of eigenvectors is larger than we would like. How do we select $K < \text{rank}(\mathbf{X})$?

Choosing K

We will use the *proportion of explained variance*:

- the overall variance of a data set is the sum of the variances of the individual components
- the diagonal term of a covariance matrix is the variance for each element, so this is equivalent to the trace of the covariance matrix, which happens to be the sum of the eigenvalues

$$\text{trace}(\Sigma) = \sum_{j=1}^{d} \lambda_j$$

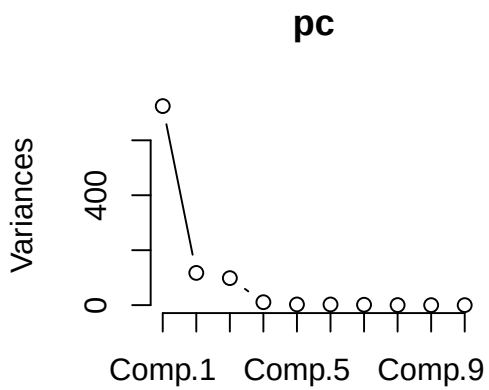
- if we are using only the first K eigenvectors, the variance of the projected data set is $\lambda_1 + \cdots + \lambda_K$
- therefore, the proportion of variance explained using the first K eigenvectors is:

$$\frac{\lambda_1 + \cdots + \lambda_K}{\lambda_1 + \cdots + \lambda_d}$$

Choosing K

Using the proportion of explained variance:

- often, a user will want the smallest data set that is "sufficiently accurate," such as 95% or 99% of the variance explained
- sometimes, we will plot the explained variance and look for a natural break point



- later in the semester, we will learn about model selection tools to balance the number of components against the explained variance

Principal Components Analysis

OK, my head hurts. Can't we just do some R?

R has the function princomp in the stats package[1]

```
> dat = read.table("marks.dat",head=T)
> dim(dat)
> names(dat)
> plot(dat$Phys,dat$Stat)
> pc = princomp(~Stat+Phys,dat)
> pc
> names(pc)
> pc$loading
> plot(pc)
> screeplot(pc,type="lines")
```

[1] Credit: http://astrostatistics.psu.edu/su09/lecturenotes/pca.html

Principal Components Analysis

In higher dimensions, let's look at some quasar data. We have 4817 observations, each with 22 dimensions.[2]

```
> quas = read.table("SDSS_quasar.dat",head=T)
> dim(quas)
> names(quas)
> quas = na.omit(quas)
> dim(quas)
> pc = princomp(quas[,-1],scores=T)
> pc
> plot(pc)
> screeplot(pc)
> screeplot(pc,type="lines")
> pc$loading[,1:2]
> M = pc$loading[,1:2]
> t(M) %*% M #should ideally produce the 2 by 2 identity matrix
> plot(pc$scores[,1],pc$scores[,2],pch=".")
```

[2] Credit: http://astrostatistics.psu.edu/su09/lecturenotes/pca.html

Principal Components Analysis

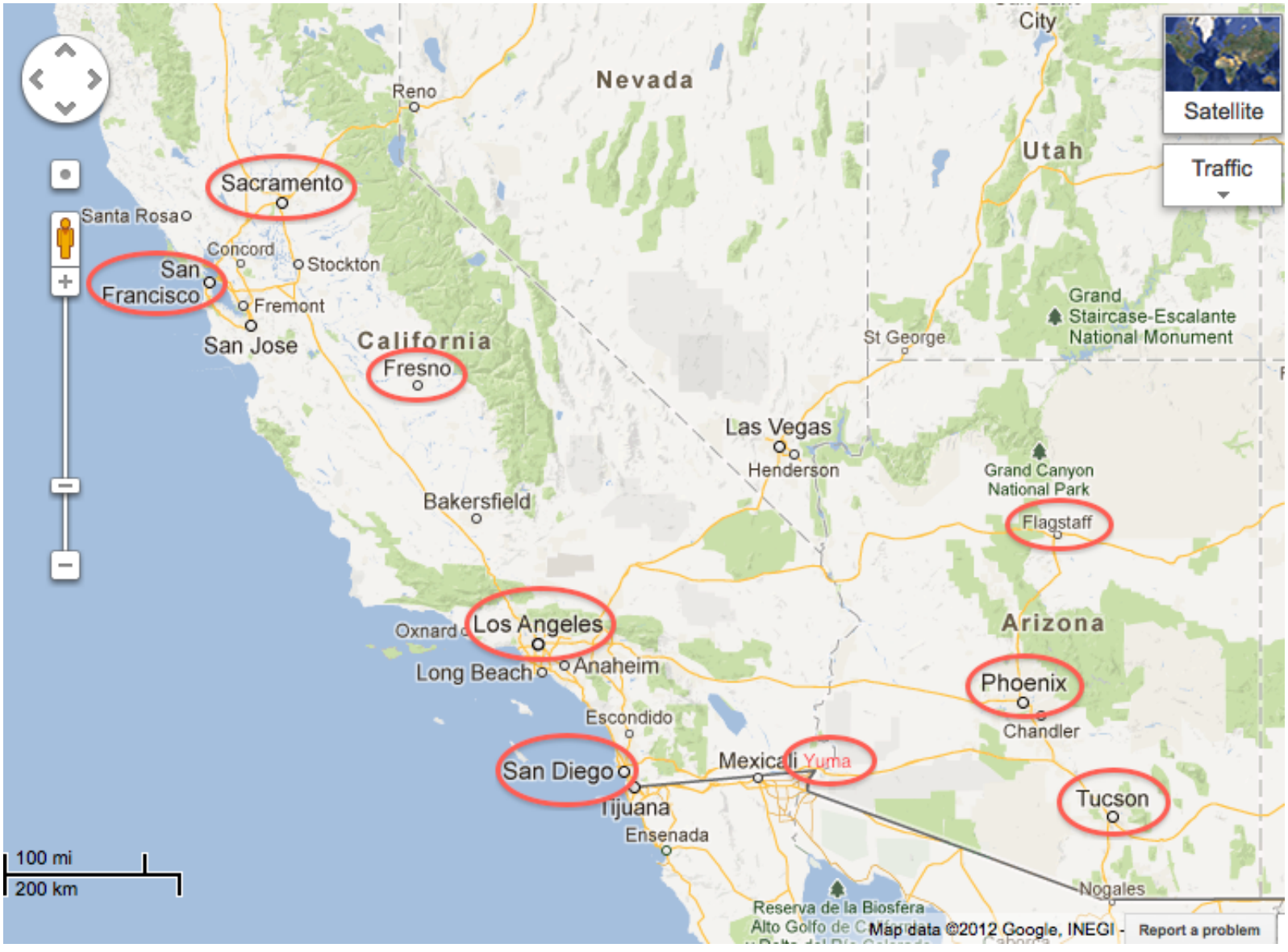
Often, data has many more covariates than observations

Example: average daily temperatures in a set of locations

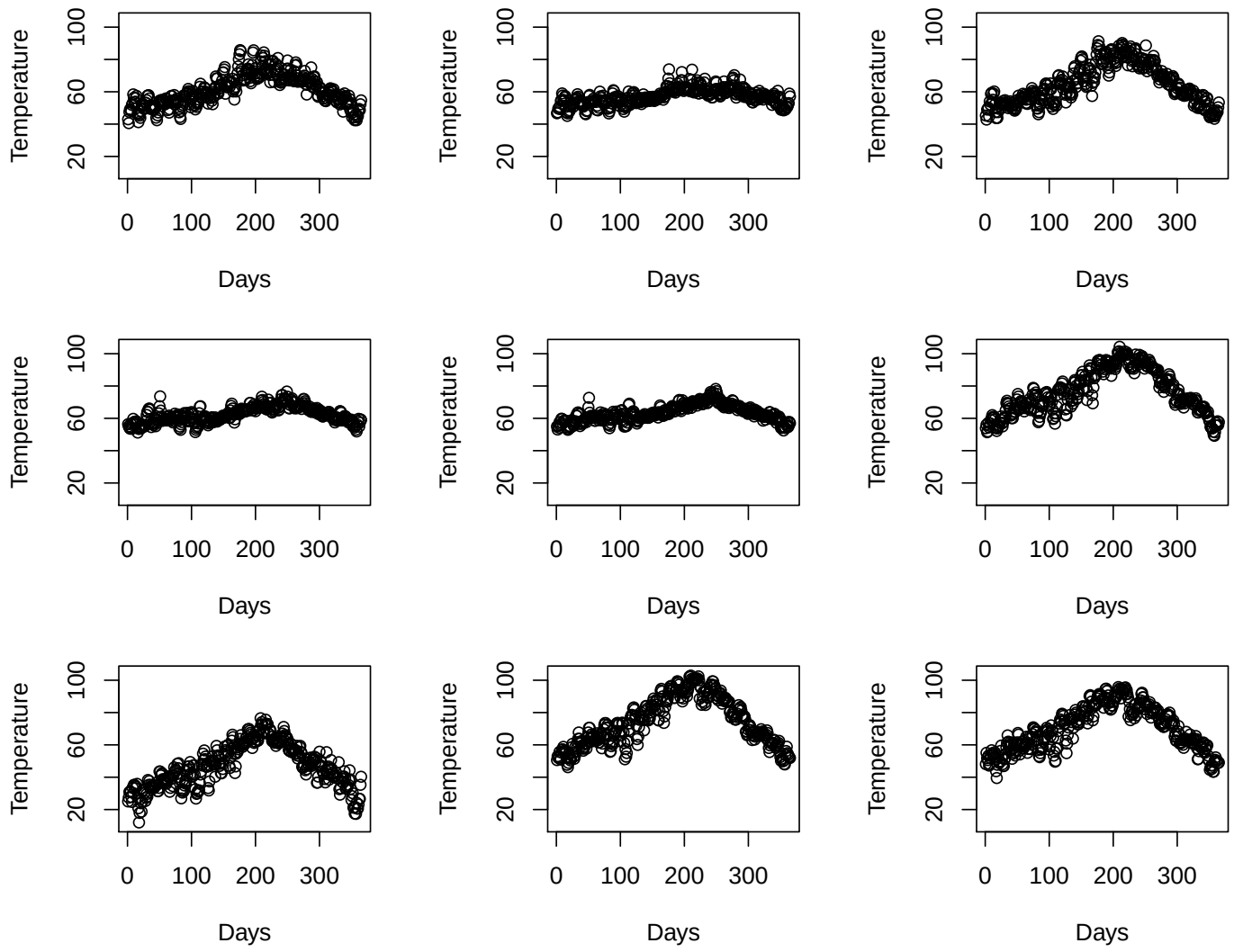
- Cities: Los Angeles, San Diego, Sacramento, San Francisco, Fresno, Phoenix, Tucson, Yuma, and Flagstaff
- Data: average daily temperature for 1995
- Problem: 9 observations and 365 covariates

```
> daily.1995 <- read.csv("Daily1995.csv")
> daily.mean <- apply(daily.1995,1,mean)
> daily.cent <- t(scale(t(daily.1995),center=T,scale=F))
> daily.1995[1:10,]
   los.angeles san.diego sacramento san.francisco fresno phoenix tucson yuma flagstaff
1         56.4      55.0       43.0          46.7   45.3    50.6   48.1 53.9      25.0
2         55.1      53.1       40.6          47.3   42.8    53.0   55.1 56.1      27.4
3         54.3      55.4       47.5          49.6   49.0    52.0   51.8 51.4      30.9
4         53.6      54.2       49.2          50.0   49.2    52.5   50.6 51.9      31.1
5         56.6      57.7       48.6          50.8   50.2    55.7   53.3 57.3      30.8
6         54.4      55.6       48.0          49.3   44.8    51.2   47.2 54.4      27.8
7         53.5      56.3       51.9          54.4   54.0    51.0   48.1 55.4      24.7
8         56.8      59.8       52.9          54.9   52.1    55.3   51.8 56.9      33.5
9         59.7      59.6       58.4          59.0   58.7    57.0   53.5 58.9      30.2
10        57.6      56.8       56.3          57.7   59.4    57.3   56.7 60.2      35.3
```

Principal Components Analysis

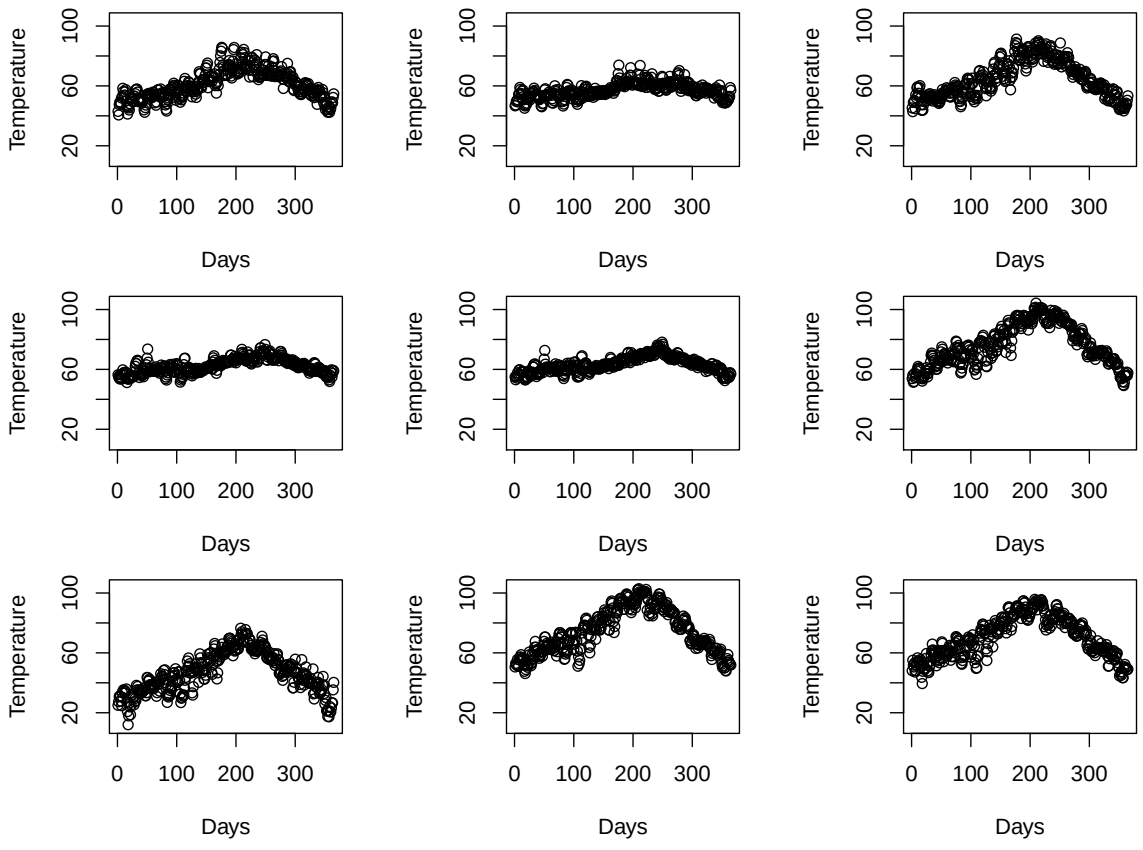


Principal Components Analysis



Top: Sacramento, San Francisco, Fresno. Center: Los Angeles, San Diego, Yuma. Bottom: Flagstaff, Phoenix, Tucson.

Principal Components Analysis



Lots of similarities. Can we use PCA to more compactly represent the data?

Principal Components Analysis

Let's try `princomp`:

```
> plot(1:365,daily.mean,xlab="Days",ylab="Temperature")
> min.val <- min(min(daily.cent))
> max.val <- max(max(daily.cent))
> plot(1:365,daily.cent[,1],xlab="Days",ylab="Temperature",ylim=c(min.val,max.val))
> plot(1:365,daily.cent[,2],xlab="Days",ylab="Temperature",ylim=c(min.val,max.val))
> plot(1:365,daily.cent[,9],xlab="Days",ylab="Temperature",ylim=c(min.val,max.val))
> ppc <- princomp(t(daily.cent))
Error in princomp.default(t(daily.cent)) :
  'princomp' can only be used with more units than variables
```

How do we fix this?

Well, `R` has another method in the `stats` package called `prcomp`

- `princomp` uses `eigen` on the covariance matrix
- `prcomp` uses a singular value decomposition (better stability)

Principal Components Analysis

Let's try `prcomp`:

```
> ppc <- prcomp(t(daily.cent))
> ? prcomp
> names(ppc)
> plot(ppc)
> screeplot(ppc,type="lines")
> summary(ppc)
> plot(1:365,ppc$rotation[,1])
> plot(1:365,ppc$rotation[,2])
> plot(1:365,ppc$rotation[,3])
> ppc$x
```

Principal Components Analysis

So what if I have image data? Can I use PCA then?

Principal Components Analysis

Overview of Homework 2

Principal Components Analysis

PCA summary:

- maps original data to new space in a linear manner by minimizing variance
- sensitive to outliers (variance)
- only finds linear mapping
- if you do not have a lot of structure, you need a lot of components to represent data
- great first step for high dimensional data