Data Mining
S4240 Section 001

Giovanni Motta

Columbia University, Department of Statistics

September 28, 2015

Outline

Today: Principal components analysis (PCA)

1. PCA math
2. PCA examples
3. PCA with R

Reminder of data setup

Data with n observations and p dimensions:

$$\underset{n\times p}{\mathbf{X}} = \begin{bmatrix} x_{11} & \dots & x_{1p} \\ x_{21} & \dots & x_{2p} \\ \vdots & & \vdots \\ x_{n1} & \dots & x_{np} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_n^\top \end{bmatrix} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \dots, \boldsymbol{X}_p]$$

Here $\mathbf{x}^\top$ means the transpose of $\mathbf{x}$.

Today (and often, but not always), **bold** means matrix or vector, and *plain* means scalar.

Note that $\mathbf{X} \in \mathbb{R}^{n\times p}$

Dimensionality Reduction

PCA finds *linear projections* of data

$$\underset{p\times n}{\begin{bmatrix} \boldsymbol{X}_1^\top \\ \boldsymbol{X}_2^\top \\ \vdots \\ \boldsymbol{X}_j^\top \\ \vdots \\ \boldsymbol{X}_p^\top \end{bmatrix}} \rightarrow \underset{\kappa\times n}{\begin{bmatrix} \boldsymbol{Y}_1^\top \\ \boldsymbol{Y}_2^\top \\ \vdots \\ \boldsymbol{Y}_\kappa^\top \end{bmatrix}} = \underset{\kappa\times p}{\begin{bmatrix} w_{11} & w_{12} & \dots & w_{1j} & \dots & w_{1p} \\ w_{21} & w_{22} & \dots & w_{2j} & \dots & w_{2p} \\ \vdots & \vdots & & \vdots & & \vdots \\ w_{\kappa 1} & w_{\kappa 2} & \dots & w_{\kappa j} & \dots & w_{\kappa p} \end{bmatrix}} \underset{p\times n}{\begin{bmatrix} \boldsymbol{X}_1^\top \\ \boldsymbol{X}_2^\top \\ \vdots \\ \boldsymbol{X}_j^\top \\ \vdots \\ \boldsymbol{X}_p^\top \end{bmatrix}}$$

$$\underset{k\times n}{\mathbf{Y}^\top} = \underset{\kappa\times p}{\mathbf{W}^\top} \ \underset{p\times n}{\mathbf{X}^\top} \qquad \text{or} \qquad \underset{n\times\kappa}{\mathbf{Y}} = \underset{n\times p}{\mathbf{X}} \ \underset{p\times\kappa}{\mathbf{W}}$$

Terminology:

- $\boldsymbol{Y}_1, \dots, \boldsymbol{Y}_\kappa$ are the *scores*,
- $\mathbf{w}_1, \dots, \mathbf{w}_\kappa$ are the *loadings*.

PCA procedurally

1. center the data:

$$\sum_{i=1}^{n} x_{ij} = 0 \ \forall j = 1, \ldots, d \quad \text{in R: } \texttt{sum(}\boldsymbol{X}_j\texttt{)} = 0 \ \forall j$$

2. compute the sample covariance matrix $\widehat{\mathbf{\Sigma}} = \frac{1}{n}\mathbf{X}^\top\mathbf{X}$
3. compute the eigenvectors $\mathbf{W}$ corresponding to the largest κ eigenvalues of $\widehat{\mathbf{\Sigma}}$ to get the loadings:

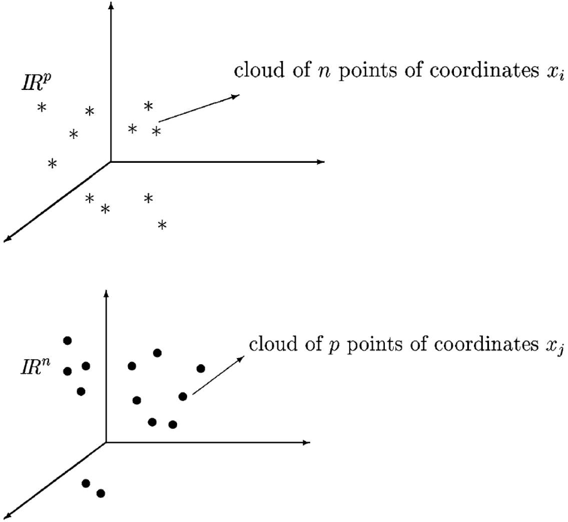
$$\underbrace{\left[\tfrac{1}{n}\mathbf{X}^\top\mathbf{X}\right]}_{p\times p} \ \underbrace{\mathbf{W}}_{p\times\kappa} = \underbrace{\mathbf{W}}_{p\times\kappa} \ \underbrace{\mathbf{\Lambda}}_{\kappa\times\kappa}$$

4. compute factor scores for all $i = 1, \ldots, n$

$$\mathbf{y}_i = \begin{bmatrix} y_{i1} \\ y_{i2} \\ \vdots \\ y_{ij} \\ \vdots \\ y_{i\kappa} \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & \ldots & w_{1p} \\ w_{21} & w_{22} & \ldots & w_{2p} \\ \vdots & & & \vdots \\ w_{\kappa 1} & w_{\kappa 2} & \ldots & w_{\kappa p} \end{bmatrix} \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{ij} \\ \vdots \\ x_{ip} \end{bmatrix}$$

PCA: $\mathbf{X} =$ n points in $\mathbb{R}^p =$ p points in $\mathbb{R}^n$

from Hardle & Simar (2012)



Transition formulas

$$\underset{p\times p}{\left[\mathbf{X}^\top\mathbf{X}\right]}\ \underset{p\times\kappa}{\mathbf{W}} = \underset{p\times\kappa}{\mathbf{W}}\ \underset{\kappa\times\kappa}{\mathbf{\Lambda}}$$

$$\underset{n\times n}{\left[\mathbf{X}\mathbf{X}^\top\right]}\ \underset{n\times\kappa}{\mathbf{X}\mathbf{W}} = \underset{n\times\kappa}{\mathbf{X}\mathbf{W}}\ \underset{\kappa\times\kappa}{\mathbf{\Lambda}}$$

$$\underset{n\times n}{\left[\mathbf{X}\mathbf{X}^\top\right]}\ \underset{n\times\kappa}{\mathbf{X}\mathbf{W}\mathbf{\Lambda}^{-1/2}} = \underset{n\times\kappa}{\mathbf{X}\mathbf{W}\mathbf{\Lambda}^{-1/2}}\ \underset{\kappa\times\kappa}{\mathbf{\Lambda}}$$

- $\underset{p\times\kappa}{\mathbf{W}}$ = eigenvectors of $\mathbf{X}^\top\mathbf{X}$
- $\underset{n\times\kappa}{\mathbf{V}} = \mathbf{X}\mathbf{W}\mathbf{\Lambda}^{-1/2} = \mathbf{Y}\,\mathbf{\Lambda}^{-1/2}$ = eigenvectors of $\mathbf{X}\mathbf{X}^\top$
- $\underset{n\times\kappa}{\mathbf{Y}} = \underset{n\times p}{\mathbf{X}}\ \underset{p\times\kappa}{\mathbf{W}}$ = projection of n observations from $\mathbb{R}^p$ to $\mathbb{R}^\kappa$
- $\underset{p\times\kappa}{\mathbf{\Upsilon}} = \underset{p\times n}{\mathbf{X}^\top}\ \underset{n\times\kappa}{\mathbf{V}}$ = projection of p variables from $\mathbb{R}^n$ to $\mathbb{R}^\kappa$

Notice that $\underset{p\times\kappa}{\mathbf{\Upsilon}} = \underset{p\times n}{\mathbf{X}^\top}\ \underset{n\times\kappa}{\mathbf{V}} = \mathbf{X}^\top\mathbf{X}\mathbf{W}\mathbf{\Lambda}^{-1/2} = \mathbf{W}\mathbf{\Lambda}^{1/2}$

Transition formulas

Projections

- $\underset{n\times\kappa}{\mathbf{Y}} = \underset{n\times p}{\mathbf{X}}\ \underset{p\times\kappa}{\mathbf{W}}$
- $\underset{p\times\kappa}{\boldsymbol{\Upsilon}} = \underset{p\times\kappa}{\mathbf{W}}\,\underset{\kappa\times\kappa}{\boldsymbol{\Lambda}^{1/2}}$

Transitions between projections

- $\mathbf{V} = \mathbf{X}\mathbf{W}\boldsymbol{\Lambda}^{-1/2}$
- $\mathbf{W} = \mathbf{X}^\top\mathbf{V}\boldsymbol{\Lambda}^{-1/2}$

Spectral decompositions

$$\underset{p\times p}{\left[\mathbf{X}^\top\mathbf{X}\right]}\ \underset{p\times\kappa}{\mathbf{W}} = \underset{p\times\kappa}{\mathbf{W}}\ \underset{\kappa\times\kappa}{\boldsymbol{\Lambda}}$$

$$\underset{n\times n}{\left[\mathbf{X}\mathbf{X}^\top\right]}\ \underset{n\times\kappa}{\mathbf{V}} = \underset{n\times\kappa}{\mathbf{V}}\ \underset{\kappa\times\kappa}{\boldsymbol{\Lambda}}$$

Singular values decomposition ($\mathrm{r} := \mathrm{rk}(\mathbf{X}) \leq \min\{n, p\}$)

$$\underset{n\times p}{\mathbf{X}} = \underset{n\times \mathrm{r}}{\mathbf{V}}\ \underset{\mathrm{r}\times \mathrm{r}}{\boldsymbol{\Lambda}^{1/2}}\ \underset{\mathrm{r}\times p}{\mathbf{W}^\top} = \underset{n\times \mathrm{r}}{\mathbf{Y}}\ \underset{\mathrm{r}\times p}{\mathbf{W}^\top}$$

Matrix factorization

Let $\underset{n\times p}{\mathbf{X}}$, with $n > p$; then $\mathrm{r} \le p$, and thus $\kappa \le \mathrm{r} \le p$.

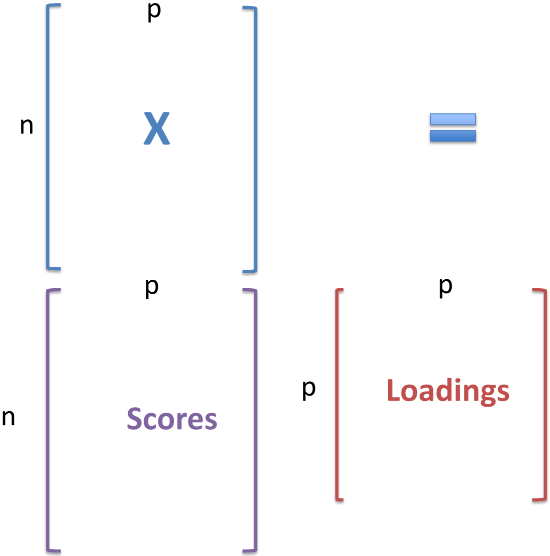
$$\underset{p\times p}{\hat{\mathbf{\Sigma}}} = \underset{p\times\kappa}{\mathbf{W}}\ \underset{\kappa\times\kappa}{\mathbf{\Lambda}}\ \underset{\kappa\times p}{\mathbf{W}^\top} + \underset{p\times(\mathrm{r}-\kappa)}{\mathbf{\Omega}}\ \underset{(\mathrm{r}-\kappa)\times(\mathrm{r}-\kappa)}{\mathbf{\Delta}}\ \underset{(\mathrm{r}-\kappa)\times p}{\mathbf{\Omega}^\top}\ \left(+ \underset{p\times(p-\mathrm{r})}{\mathbf{\Psi}}\ \underset{(p-\mathrm{r})\times(p-\mathrm{r})}{\mathbf{O}}\ \underset{p\times(p-\mathrm{r})}{\mathbf{\Psi}^\top} \right)$$

$$\begin{aligned}
\underset{p\times p}{\mathbf{I}_p} &= \underset{p\times\kappa}{\mathbf{W}}\ \underset{\kappa\times p}{\mathbf{W}^\top} + \underset{p\times(\mathrm{r}-\kappa)}{\mathbf{\Omega}}\ \underset{(\mathrm{r}-\kappa)\times p}{\mathbf{\Omega}^\top} \\
\underset{n\times p}{\mathbf{X}}\ \underset{p\times p}{\mathbf{I}_p} &= \underset{n\times p}{\mathbf{X}}\ \underset{p\times\kappa}{\mathbf{W}}\ \underset{\kappa\times p}{\mathbf{W}^\top} + \underset{n\times p}{\mathbf{X}}\ \underset{p\times(\mathrm{r}-\kappa)}{\mathbf{\Omega}}\ \underset{(\mathrm{r}-\kappa)\times p}{\mathbf{\Omega}^\top} \\
\underset{n\times p}{\mathbf{X}} &= \underset{n\times\kappa}{\mathbf{Y}}\ \underset{\kappa\times p}{\mathbf{W}^\top} + \underset{n\times p}{\mathbf{X}}\ \underset{p\times(\mathrm{r}-\kappa)}{\mathbf{\Omega}}\ \underset{(\mathrm{r}-\kappa)\times p}{\mathbf{\Omega}^\top}
\end{aligned}$$

- If $\kappa = \mathrm{r}$: $\underset{n\times p}{\mathbf{X}} = \underset{n\times\kappa}{\mathbf{Y}}\ \underset{\kappa\times p}{\mathbf{W}^\top}$
- If $\kappa < \mathrm{r}$: $\underset{n\times p}{\mathbf{X}} \approx \underset{n\times\kappa}{\mathbf{Y}}\ \underset{\kappa\times p}{\mathbf{W}^\top}$

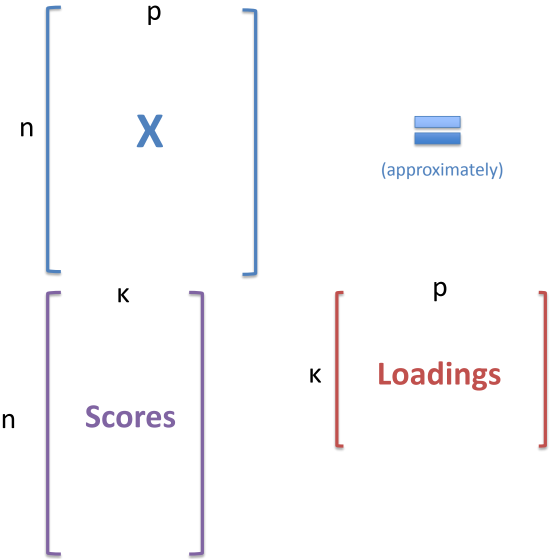
Linear dimensionality reduction more broadly ($\mathrm{r} = p$)

Linear feature extraction can be viewed as a matrix factorization:

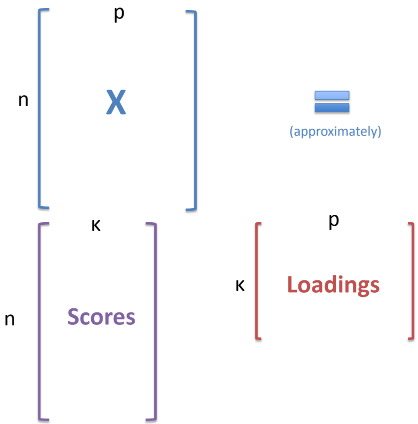


Linear dimensionality reduction more broadly $(\mathrm{r} = p)$

Linear feature extraction can be viewed as a matrix factorization:



Linear dimensionality reduction more broadly $(\mathrm{r} = p)$



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

Principal Components Analysis

Let's do an example:

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

$$\begin{aligned} \tfrac{1}{n}\mathbf{X}^\top\mathbf{X} &= \\ \det\!\left(\tfrac{1}{n}\mathbf{X}^\top\mathbf{X} - \lambda\mathbf{I}_2\right) &= \\ \left[\lambda_1, \lambda_2\right] &= \\ \left[\mathbf{w}_1, \mathbf{w}_2\right] &= \\ \left[\mathbf{y}_1, \mathbf{y}_2\right] &= \end{aligned}$$

Choosing κ

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 use **all** of the eigenvectors?
3. Our number of eigenvectors is larger than we would like. How do we select $\kappa < \mathrm{rank}(\mathbf{X})$?

Choosing κ

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, aka the sum of the eigenvalues

$$\boxed{\mathrm{trace}(\boldsymbol{\Sigma}) = \sum_{j=1}^{p} \lambda_j}$$

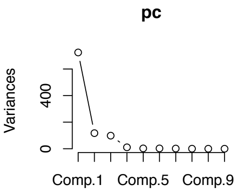
- if we are using only the first κ eigenvectors, the variance of the projected data set is $\lambda_1 + \cdots + \lambda_\kappa$
- therefore, the proportion of variance explained using the first κ eigenvectors is:

$$\frac{\lambda_1 + \cdots + \lambda_\kappa}{\lambda_1 + \cdots + \lambda_p}$$

Choosing κ

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

PCA: interpretation

Covariance between the p original variables and the κ PCs

$$\underset{p\times\kappa}{\hat{\mathbf{\Sigma}}_{XY}} = \frac{1}{n}\underset{p\times n}{\mathbf{X}^\top}\underset{n\times\kappa}{\mathbf{Y}} = \frac{1}{n}\mathbf{X}^\top\mathbf{X}\mathbf{W} = \underset{p\times\kappa}{\mathbf{W}}\underset{\kappa\times\kappa}{\mathbf{\Lambda}}$$

Correlation between the p original variables and the κ PCs

$$\begin{aligned}
\underset{p\times\kappa}{\hat{\mathbf{R}}_{XY}} &= \mathrm{diag}[(\frac{1}{n}\underset{p\times p}{\mathbf{X}^\top\mathbf{X}})]^{-1/2}\ \underset{p\times\kappa}{\hat{\mathbf{\Sigma}}_{XY}}\ [\mathrm{diag}(\frac{1}{n}\underset{\kappa\times\kappa}{\mathbf{Y}^\top\mathbf{Y}})]^{-1/2} \\
&= \mathrm{diag}[(\frac{1}{n}\underset{p\times p}{\mathbf{X}^\top\mathbf{X}})]^{-1/2}\ \left(\underset{p\times\kappa}{\mathbf{W}}\underset{\kappa\times\kappa}{\mathbf{\Lambda}}\right)\ [\underset{\kappa\times\kappa}{\mathbf{\Lambda}}]^{-1/2} \\
&= \mathrm{diag}[(\frac{1}{n}\underset{p\times p}{\mathbf{X}^\top\mathbf{X}})]^{-1/2}\ \underset{p\times\kappa}{\mathbf{W}}\underset{\kappa\times\kappa}{\mathbf{\Lambda}^{1/2}}
\end{aligned}$$

Correlation between the $j-$th original variable and the $k-$th PC, $j = 1,\ldots,p$, $k = 1,\ldots,\kappa \leq p$

$$\hat{\rho}(\boldsymbol{X}_j, \boldsymbol{Y}_k) = \frac{w_{jk}}{\hat{\sigma}_{X_j}}\sqrt{\lambda_k},$$

where $\hat{\sigma}^2_{X_j} := \frac{1}{n}\sum_{i=1}^n x_{ij}^2$. Note that $\sum_{k=1}^p [\hat{\rho}(\boldsymbol{X}_j, \boldsymbol{Y}_k)]^2 = 1\,\forall j$.

PCA: interpretation

Let $\mathbf{v}$ and $\mathbf{z}$ be two vectors in $\mathbb{R}^p$. The angle θ between $\mathbf{v}$ and $\mathbf{z}$ is defined by the cosine of θ

$$\cos\theta = \frac{\mathbf{v}^\top \mathbf{z}}{\|\mathbf{v}\|\ \|\mathbf{z}\|}$$

Assume that $\mathbf{v}$ and $\mathbf{z}$ are centered data vectors, that is, $\sum_{j=1}^p \mathrm{v}_j = \sum_{j=1}^p \mathrm{z}_j = 0$. Then the cosine of the angle between them is equal to their correlation

$$\hat{\rho}_{\mathrm{vz}} = \frac{\sum_{j=1}^p \mathrm{v}_j \mathrm{z}_j}{\sqrt{\left(\sum_{j=1}^p \mathrm{v}_j^2\right)\ \left(\sum_{j=1}^p \mathrm{z}_j^2\right)}} = \frac{\mathbf{v}^\top \mathbf{z}}{\|\mathbf{v}\|\ \|\mathbf{z}\|} = \cos\theta$$

Quality of the representation of the i-th individual on the k-th factorial axis, $i = 1, \ldots, n$, $k = 1, \ldots, \kappa \leq p$:

$\hat{\rho}(\mathrm{x}_i, \mathrm{w}_k) = \frac{\mathbf{x}_i^\top \mathbf{w}_k}{\|\mathbf{x}_i\|\,\|\mathbf{w}_k\|} = \frac{y_{ik}}{\|\mathbf{x}_i\|}$. Note that $\sum_{k=1}^p [\hat{\rho}(\mathrm{x}_i, \mathrm{w}_k)]^2 = 1\,\forall i$.

Principal Components Analysis in R

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

```
> dat = read.table("marks.dat",head=TRUE)
> 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 R

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 in R

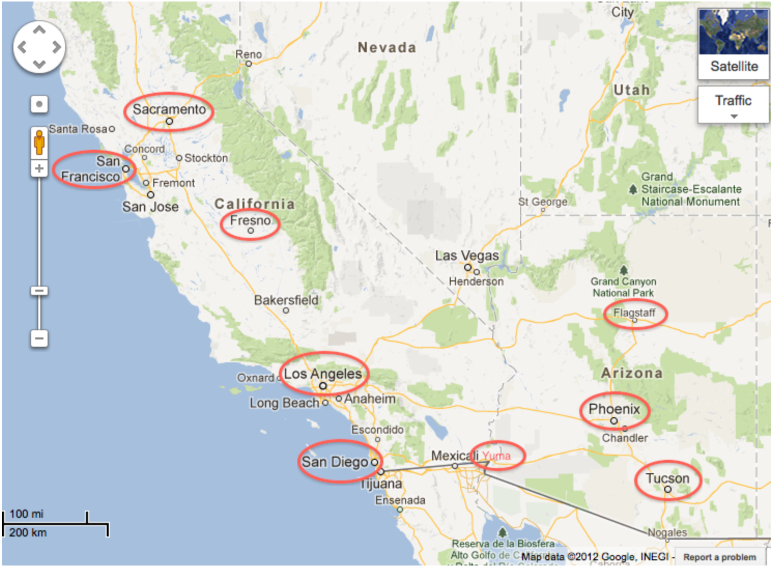
Often, data has many more covariates than observations

Example: average daily temperatures in a set of locations

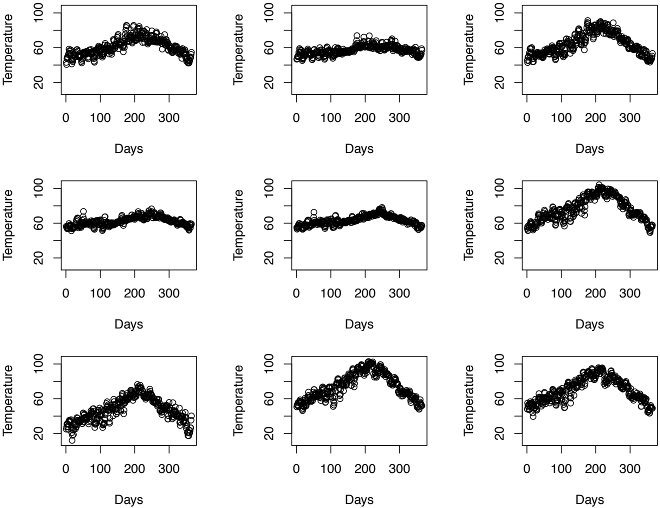
- Cities: Los Angeles, San Diego, Sacramento, San Francisco, Fresno, Phoenix, Tucson, Yuma, and Flagstaff
- Data: average daily temperature for 1995
- Problem: 9 observations and 365 covariates

```
> 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 in R

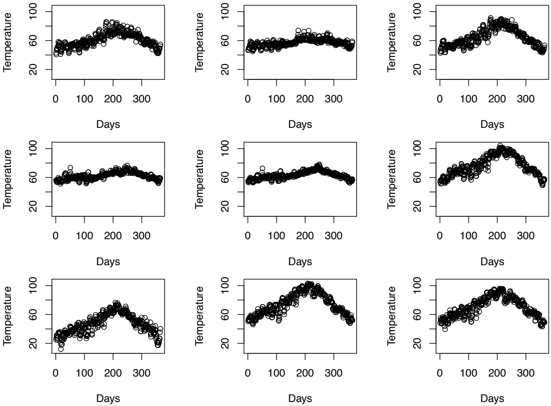


Principal Components Analysis in R



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

Principal Components Analysis in R



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

Principal Components Analysis in R

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 in R

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

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