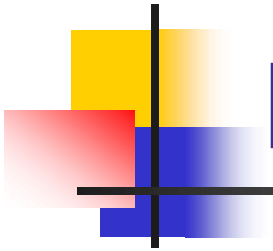
Session 7
Regularization & Dimension Reduction Methods

PSYC 560
Heungsun Hwang



Linear Regression Model

- In the regression setting, the standard linear model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_P X_P + e$$

is commonly used to describe the relationship between a response Y and a set of predictors.

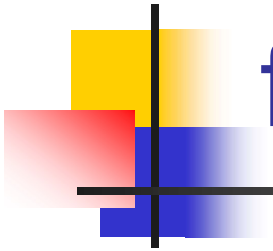
How to estimate coefficients?

- We typically use least squares to estimate the coefficients in the regression model.
  - This method chooses the values of the intercept and slopes that make the sum of the squared residuals as small as possible.
  - In other words, the coefficients are estimated to minimize

$$\text{SS(Residual)} = \sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2$$

$$= \sum_{i=1}^{N}(Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_1 - \cdots - \hat{\beta}_P X_P)^2$$

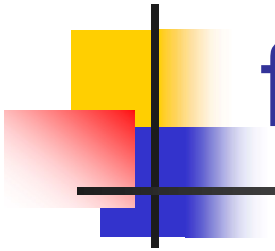
Alternative Fitting Procedures for Linear Models



- We will discuss some ways of improving linear models by replacing least squares fitting with alternative fitting procedures.

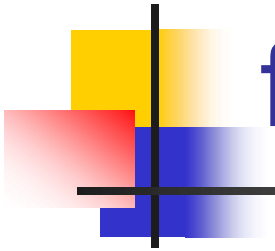
- Alternative fitting procedures may yield better prediction accuracy and/or model interpretability.

Alternative Fitting Procedures for Linear Models



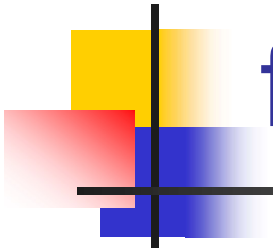
- Prediction accuracy: If the sample size (N) is not much larger than the number of predictors (P), then there can be a lot of variability in the least squares fit, resulting in poor predictions on future observations. If P > N, there is no longer a unique least squares solution, so the method cannot be used at all.

Alternative Fitting Procedures for Linear Models



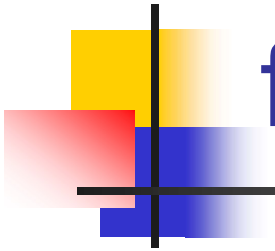
- Model interpretability: Including irrelevant variables leads to unnecessary complexity in the model. By removing these variables (i.e., by setting the corresponding coefficient estimates to zero), we can obtain a model that is more easily interpreted. Least squares is extremely unlikely to lead any coefficient estimates to be exactly zero. Excluding irrelevant variables is called variable selection or feature selection.

Alternative Fitting Procedures for Linear Models

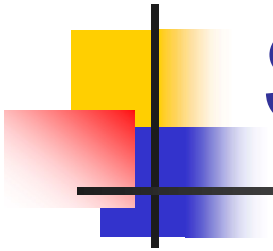


- We will discuss two alternatives to using least squares.
  - Shrinkage (Regularization): This approach involves fitting a model involving all predictors. However, the estimated coefficients are shrunken towards zero relative to the least squares estimates. This shrinkage has the effect of reducing variance. A type of shrinkage can also lead some coefficient estimates to be exactly zero (variable selection).

Alternative Fitting Procedures for Linear Models

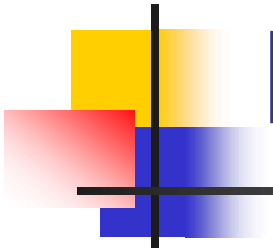


- Dimension Reduction: This approach involves reducing P predictors to D new variables, where D < P. The new variables are computed as components (linear combinations or weighted sums) of the original P predictors. Then, the D components are used as predictors to fit a linear regression via least squares.



Shrinkage Methods

- We fit a model containing all P predictors using a technique that constrains or regularizes the coefficient estimates, or equivalently, that shrinks the estimates towards zero.
- Shrinking the coefficient estimates can significantly reduce their variance.
- Two best-known shrinkage methods are ridge regression and the lasso.

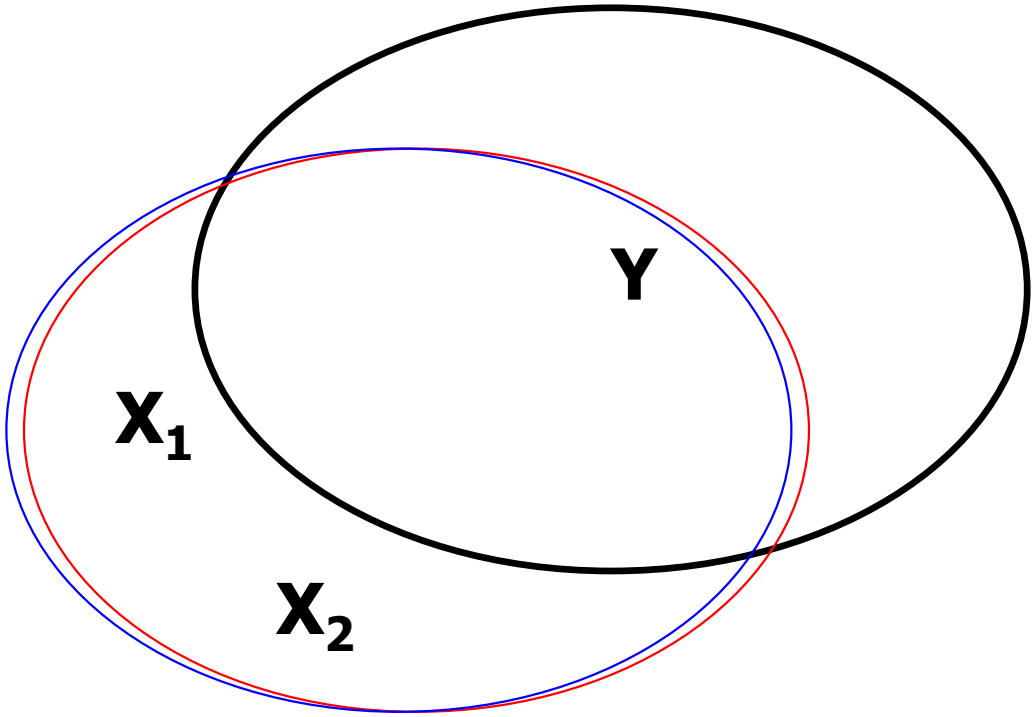


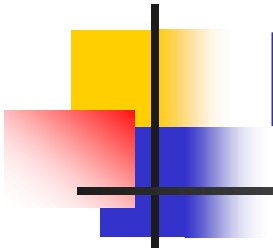
Ridge Regression

- Ridge regression was developed to address multicollinearity (Hoerl & Kennard, 1970).

- Multicollinearity, also termed collinearity or ill-conditioning, generally refers to a data problem that two or more predictors are highly correlated.

Multicollinearity





Multicollinearity

- Consequences:
  - Make it difficult to interpret the unique influence of a given predictor variable
  - Unexpected signs of regression coefficient estimates
  - Unstable regression coefficient estimates
    - Large variances of estimates
  - Computationally, the matrix inversion problem

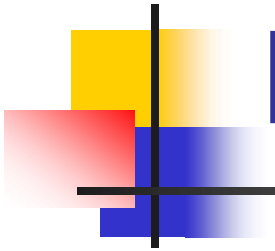
Ridge Regression

- Linear regression estimates the coefficients by minimizing the RSS or SS(Residual).

$$\mathrm{RSS} = \sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{p=1}^{P}\beta_p x_{ip}\right)^2$$

- Ridge regression estimates the coefficients by minimizing a slightly different quantity.

$$\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{p=1}^{P}\beta_p x_{ip}\right)^2 + \lambda\sum_{p=1}^{P}\beta_p^2$$



Ridge Regression

- As with least squares, ridge regression seeks coefficient estimates that fit the data well, by making the RSS small.

- However, the second term, called a shrinkage (or ridge, L_2-norm, or quadratic) penalty, has the effect of shrinking their estimates towards zero.

Ridge Regression

- The tuning parameter λ controls for the relative impact of these two terms on the estimation of coefficients.
- When λ = 0, the penalty term has no effect, and ridge regression will produce the LS estimates.
- As λ → ∞, the term's impact grows, and the ridge regression estimates will approach zero.
- Ridge regression will produce a different set of coefficient estimates for each value of λ. Selecting a good value of λ is critical (cross validation).

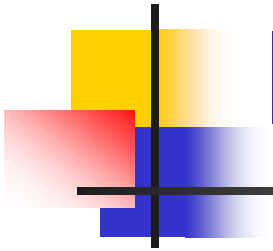
Why Does Ridge Regression Improve over Least Squares?

- To give further insight into how ridge regression works, we can consider an alternative way of obtaining ridge regression estimates. That is, the same problem can be solved by minimizing the RSS

$$\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{p=1}^{P}\beta_p x_{ip}\right)^2$$

subject to $\sum_{p=1}^{P}\beta_p^2 \leq \tau$ (Hastie et al., 2001, p. 59).

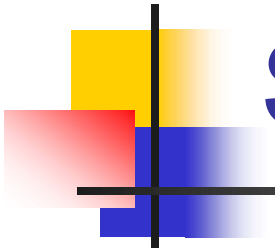
Why Does Ridge Regression Improve over Least Squares?



- The size constraint ($\sum_{p=1}^{P} \beta_p^2 \leq \tau$) is imposed on the regression coefficients.

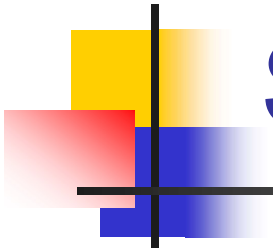
- When two predictor variables are highly correlated, a very large positive regression coefficient of one predictor variable can be offset by a very large negative regression coefficient of the other predictor variable. By imposing the size constraint, ridge regression keeps the magnitudes of the regression coefficients within a certain range.

Ridge Regression vs. Least Squares



- Ridge regression's advantage over least squares is rooted in the bias-variance trade-off.
- As λ increases, the flexibility of the ridge regression fit decreases, leading to decreased variance but increased bias.
- In general, in situations when the relationship between the response and the predictors is close to linear, the least squares estimates will have low bias but may have high variance.
- For any fixed value of λ, ridge regression is computationally as efficient as least squares.

Ridge Regression vs. Least Squares



- Ridge regression works best when the least squares estimates have high variance, e.g., when the number of predictors (P) is almost as large as the number of observations (N) or when there is a high level of multicollinearity, the least squares estimates will be extremely variable.

- If P > N, the least squares estimates do not even have a unique solution.

Example: Ridge Regression

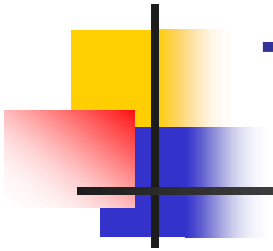
- The Major League Baseball data (Hitters.csv) that include records and salaries for baseball players in the 1986 and 1987 seasons
- DV:
  - Salary: 1987 annual salary on opening day in thousands of dollars
- Predictors:

- AtBat: # of times at bat in 1986
- Hits: # of hits in 1986
- HmRun: # of home runs in 1986
- Runs: # of runs in 1986
- RBI: # of runs batted in in 1986
- Walks: # of walks in 1986
- Years: # of years in the major leagues
- CAtBat: # of times at bat during his career
- CHits: # of hits during his career
- CHmRun: # of home runs during his career
- CRuns: # of runs during his career
- CRBI: # of runs batted in during his career
- CWalks: # of walks during his career
- League: player's league at the end of 1986 (A and N)
- Division: player's division at the end of 1986 (E and W)
- PutOuts: # of put outs in 1986
- Assists: # of assists in 1986
- Errors: # of errors in 1986
- NewLeague: player's league at the beginning of 1987 (A and N)

https://cran.r-project.org/web/packages/ISLR/ISLR.pdf

Example: Ridge Regression

	lam=10^9	lam=10^6	lam=10^3	lam=10^0
(Intercept)	5.501681e+02	5.479665e+02	87.08331332	211.29676862
AtBat	6.950346e-07	6.947008e-04	0.12316656	-2.52020209
Hits	2.464619e-06	2.462385e-03	0.62191426	9.08804182
HmRun	9.792868e-06	9.781230e-03	1.27359344	-3.55051191
Runs	3.944192e-06	3.937542e-03	0.79695583	-1.56243688
RBI	4.114842e-06	4.107052e-03	0.83256809	0.28953783
Walks	5.009139e-06	4.999945e-03	1.25990685	5.81827123
Years	1.969997e-05	1.968102e-02	2.46487778	-2.22774082
CAtBat	5.565417e-08	5.556640e-05	0.01031190	-0.15450071
CHits	2.080608e-07	2.076415e-04	0.04450730	0.05660042
CHmRun	1.617496e-06	1.613645e-03	0.39097718	0.96365607
CRuns	4.195439e-07	4.183310e-04	0.09052010	1.14332672
CRBI	4.272296e-07	4.258274e-04	0.10079606	0.86889954
CWalks	4.497039e-07	4.479279e-04	0.07525672	-0.72653672
League	-1.479685e-05	-1.463203e-02	8.89517753	26.05212188
Division	-1.039619e-04	-1.037738e-01	-51.15351556	-124.87961918
PutOuts	3.153370e-07	3.147069e-04	0.14430748	0.41476962
Assists	1.294450e-07	1.291365e-04	0.06009958	0.68090169
Errors	3.179735e-07	3.149243e-04	-0.35207093	-5.48217814
NewLeague	-2.225218e-07	-1.045981e-04	12.92124231	36.43239183



The Lasso (Lasso Regression)

- Ridge regression keeps all P predictors and produces their coefficient estimates.
- The ridge penalty will shrink all coefficients towards zero, but it will not set any of them exactly to zero (unless $\lambda = \infty$).
- This may not be a problem for prediction accuracy. But, it can create a challenge in model interpretation when P is quite large.
- The lasso is an alternative to ridge regression that overcomes this disadvantage.

The Lasso

- The Lasso estimates the coefficients by minimizing the following quantity.

$$\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{p=1}^{P}\beta_p x_{ip}\right)^2 + \lambda\sum_{p=1}^{P}|\beta_p|$$

- As compared to ridge regression, the lasso replaces the ridge penalty with the lasso (or L_1-norm) penalty.

- This L1 penalty has the effect of forcing some of the coefficient estimates to be exactly zero when λ is sufficiently large.

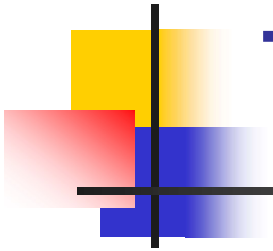
The Lasso

- The lasso problem is equivalent to minimizing the RSS

$$\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{p=1}^{P}\beta_p x_{ip}\right)^2$$

subject to $\sum_{p=1}^{P}|\beta_p| \leq \tau$.

- When we perform the lasso, we are trying to find the set of coefficient estimates that lead to the smallest RSS, subject to the size constraint.

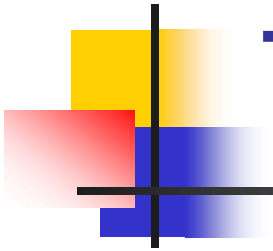


The Lasso

- When τ is very large, then this constraint is not difficult to satisfy, and the coefficient estimates can be large.

- In fact, if τ is large enough, then the lasso will simply yield the least squares solution.
  - If τ is greater than the sum of the least squares estimates in absolute value, the constraint has no effect.

- If τ is small, some of the lasso coefficient estimates are to be exact zero to satisfy the constraint.



The Lasso

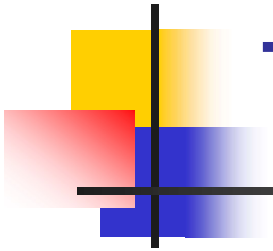
- Thus, the lasso performs variable selection. That is, it tends to produce simpler models than those produced by ridge regression, improving interpretability.

- We say that the lasso yields sparse models, i.e., models that involve only a subset of predictors.

- As in ridge regression, selecting a good value of λ is critical (cross validation).

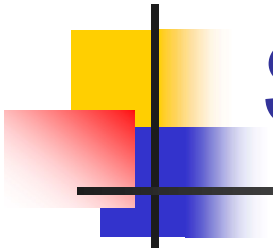
Example: The Lasso

	lam=10^1	lam=10^0	lam=10^-1	lam=10^-2
(Intercept)	74.4405180	205.1015733	208.86737289	209.14451375
AtBat	.	-2.4740268	-2.65866553	-2.66575403
Hits	2.0177886	8.8966278	9.77462213	9.79731222
HmRun	-0.5976880	-3.0099117	-2.44009624	-2.50739018
Runs	.	-1.0495697	-2.23773090	-2.25485461
RBI	.	.	.	0.02416709
Walks	1.9368036	5.5398404	6.06865524	6.08937697
Years	.	-1.7990503	.	0.03521957
CAtBat	.	-0.1365554	-0.18797197	-0.18810447
CHits	.	.	-0.01301876	-0.01446144
CHmRun	0.6330213	0.7567654	0.55116389	0.55651867
CRuns	.	1.0626726	1.44962706	1.45111478
CRBI	0.5258825	0.9470540	1.04269728	1.04505266
CWalks	.	-0.6837595	-0.78845582	-0.79163410
League	10.2640608	23.2187281	23.77164154	23.95545730
Division	-132.9306928	-125.0830367	-121.65380042	-121.82959313
PutOuts	0.3513209	0.4107125	0.41818134	0.41835661
Assists	0.1499942	0.6380844	0.70811357	0.71164465
Errors	-0.2243985	-4.7520735	-5.40594651	-5.46737391
NewLeague	24.9382535	37.0937356	40.58121438	40.63039496



The Lasso vs. Ridge Regression

- The lasso has an advantage over ridge regression, in that it can produce simpler and more interpretable models that involve only a subset of predictors.
- In terms of prediction accuracy, either of the two methods will not universally dominate the other. In general, the lasso is expected to perform better when a relatively small number of predictors have substantially large, whereas ridge regression is to perform better when all coefficients are of roughly equal size.
- CV can be used to compare the two methods on a particular data set.



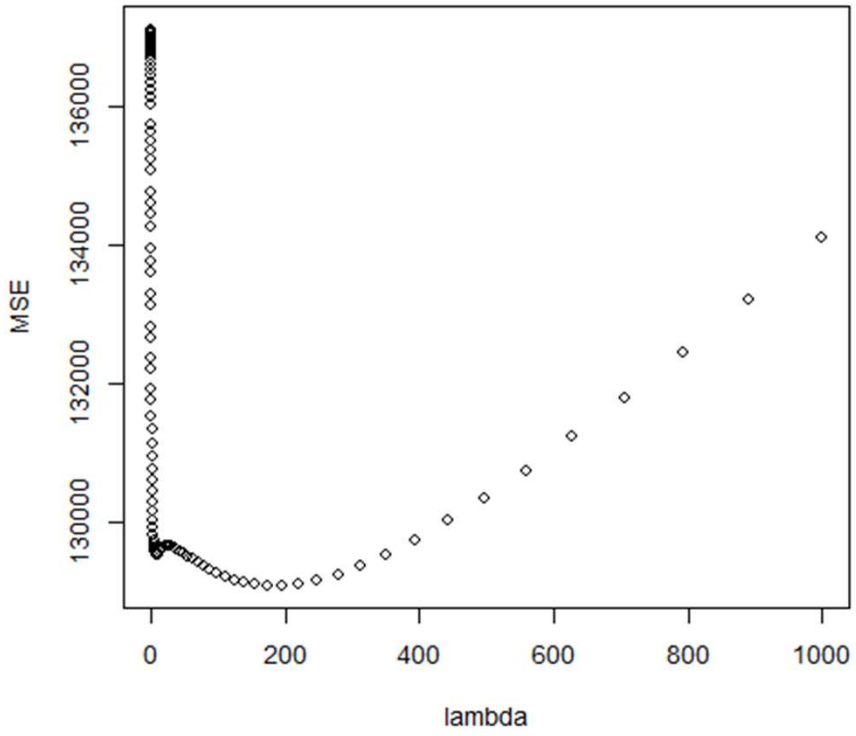
Selecting the Tuning Parameter

- Cross validation provides a simple way of selecting a value for λ in ridge regression and the lasso.
- We choose a grid of λ values and compute the cross-validation error for each value of λ. We then select the tuning parameter for which the cross-validation error is the smallest.
- Finally, the model is re-fit using all the available observations and the selected value of the tuning parameter.

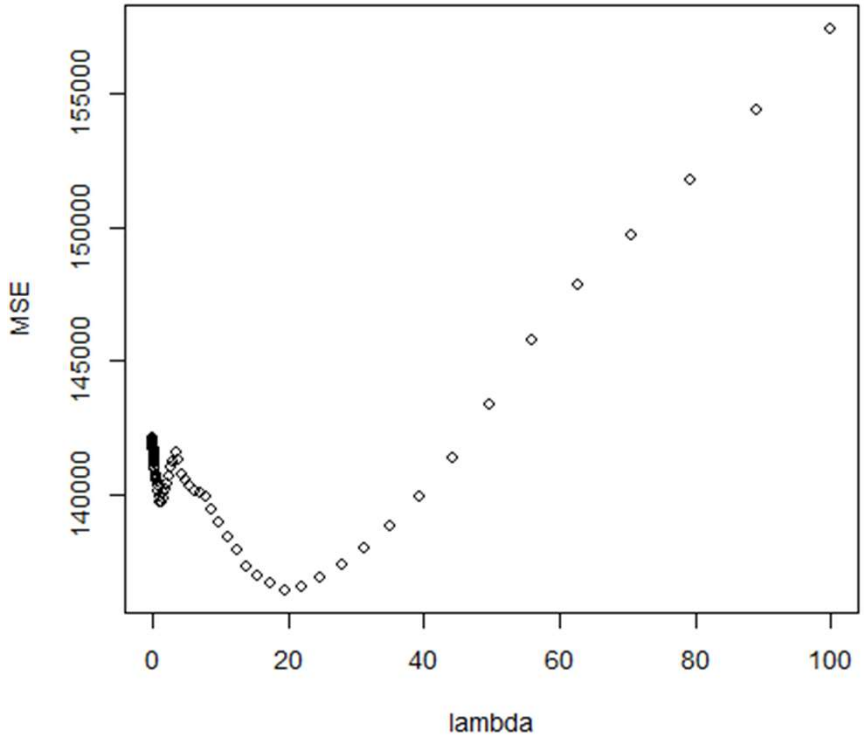
Example: Cross Validation for Ridge and Lasso

Hitters_training.csv

Ridge



Lasso



Example: Ridge and Lasso

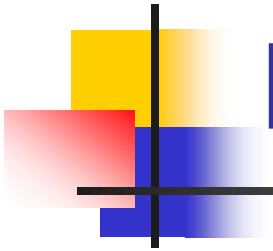
Hitters_training.csv

	Ridge	Lasso	No Reg
(Intercept)	5.445742e+01	114.90815640	152.20667028
AtBat	-1.036336e-02	.	-2.73455318
Hits	1.180396e+00	1.88302825	10.16845282
HmRun	-9.543557e-01	.	-2.04325704
Runs	8.032000e-01	.	-2.55611277
RBI	9.660890e-01	.	-0.14001318
Walks	1.932747e+00	1.92631459	6.23602955
Years	-1.997343e+00	.	0.01603933
CAtBat	7.559330e-03	.	-0.16397743
CHits	6.447880e-02	.	-0.19808368
CHmRun	7.511652e-01	0.32265343	0.20762834
CRuns	1.364149e-01	.	1.60133635
CRBI	1.846534e-01	0.57158603	1.20316567
CWalks	-3.657721e-03	.	-0.84351392
League	2.043242e+01	.	20.82926895
Division	-1.084553e+02	-114.19332716	-122.31543710
PutOuts	2.985322e-01	0.32270596	0.41945478
Assists	1.917919e-01	0.05638781	0.70481669
Errors	-2.046842e+00	.	-5.30400295
NewLeague	2.690478e+01	9.22924278	43.79582769

Example: Ridge and Lasso

Hitters_test.csv

MSE_Ridge:	94256.1
MSE_Lasso:	97640.2
MSE_No Reg:	116358.2

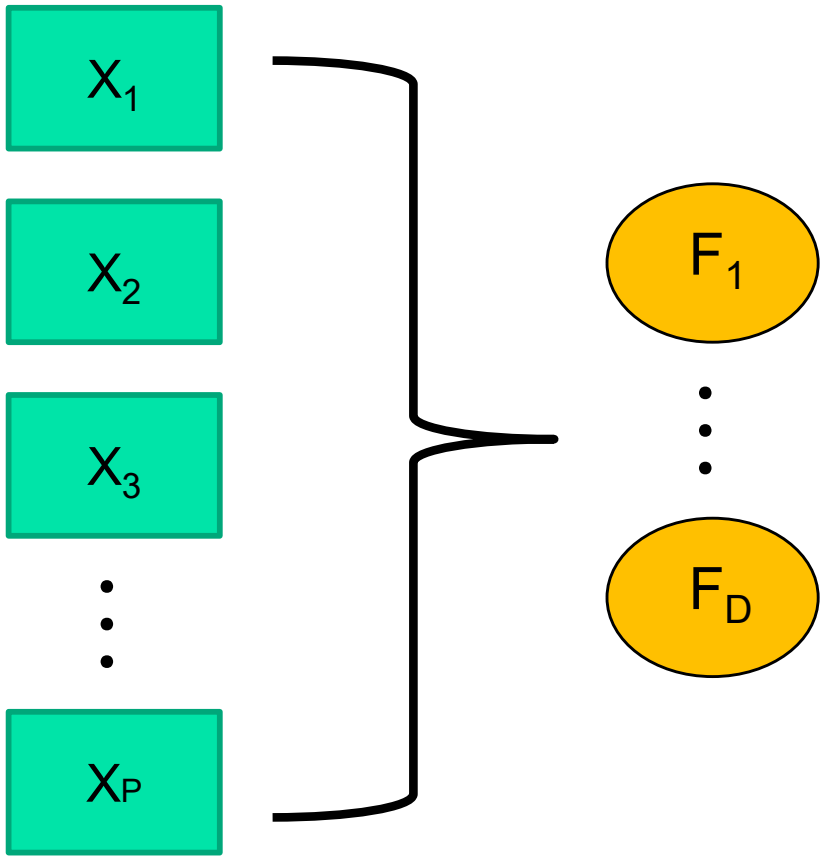


Dimension Reduction Methods

- Dimension reduction methods refer to a class of approaches that transform original predictors to new variables and then fit a least squares model using the new variables.

- In general, dimension reduction refers to reducing P variables to D (new) variables (D < P).
  - It may be great to account for most of the information of the original variables with a smaller number of new variables (a low-dimensional set of new variables).

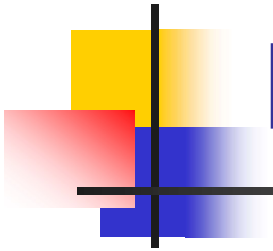
Dimension Reduction?



P >> D

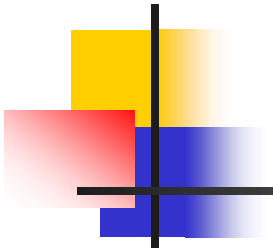
Imaginary Example: Bank

- Friendliness of staff. - Time spent in line-up. - Assistance via telephone.	Service
- Distance of bank from home. - Hours of operation. - Availability of parking. - Proximity to other stores where you frequently shop.	Convenience
- Monthly account fee. - Charge for with-drawls and deposits. - Loan interest rate.	Cost



Principal Components Analysis

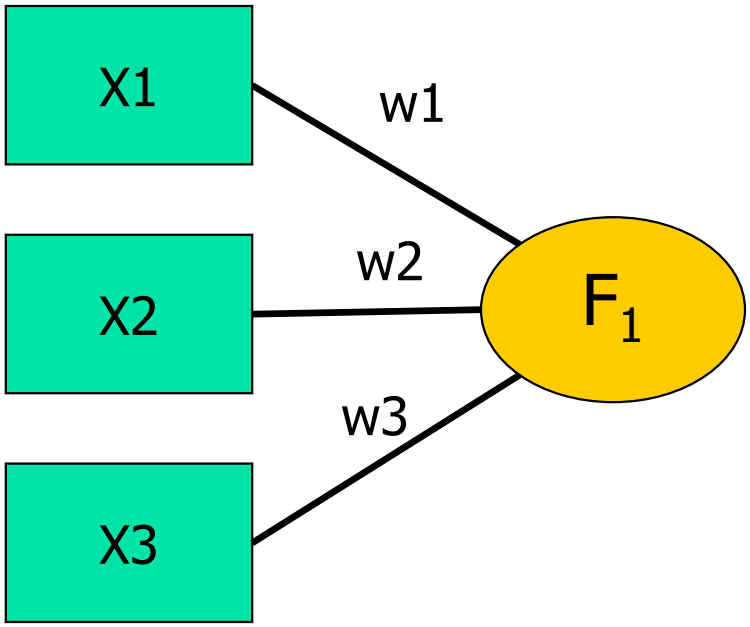
- Principal Components Analysis (PCA) is a widely used method for reducing a set of *P* continuous variables to a set of *D* new variables (*D* < *P*).
  - No distinction between DV and predictors
  - Unsupervised learning



Principal Components Analysis

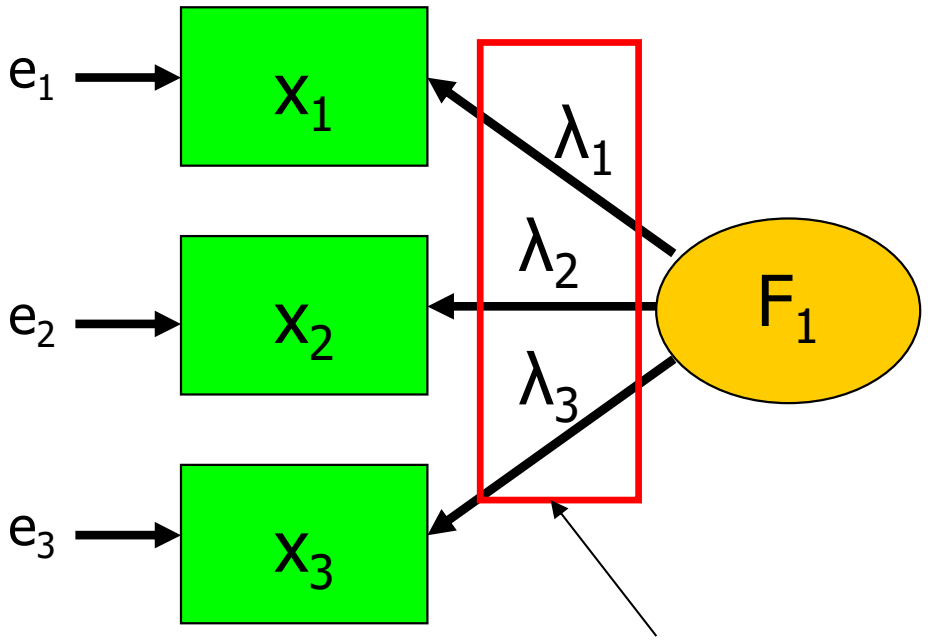
- Requires no distributional assumption of variables (e.g., normality).
- Defines new *D* variables as components (= linear combinations of variables).
- The components are extracted to explain the maximum variance of all variables.

PCA: Basic Concepts



Component: $F_1 = X_1 w_1 + X_2 w_2 + X_3 w_3$

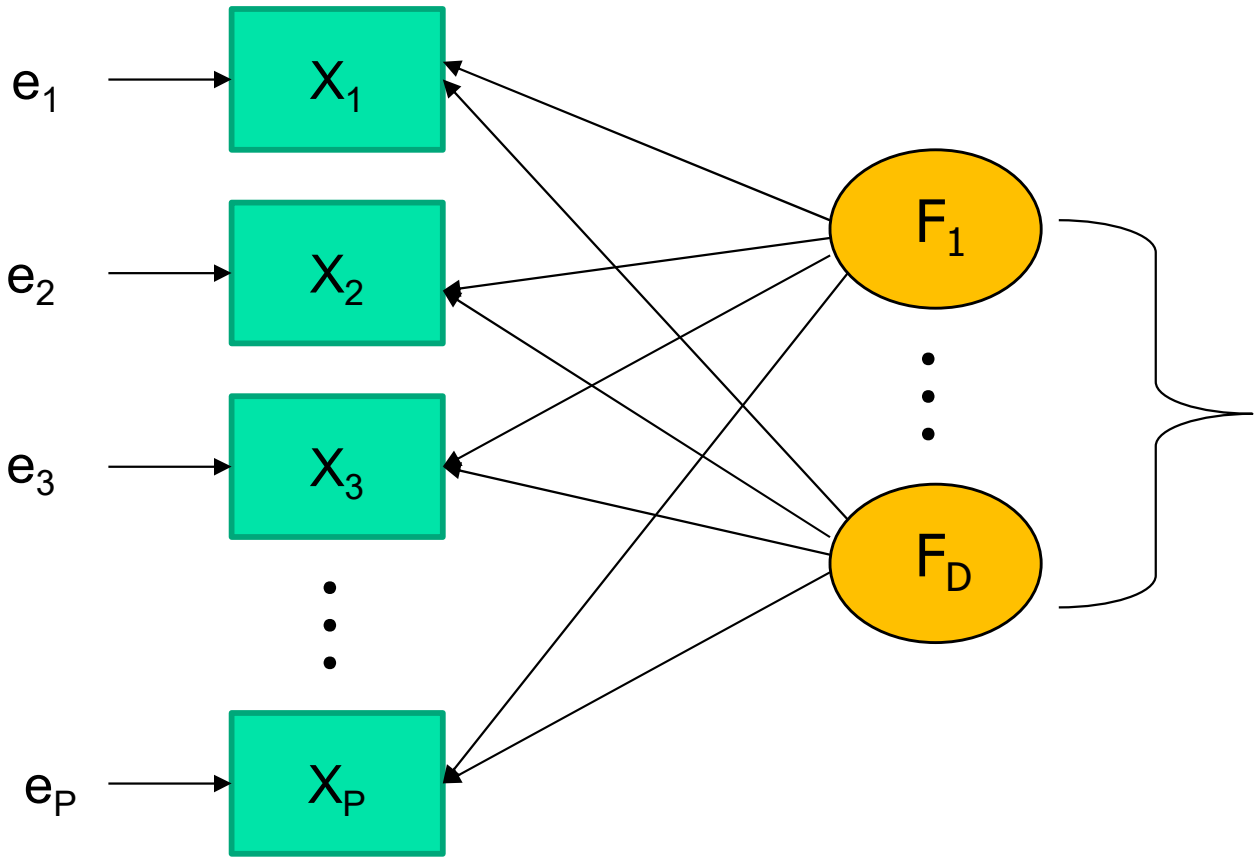
PCA: Basic Concepts



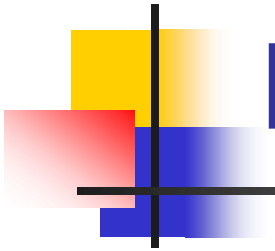
$$X_1 = F_1\lambda_1 + e_1$$
$$X_2 = F_1\lambda_2 + e_2$$
$$X_3 = F_1\lambda_3 + e_3$$

component loadings (λ's): correlations between a component and variables

PCA: Basic Concepts



Orthogonal components:
A component is uncorrelated with the other components.



PCA: Basic Concepts

- In PCA, usually, all variables are first transformed to standard scores (z scores) and then components are extracted from the standardized variables.
  - Z scores: mean = 0 and variance = 1

- The maximum number of components we can extract is equal to the number of variables.
  - The first component explains the most variance; the second represents the component that accounts for most of what is left, and so on.

PCA: Key Terms

- Squared loading: The proportion (%) of the variance of a variable explained by a component.
- Communality (h^2): The sum of squared loadings for a variable for all components.
  - How much variance of the variable is explained by all components? (max = 1)
- The proportion (%) of variance in all variables explained by a single component is the sum of squared loadings for the component divided by the number of variables.

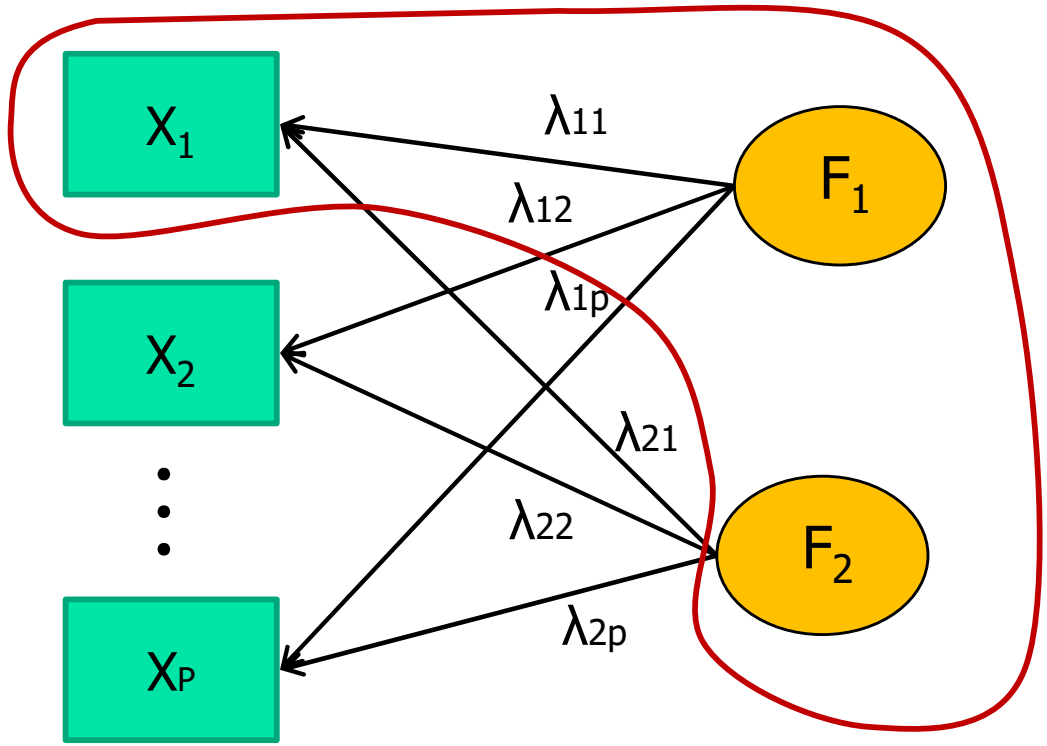
PCA: Communality

$$h_{x1}^2 = \sum_{d=1}^{D} \lambda_{d1}^2 = \lambda_{11}^2 + \lambda_{21}^2$$

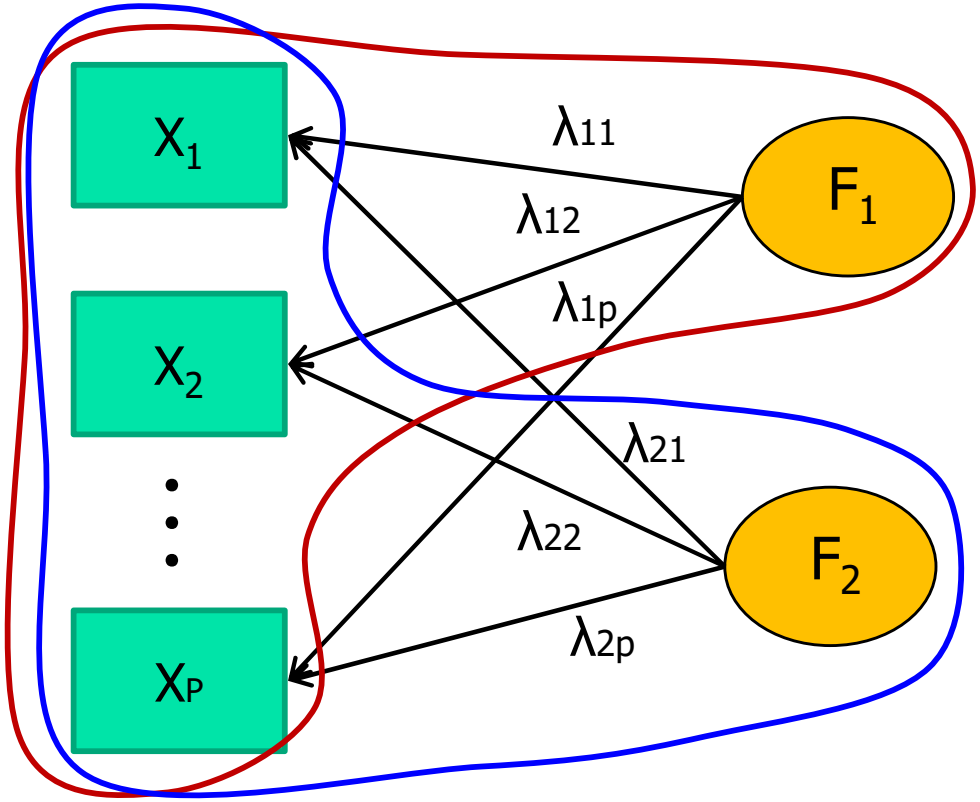
$$h_{x2}^2 = \sum_{d=1}^{D} \lambda_{d2}^2 = \lambda_{12}^2 + \lambda_{22}^2$$

$$\vdots$$

$$h_{xp}^2 = \sum_{d=1}^{D} \lambda_{d1}^2 = \lambda_{1p}^2 + \lambda_{2p}^2$$



PCA: The proportion of variance



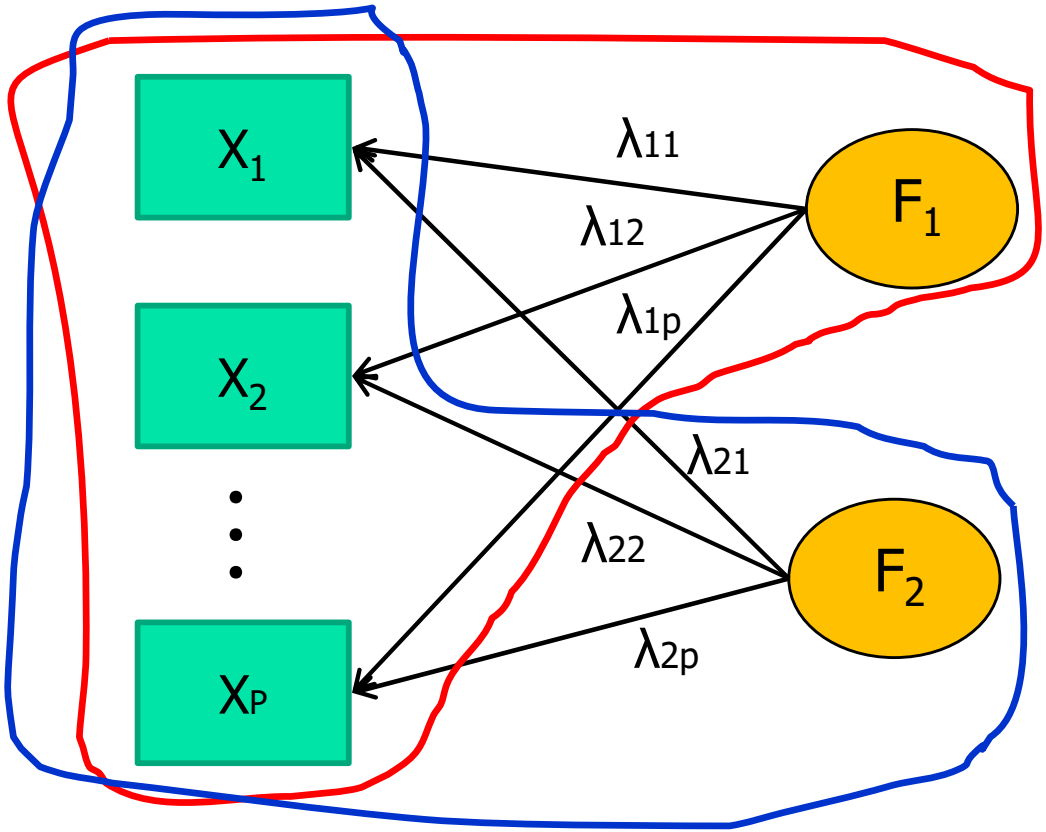
$$PV_{F1} = \sum_{p=1}^{P} \frac{\lambda_{1p}^2}{P} = \frac{\lambda_{11}^2 + \lambda_{12}^2 + \cdots + \lambda_{1P}^2}{P}$$

$$PV_{F2} = \sum_{p=1}^{P} \frac{\lambda_{2p}^2}{P} = \frac{\lambda_{21}^2 + \lambda_{22}^2 + \cdots + \lambda_{2P}^2}{P}$$

PCA: Basic Concepts

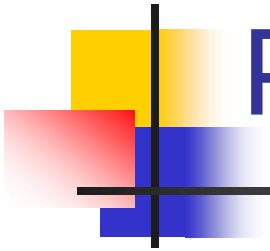
- Eigenvalues: The eigenvalue for a component measures the variance in all the variables, which is accounted for by the component.
  - The total variance = the number of variables.
    - Eigenvalue ≈ the number of variables which the component represents
  - If a component has a low eigenvalue, then it is contributing little to the explanation of variances in the variables.
  - A component's eigenvalue may be computed as the sum of its squared loadings for all the variables.

PCA: Eigenvalues



$$EV_{F1} = \sum_{p=1}^{P} \lambda_{1i}^2 = \lambda_{11}^2 + \lambda_{12}^2 + \cdots + \lambda_{1P}^2$$

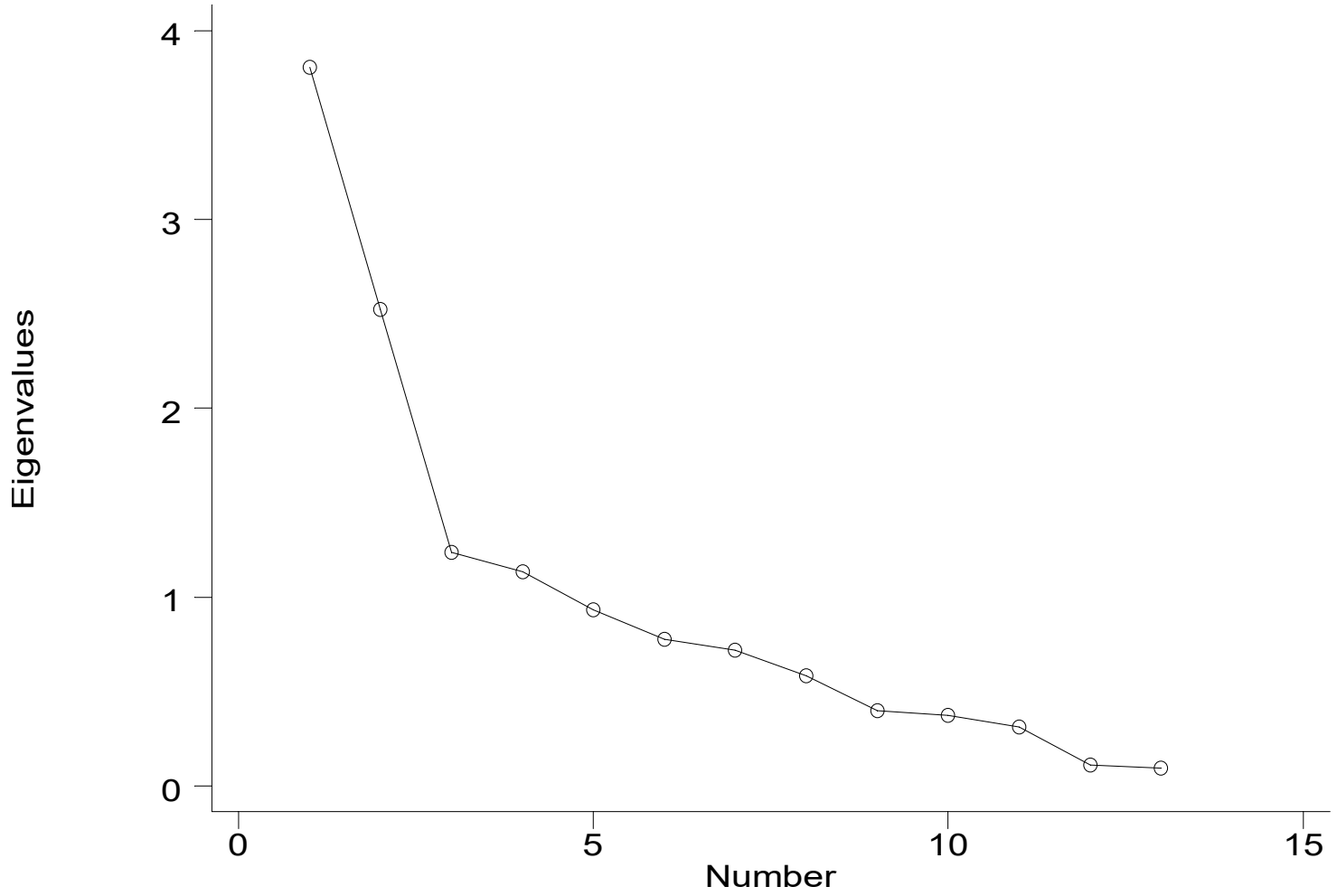
$$EV_{F2} = \sum_{p=1}^{P} \lambda_{2p}^2 = \lambda_{21}^2 + \lambda_{22}^2 + \cdots + \lambda_{2P}^2$$



PCA: Technical Issues

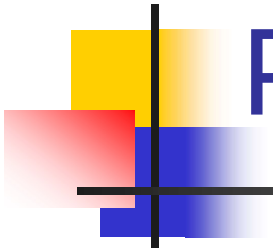
- How many components to extract?
  - Eigenvalue criteria (Kaiser's rule) (Kaiser, 1959):
    - Retain only components with eigenvalues > 1
  - The scree test (Cattell, 1966): The plot of eigenvalues against components. An appropriate number of components may be chosen as an elbow point of the decreasing trajectory of eigenvalues.

Scree plot of eigenvalues



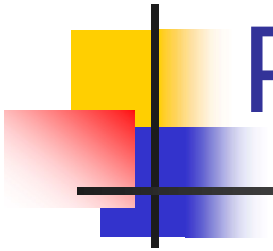
PCA: Technical Issues

- How many components to extract?
  - The Kaiser rule appears to be accurate
    - When the number of variables < 30 and the communalities for all variables > .70, OR
    - When N > 250 and the mean communality ≥ .60.
  - The scree test seems to perform well when N > 200 with reasonably large communalities.



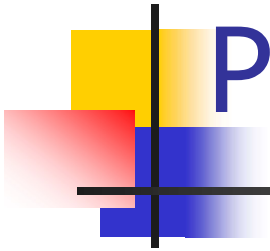
PCA: Technical Issues

- How many components to extract?
  - Interpretability: It is always important to check whether there is much to be said for extracted components retained by other criteria.
    - For example, if only one variable is highly correlated (loaded) with a component, the component will be poorly defined.



PCA: Technical Issues

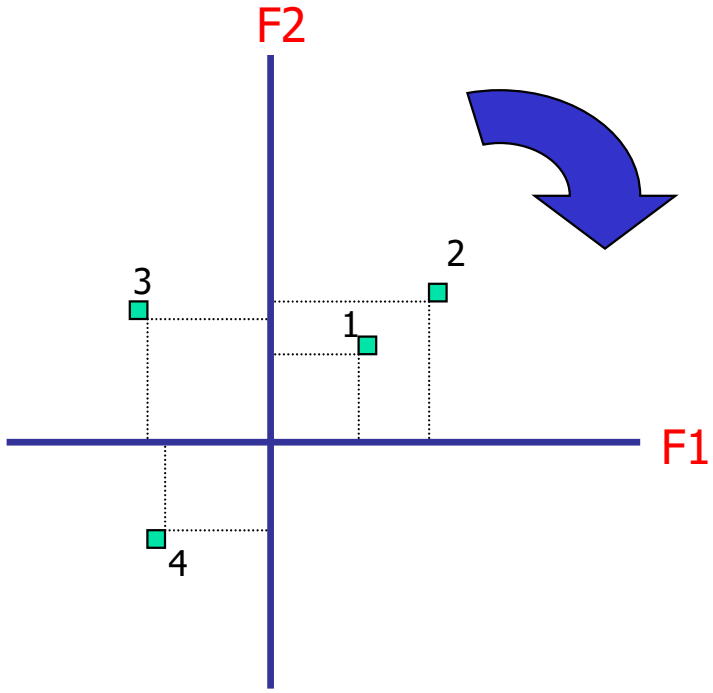
- How to interpret components?
  - Look at "Rotated" component loadings
    - Loadings that are:
      - more than |.5| are good
      - between |.4| and |.5| are ok
      - between |.3| and |.4| are barely tolerable
      - Less than |.3| -- forget about it
    - Start with the variable with the highest loading

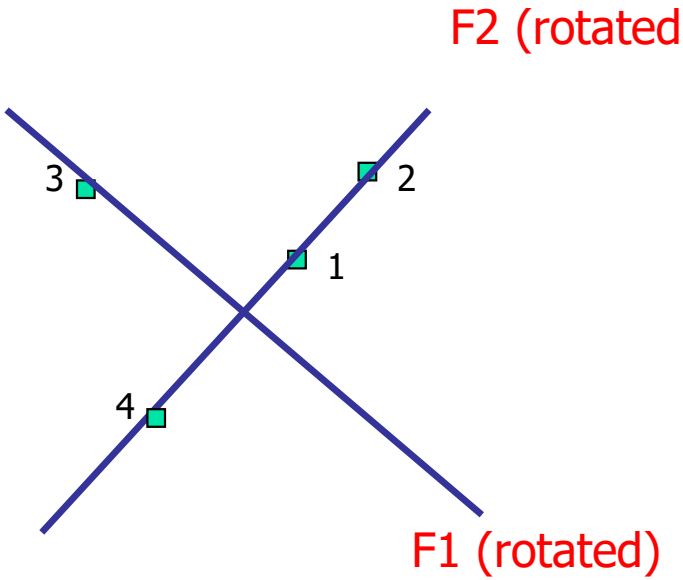


PCA: Technical Issues

- Rotation?
  - The loadings could be plot in a scatterplot, with each variable represented as a point.
  - The axis of this plot could be rotated in any direction without changing the *relative* locations of the points to each other.
  - The goal of rotation is to obtain a clear pattern of loadings – easy interpretation of components
  - Rotation does not apply when only a single component is retained.

Rotating principal components





	F1	F2
X1	0.5	0.5
X2	0.8	0.8
X3	-0.7	0.7
X4	-0.5	-0.5

	F1 (rotated)	F2 (rotated)
X1	0	0.6
X2	0	0.9
X3	-0.9	0
X4	0	-0.9

PCA: Rotation

- Two ways of rotation
  - Orthogonal rotation: components are still uncorrelated.
    - Easy to interpret results.
    - Yet, maybe less realistic in many cases.
    - 'Varimax' rotation is the most common choice
  - Oblique rotation: components are correlated.
    - Maybe more realistic.
    - But, difficult to interpret results.
    - Users set (delta (δ) to specify how much correlation is allowed between components.
    - Refer to Tabachnick & Fidell (2007, Chap. 13)

PCA: Varimax Rotation

- We want to achieve a 'simple structure' that the squared loadings of all variables for a component is close either one or zero.

- The varimax rotation (Kaiser, 1958) attempts to do this by maximizing the sum of the variance of the squared loadings of all variables within a component.
  - The maximum variance is achieved when the values of squared loadings are close to 1 or 0 for a component; by definition of the communality, when squared loadings approach 1 for a component, they are toward zeros for other components.
  - Varimax = *variance maximizing*

Varimax Rotation: Intuition

Squared Loadings

max $h^2 = 1$

	F1	F2	h^2
x1	λ^2_{11}	λ^2_{21}	$\lambda^2_{11} + \lambda^2_{21}$
x2	λ^2_{12}	λ^2_{22}	$\lambda^2_{12} + \lambda^2_{22}$
x3	λ^2_{13}	λ^2_{23}	$\lambda^2_{13} + \lambda^2_{23}$
x4	λ^2_{14}	λ^2_{24}	$\lambda^2_{14} + \lambda^2_{24}$

variance of squared loadings of all variables for each component

→

Maximize the sum of this variance across components

	F1	F2	h^2
x1	1	0	$\lambda^2_{11} + \lambda^2_{21}$
x2	1	0	$\lambda^2_{12} + \lambda^2_{22}$
x3	0	1	$\lambda^2_{13} + \lambda^2_{23}$
x4	0	1	$\lambda^2_{14} + \lambda^2_{24}$

Ideally…

Example: PCA

- Toothpaste attribute rating data (Malhotra, 2004, p. 563; toothpaste attribute rating.csv)
- N = 30 and 6 variables (1 = strongly disagree, 7 = strongly agree)
  - V1: It is important to buy a toothpaste that prevents cavities
  - V2: I like a toothpaste that gives shiny teeth
  - V3: A toothpaste should strengthen your gums
  - V4: I prefer a toothpaste that freshens breath
  - V5: Prevention of tooth decay is not an important benefit offered by a toothpaste
  - V6: The most important consideration in buying a toothpaste is attractive teeth

Example: Toothpaste attribute rating data (Malhotra, 2004, p. 563)

Component Loadings

	PC1	PC2	Uniqueness
v1	0.962	-0.031	0.074
v5	-0.934	-0.079	0.122
v3	0.933	-0.151	0.106
v4	-0.094	0.855	0.261
v6	0.088	0.885	0.210
v2	-0.053	0.848	0.277

Note. Applied rotation method is varimax.

Example: Toothpaste attribute rating data (Malhotra, 2004, p. 563)

Component Characteristics

	Unrotated solution			Rotated solution		
	Eigenvalue	Proportion var.	Cumulative	Variance explained	Proportion var.	Cumulative
Component 1	2.731	0.455	0.455	2.687	0.448	0.448
Component 2	2.218	0.370	0.825	2.263	0.377	0.825

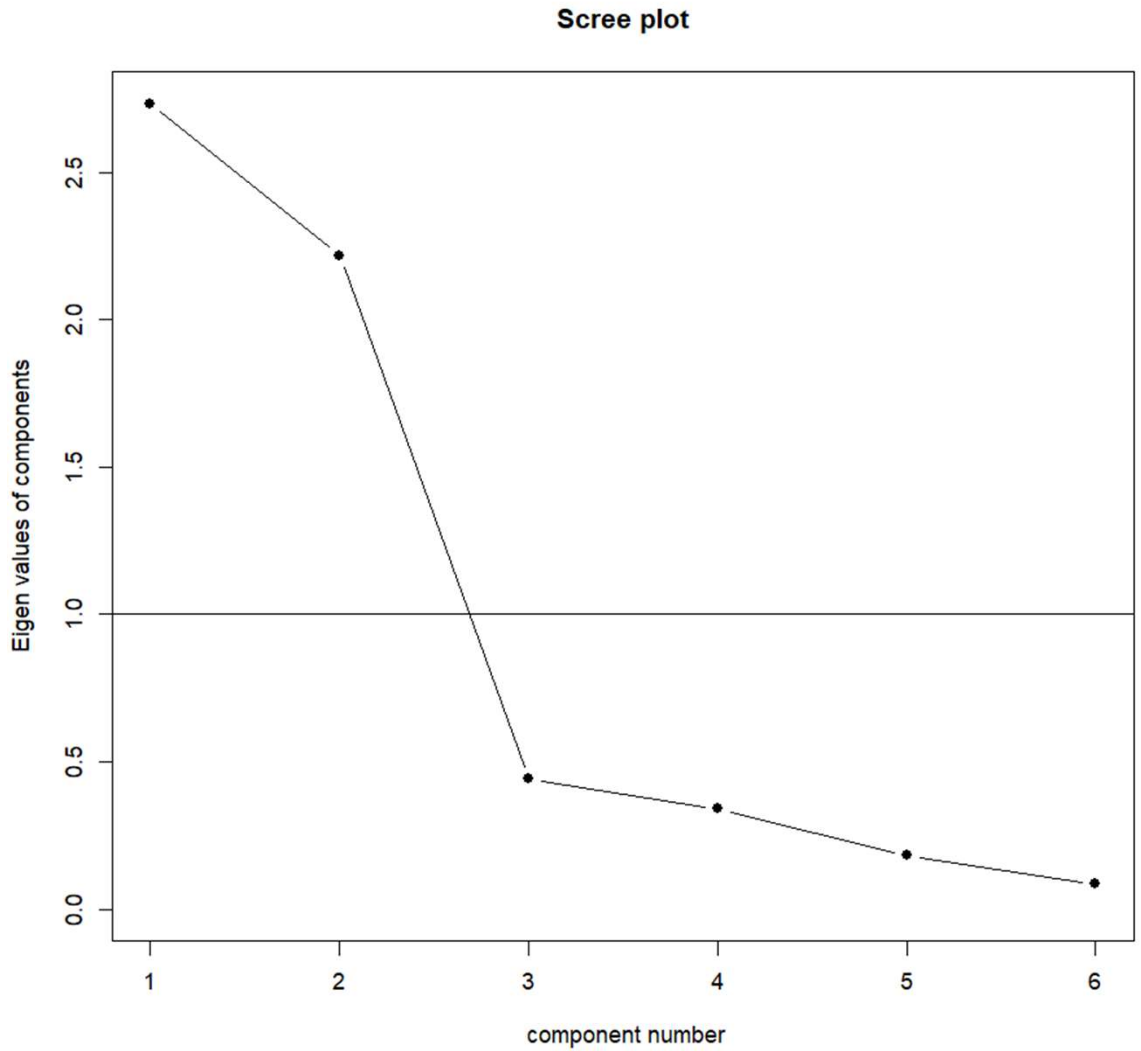
Example: Toothpaste attribute rating data (Malhotra, 2004, p. 563)

```
Principal Components Analysis

Call: principal(r = mydata2, nfactors = 2, rotate = "none")
Standardized loadings (pattern matrix) based upon correlation matrix
     PC1   PC2   h2    u2 com
v1  0.93  0.25 0.93 0.074 1.1
v2 -0.30  0.80 0.72 0.277 1.3
v3  0.94  0.13 0.89 0.106 1.0
v4 -0.34  0.79 0.74 0.261 1.4
v5 -0.87 -0.35 0.88 0.122 1.3
v6 -0.18  0.87 0.79 0.210 1.1

                       PC1  PC2
SS loadings           2.73 2.22
Proportion Var        0.46 0.37
Cumulative Var        0.46 0.82
Proportion Explained  0.55 0.45
Cumulative Proportion 0.55 1.00

Mean item complexity =  1.2
Test of the hypothesis that 2 components are sufficient.

The root mean square of the residuals (RMSR) is  0.07
 with the empirical chi square  3.94  with prob <  0.41

Fit based upon off diagonal values = 0.98
```

Example: Toothpaste attribute rating data (Malhotra, 2004, p. 563)

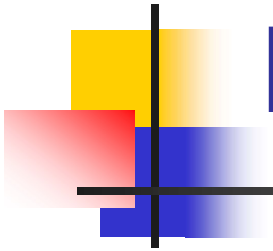
```
Principal Components Analysis
Call: principal(r = mydata2, nfactors = 2, rotate = "varimax",
    cor = "cor")
Standardized loadings (pattern matrix) based upon correlation matrix
     RC1   RC2   h2    u2 com
v1  0.96 -0.03 0.93 0.074 1.0
v2 -0.05  0.85 0.72 0.277 1.0
v3  0.93 -0.15 0.89 0.106 1.1
v4 -0.09  0.85 0.74 0.261 1.0
v5 -0.93 -0.08 0.88 0.122 1.0
v6  0.09  0.88 0.79 0.210 1.0

                       RC1  RC2
SS loadings           2.69 2.26
Proportion Var        0.45 0.38
Cumulative Var        0.45 0.82
Proportion Explained  0.54 0.46
Cumulative Proportion 0.54 1.00

Mean item complexity =  1
Test of the hypothesis that 2 components are sufficient.

The root mean square of the residuals (RMSR) is  0.07
 with the empirical chi square  3.94  with prob <  0.41

Fit based upon off diagonal values = 0.98
```

Example: Toothpaste attribute rating data (Malhotra, 2004, p. 563)



What shall these components be called?

Component 1	Component 2
Prevents Cavities (V1)	Shiny Teeth (V2)
Strengthen Gums (V3)	Freshens Breath (V4)
Tooth Decay Important (V5)	Attractive Teeth (V6)



PCA: Other Consideration

- Component Scores
  - The scores of components for each individual or observation.
  - These scores can be used for subsequent analyses (regression, clustering, etc.) instead of original variables.

*toothpaste attribute rating.sav [DataSet1] - IBM SPSS Statistics Data Editor

File Edit View Data Transform Analyze Graphs Utilities Extensions Window Help

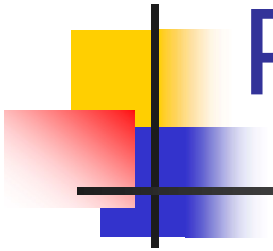
Visible: 8 of 8 Variables

	v1	v2	v3	v4	v5	v6	F1	F2	var	var	var	var	var	var	var	var
1	7	3	6	4	2	4	1.182	.048								
2	1	3	2	4	5	4	-1.014	-.661								
3	6	2	7	4	1	3	1.470	-.442								
4	4	5	4	6	2	5	-.055	1.140								
5	1	2	2	3	6	2	-.917	-1.829								
6	6	3	6	4	2	4	1.010	-.010								
7	5	3	6	3	4	3	.643	-.775								
8	6	4	7	4	1	4	1.264	.363								
9	3	4	2	3	6	3	-.781	-.909								
10	2	6	2	6	7	6	-1.692	1.096								
11	6	4	7	3	2	3	1.234	-.261								
12	2	3	1	4	5	4	-1.009	-.633								
13	7	2	6	4	1	3	1.475	-.413								
14	4	6	4	5	3	6	-.257	1.341								
15	1	3	2	2	6	4	-.999	-1.262								
16	6	4	6	3	3	4	.854	-.091								
17	5	3	6	3	3	4	.763	-.409								
18	7	3	7	4	1	4	1.515	.159								
19	2	4	3	3	6	3	-.785	-.938								
20	3	5	3	6	4	6	-.773	1.170								
21	1	3	2	3	5	3	-.877	-1.203								
22	5	4	5	4	2	4	.592	.165								
23	2	2	1	5	4	4	-.854	-.552								
24	4	6	4	6	4	7	-.562	1.799								
25	6	5	4	2	1	4	.866	.020								
26	3	5	4	6	4	7	-.653	1.481								
27	4	4	7	2	2	5	.889	-.071								
28	3	7	2	6	4	3	-.961	.817								
29	4	6	3	7	2	7	-.486	2.196								
30	2	3	2	4	7	2	-1.083	-1.334								

Data View Variable View

IBM SPSS Statistics Processor is ready Unicode:ON

Principal Components Regression



- Principal components regression (PCR) involves constructing the first D principal components of P predictors and then using these components as the predictors in a linear regression model that is fit using least squares.

- The key idea is that often a smaller number of principal components suffice to explain most of the variability in predictors, as well as the relationship with the response.

Principal Components Regression

- In PCR, the number of principal components is typically chosen by cross validation.

- As in PCA, it is generally recommended to standardize each predictor in PCR.

Partial Least Squares Regression

- In PCR, there is no guarantee that the principal components of predictors will best explain/predict the response as well. This is because the PCs are obtained without reference to the response.

- Partial least squares regression (PLSR) is a supervised alternative to PCR.

Partial Least Squares Regression

- Like PCR, PLSR is a dimension reduction method, which obtains D components of original predictors that are linear combinations of the predictors.
- Unlike PCR, PLSR obtains these components in such a way that they explain the variances of both response and predictors.
- Cross validation can be used to determine the number of components.

Partial Least Squares Regression

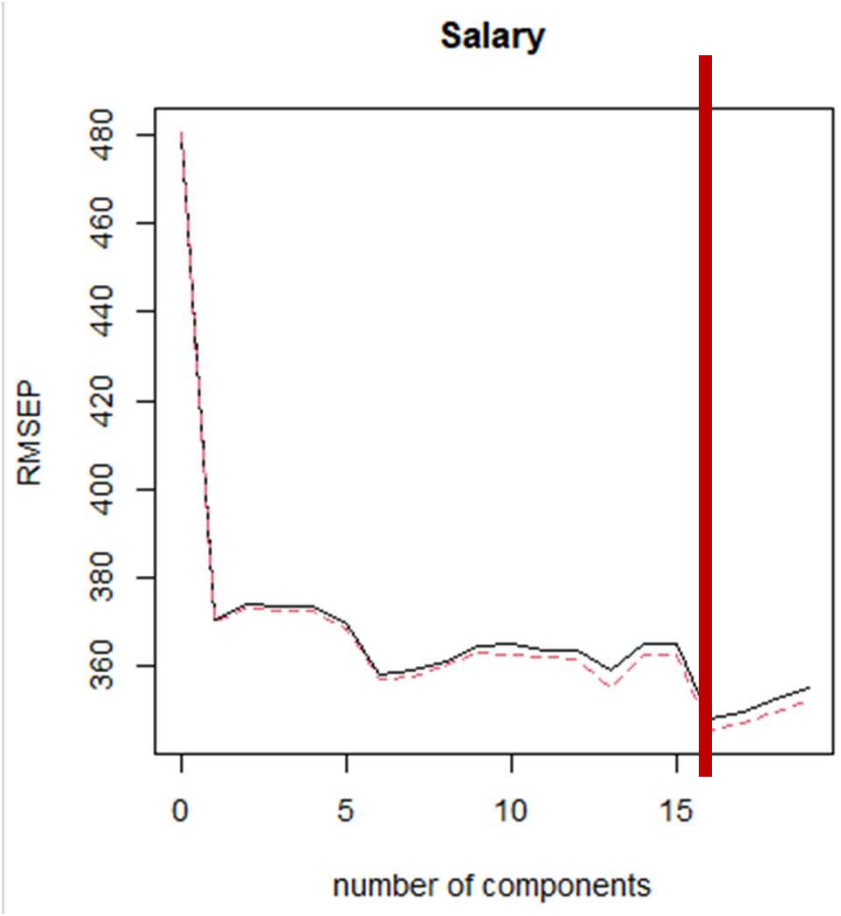
- Both response and predictors are standardized before performing PLSR.
- In practice, PLSR often preforms no better than ridge regression or PCR.

Example: PCR

Hitters_training.csv

```
Data:       X dimension: 184 19
            Y dimension: 184 1
Fit method: svdpc
Number of components considered: 19

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps  9 comps
CV           480.2    370.8    374.0    373.6    373.7    369.5    358.3    359.1    361.2    364.8
adjCV        480.2    370.2    373.2    372.9    372.6    368.8    357.2    358.0    360.0    363.3
       10 comps  11 comps  12 comps  13 comps  14 comps  15 comps  16 comps  17 comps  18 comps  19 comps
CV        365.2     363.9     363.7     359.0     365.1     365.1     348.2     349.8     352.7     355.1
adjCV     362.6     362.1     361.9     355.3     362.9     362.8     345.9     347.3     349.9     352.1

TRAINING: % variance explained
        1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps  9 comps  10 comps
X         41.76    61.14    70.95    80.12    85.24    89.68    92.75    95.34    96.59     97.47
Salary    43.01    43.29    43.89    44.93    46.18    49.98    49.98    50.07    50.24     52.62
        11 comps  12 comps  13 comps  14 comps  15 comps  16 comps  17 comps  18 comps  19 comps
X          98.25     98.84     99.21     99.56     99.79     99.91     99.97     99.99    100.00
Salary     52.73     53.06     54.68     54.69     55.40     59.29     59.74     60.41     60.45
```

Example: PCR

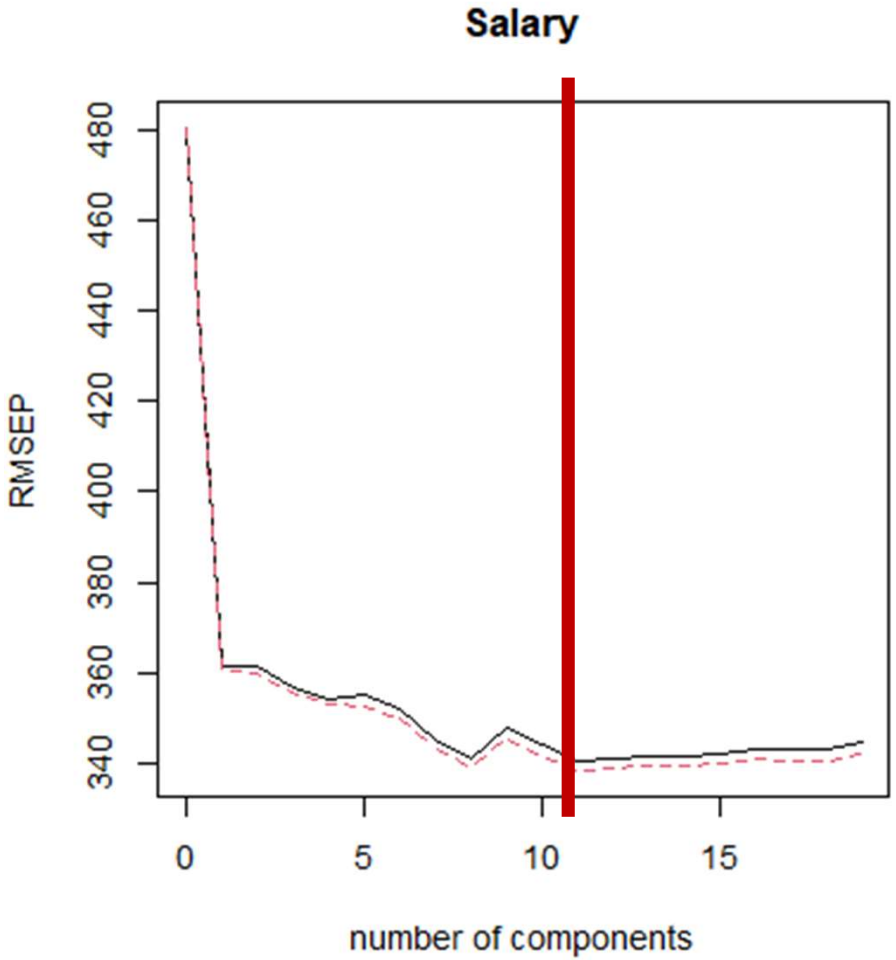

Salary
RMSEP
360
380
400
420
440
460
480
0
5
10
15
number of components

Example: PLSR

Hitters_training.csv

```
Data:       X dimension: 184 19
            Y dimension: 184 1
Fit method: kernelpls
Number of components considered: 19

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps  9 comps
CV           480.2    361.3    361.6    356.5    353.9    355.2      352    345.2    340.8    348.0
adjCV        480.2    361.0    360.0    355.5    353.0    352.7      350    343.7    338.8    345.5
       10 comps  11 comps  12 comps  13 comps  14 comps  15 comps  16 comps  17 comps  18 comps  19 comps
CV        344.0     340.5     341.1     341.7     341.7     342.1     343.1     342.9     343.0     344.4
adjCV     341.6     338.4     339.1     339.5     339.5     340.0     340.8     340.7     340.7     342.1

TRAINING: % variance explained
        1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps  9 comps  10 comps
X         41.61    50.52    64.91    74.15    76.41    83.83    88.68    91.14    93.32     94.28
Salary    45.13    50.47    51.92    53.21    56.26    56.74    57.52    58.76    59.28     59.76
       11 comps  12 comps  13 comps  14 comps  15 comps  16 comps  17 comps  18 comps  19 comps
X         95.76     97.91     98.43     98.73     99.26     99.59     99.64     99.97    100.00
Salary    59.99     60.08     60.18     60.32     60.37     60.42     60.44     60.44     60.45
```

Example: PLSR



Example: PCR vs PLSR

Hitters_test.csv

MSE_PCR: 123931.6

MSE_PLSR: 119524.2

MSE_Basic: 116358.2