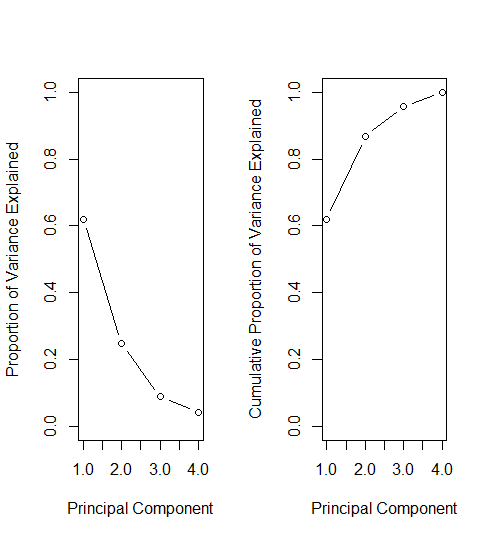
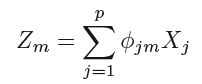
# PCA - Principal Component Analysis

* **What is Dimension Reduction?** – reducing the number of predictors in a data set to a more manageable amount (*m*) while preserving as much of the information in the original *x*’s as possible in hopes that the resulting decrease in variance will more than make up for the increase in the (squared) bias.
* **What does PCA do?** – reduces the dimensions/variance of a dataset while retaining as much model variability as possible. It does this in an unsupervised way (pays no attention to the *y*’s, assumes variable in *x*’s is a surrogate for ability to predict *y*’s)
* **When do you use PCA?** – In any setting where regression is being performed and there is reason to believe that variance is high. This occurs, for example, when you have a *large set of correlated predictors* with small *n*.
* **What to keep in mind** – the x’s need to be scaled (the PCR function will do this, but only if requested)
* **1st Principal Component** – The vector along which the data varies the most
* 2nd Principal component is that in which the data varies next most, orthogonal to 1st (uncorrelated) and so forth
  + Each additional PC adds less explanation of variation than the previous, summing to 100% when all *p* principal components are used.





* **PC Scores** -
  + Where *Zm*is the principal component score from a non-negative eigenvector *ɸ*m from the covariance matrix of x
  + *ɸ*m is the mth principal component and *Z*m+1 is such that (the *Z*’s are uncorrelated, i.e. orthogonal).
* **R (the important stuff)**

# Build model with test set and plot results  
pcr.fit <- pcr(Salary~., data=Hitters, subset=train,   
 scale=TRUE, validation="CV")  
validationplot(pcr.fit, val.type="MSEP")  
# Choose best # of components and get test MSE   
pcr.pred <- predict(pcr.fit, x[test, ], ncomp=7)  
mean((pcr.pred-y.test)^2)  
# Rebuid with full sample and best # of components  
pcr.fit <- pcr(y ~ x, scale=TRUE, ncomp=7