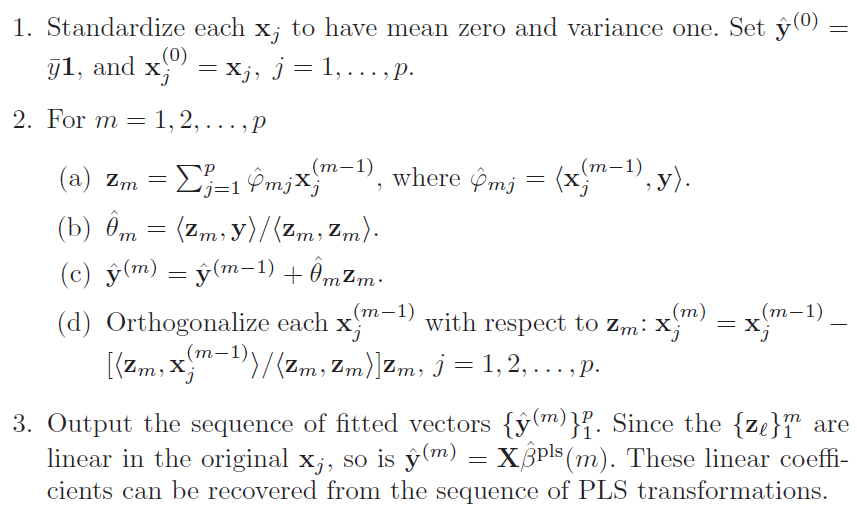
# PLS – Partial Least Squares Regression

* **Recall: 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 PLS do?** – reduces the dimensions of a dataset by considering both the x’s and the y’s (i.e. it’s a supervised approach, whereas PCR is unsupervised).
* **When do you use PLS?** – 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 PLSR function will do this, but only if requested)
* **1st PLS Direction** – The vector of *p* simple regression coefficients.
* 2nd PLS Direction is derived by considering the part of the *y*’s that remains unexplained by the previous direction (i.e. the residuals), and so forth
  + Each additional PLS direction adds less explanation of variation than the previous, summing to 100% when all *p* PLS directions are used.
* The PLS regression algorithm is as follows:



* **PLS Scores** -
  + As with PCR, the PLS directions Zm+1 are such that (the Z’s are uncorrelated, i.e. orthogonal).
* **R (the important stuff)**

# Build model with test set and plot results  
pls.fit <- plsr(Salary~., data=Hitters, subset=train,   
 scale=TRUE, validation="CV")  
validationplot(pls.fit, val.type="MSEP")  
# Choose best # of components and get test MSE   
pls.pred <- predict(pls.fit, x[test, ], ncomp=4)  
mean((pls.pred-y.test)^2)  
# Rebuid with full sample and best # of components  
pls.fit.final <- plsr(y ~ x, scale=TRUE, ncomp=4