Lecture 27: PCR and PLS

Big Data and Machine Learning for Applied Economics
Econ 4676

Ignacio Sarmiento-Barbieri

Universidad de los Andes

November 16, 2021

Agenda

Recap: PCA serves as a dimentionality reduction technique

- Eigenvalues and eigenvectors associated to the covariance matrix of the data $X_{n \times p}$, $V(X) = S$

$$\lambda_1 = \delta_1 S \delta_1' \tag{1}$$

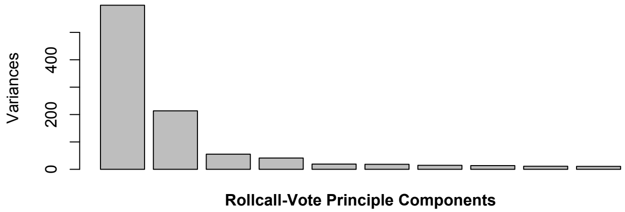
- We get an "index" $f_s = \delta_s X$: 'loadings' often suggest that a factor works as a 'index' of a group of variables.
- Important to scale the variables (sensible to units)
- Different criteria for choosing the number of PC
  - Visual examination of screeplot
  - Kaiser criterion.
  - Proportion of variance explained.

Factor Interpretation: Example

- **Congress and Roll Call Voting**
  - Votes in which names and positions are recorded are called 'roll calls'.
  - The site `voteview.com` archives vote records and the R package `pscl` has tools for this data.
  - 445 members in the last US House (the 111^{th})
  - 1647 votes: nea = -1, yea=+1, missing = 0.
  - This leads to a large matrix of observations that can probably be reduced to simple factors (party).

Factor Interpretation

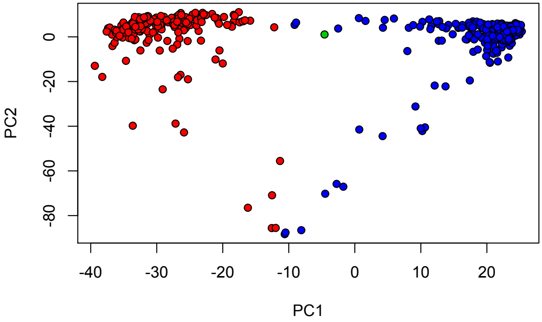
- Vote components in the $\mathbf{111}^{th}$ house
- Each PC is $f_s = \delta_s X$



- Huge drop in variance from 1^{st} to 2^{nd} and 2^{nd} to 3^{rd} PC.
- Poli-Sci holds that PC1 is usually enough to explain congress.
  2nd component has been important twice: 1860's and 1960's.

Factor Interpretation

- Top two PC directions in the **111**th house



- Republicans in red and Democrats in blue:
  - Clear separation on the first principal component.
  - The second component looks orthogonal to party.

Factor Interpretation

```
## Far right (very conservative)
> sort(votepc[,1])
    BROUN (R GA-10)       FLAKE (R AZ-6)   HENSARLIN (R TX-5)
        -39.3739409          -38.2506713          -37.5870597

## Far left (very liberal)
> sort(votepc[,1], decreasing=TRUE)
   EDWARDS (D MD-4)   PRICE (D NC-4)    MATSUI (D CA-5)
        25.2915083        25.1591151         25.1248117

## social issues?  immigration?  no clear pattern
> sort(votepc[,2])
    SOLIS (D CA-32) GILLIBRAND (D NY-20)      PELOSI (D CA-8)
       -88.31350926         -87.58871687         -86.53585568
  STUTZMAN (R IN-3)       REED (R NY-29)      GRAVES (R GA-9)
       -85.59217310         -85.53636319         -76.49658108
```

- PC1 is easy to read, PC2 is ambiguous (is it even meaningful?)

Factor Interpretation

- Look at the largest loadings in δ_2 to discern an interpretation.

```
> loadings[order(abs(loadings[,2]), decreasing=TRUE)[1:5],2]
 Vote.1146   Vote.658  Vote.1090  Vote.1104  Vote.1149
0.05605862 0.05461947 0.05300806 0.05168382 0.05155729
```

- These votes all correspond to near-unanimous symbolic action.
- For example, 429 legislators voted for resolution 1146:
  'Supporting the goals and ideals of a Cold War Veterans Day'
  If you didn't vote for this, you weren't in the house.
- Mystery Solved: the second PC is just attendance!

```
> sort(rowSums(votes==0), decreasing=TRUE)
     SOLIS (D CA-32) GILLIBRAND (D NY-20)       REED (R NY-29)
                1628                 1619                 1562
   STUTZMAN (R IN-3)      PELOSI (D CA-8)      GRAVES (R GA-9)
                1557                 1541                 1340
```

Principal Component Regression (PCR)

- Now that you've learned how to fit factor models, what are they good for?
- In some settings, as in the previous political science example, the factors themselves have clear meaning and can be useful in their own right for understanding complex systems.
- More commonly, unfortunately, the factors are of dubious origin or interpretation.
- However, they can still be useful as inputs to a regression system.
- Indeed, this is the primary practical function for PCA, as the first stage of principal components regression (PCR).

Principal Component Regression (PCR)

- The concept of PCR is simple:
  - Instead of doing $y \to X$,
  - Use a lower-dimension set of principal components as covariates.
- This is a fruitful strategy for a few reasons:
  - PCA reduces dimension, which is usually good.
  - The PCs are independent, so you have no multicollinearity and the final regression is easy to fit.
  - You might have far more unlabeled x_i than labeled (x_i, y_i) pairs. This last point is especially powerful.
  - You can use unsupervised learning (PCA) on a massive bank of unlabeled data and use the results to reduce dimension and facilitate supervised learning on a smaller set of labeled observations.

Principal Component Regression (PCR)

- The 2-stage algorithm is straightforward.
- For example,

```
mypca <- prcomp(X, scale=TRUE)

z <- predict(mypca)[,1:K]

reg <- glm(y~., data=as.data.frame(z))
```

Principal Component Regression (PCR)

- The disadvantage of PCR is that PCA will be driven by the dominant sources of variation in X.
- If the response is connected to these dominant sources of variation, PCR works well.
- If it is more of a "needle in the haystack response," driven by a small number of inputs, then PCR will not work well.
- In practice, you do not know what scenario you are in until you try both PCR and, say, a lasso regression on the raw X inputs.

Principal Component Regression (PCR)

- How many PC do we use?
  - When PCA was used as a dimensionality reduction tool *per se* we had some guidelines...
- Should we do the same here?

Principal Component Regression (PCR)

- How many PC do we use?
  - When PCA was used as a dimensionality reduction tool *per se* we had some guidelines...
- Should we do the same here?
- In PCR the approach is slightly different
  - Construct $min(n-1, p)$ components
  - Use K fold crossvalidation adding 1 PC at a time
  - Choose the model with the lowest out of sample MSE
- Because the PCs are ordered (by their variance) and independent, this works better than subset selection on the raw dimensions of X_i.

Principal Component Regression (PCR)

- An alternative mechanism is run a lasso on the full set of PCs (works best in practice).
- This procedure makes it easy to incorporate other information in addition to the PCs.
- For example, one tactic that works well in practice is to put both PC and Xs into the lasso model matrix.
  - This then allows the regression to make use of the underlying factor structure in X and still pick up individual X_j signals that are related to y.
  - This hybrid strategy is a solution to the disadvantage disadvantage of PCR mentioned earlier—that it will only pick up dominant sources of variation in X.

Principal Component Regression (PCR)

Summary of the steps

- Given a sample of regression input observations x_i, accompanied by output labels y_i for some subset of these observations:

  1. Fit PCA on the full set of X inputs to obtain PC of length $min(n-1, p)$.
  2. For the labeled subset, run a lasso regression for y on f (PC).
     - Alternatively, regress y on f and Xs to allow simultaneous selection between PCs and raw inputs.
  3. To predict for a new X_{new}, use the rotations from step 1 to get $f = \phi X_{new}$ and then feed these scores into the regression fit from step 2.

PCR Example: TV Shows

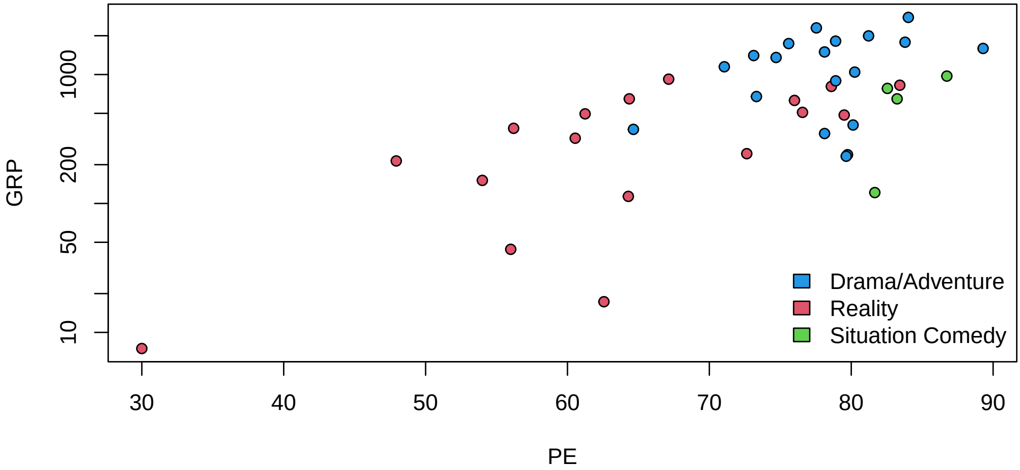
- We have data about TV shows

```r
shows <- read.csv("nbc_showdetails.csv", row.names=1)
```

```
## Rows: 40
## Columns: 5
## $ Network  <chr> "HGTV", "LIFE", "BRAVO", "FOOD", "TLC", "LIFE", "BRAVO", "MTV~
## $ PE       <dbl> 54.0000, 64.6479, 78.5980, 62.5703, 56.0000, 56.2056, 83.4243~
## $ GRP      <dbl> 151.0, 375.5, 808.5, 17.3, 44.1, 382.6, 826.2, 7.5, 320.8, 21~
## $ Genre    <fct> Reality, Drama/Adventure, Reality, Reality, Reality, Reality,~
## $ Duration <int> 30, 60, 60, 30, 60, 60, 60, 30, 30, 30, 60, 30, 60, 60, 60, 6~
```

- Classic measures of broadcast marketability are ratings. Specifically, gross ratings points (GRP) provide an estimated count of total viewership.
- In this data we also track the projected engagement (PE) as a more subtle measure of audience attention.

PCR Example: TV Shows



PCR Example: TV Shows

- We have a survey data that include 6241 views and 20 questions for 40 shows. There are two types of questions in the survey. Both ask you the degree to which you agree with a statement.

```r
survey <- read.csv("nbc_pilotsurvey.csv", as.is=TRUE)
```

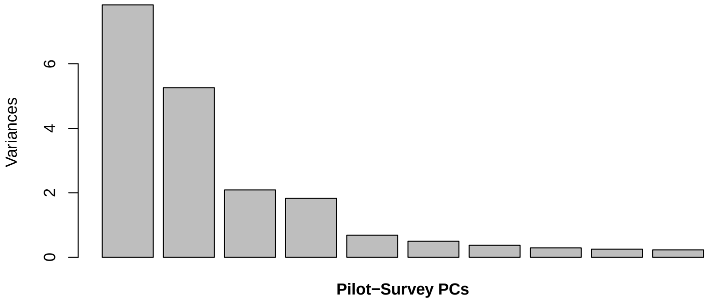
```
## Rows: 6,241
## Columns: 22
## $ Viewer        <int> 71, 71, 71, 71, 71, 73, 73, 73, 73, 73, 74, 74, 74, 76~
## $ Show          <fct> Iron Chef America, Trading Spaces: All Stars, House Hu~
## $ Q1_Attentive  <int> 3, 4, 4, 4, 4, 2, 4, 3, 2, 5, 4, 4, 4, 3, 3, 4, 5, 4, ~
## $ Q1_Excited    <int> 4, 4, 4, 3, 4, 4, 5, 3, 3, 5, 4, 2, 3, 3, 3, 4, 5, 4, ~
## $ Q1_Happy      <int> 4, 3, 4, 3, 3, 2, 4, 3, 4, 5, 5, 4, 3, 3, 3, 3, 5, 5, ~
## $ Q1_Engaged    <int> 3, 4, 5, 3, 4, 4, 5, 3, 4, 5, 5, 4, 4, 3, 3, 3, 5, 4, ~
## $ Q1_Curious    <int> 5, 5, 5, 4, 4, 2, 5, 4, 4, 5, 4, 4, 5, 4, 2, 3, 5, 3, ~
## $ Q1_Motivated  <int> 4, 2, 3, 2, 4, 3, 3, 4, 2, 5, 4, 3, 4, 3, 1, 3, 2, 2, ~
## $ Q1_Comforted  <int> 3, 3, 3, 2, 3, 3, 2, 2, 3, 4, 5, 2, 3, 2, 1, 1, 2, 3, ~
## $ Q1_Annoyed    <int> 2, 3, 2, 4, 3, 3, 4, 4, 2, 3, 3, 2, 2, 1, 1, 1, 1, 2, ~
## $ Q1_Indifferent <int> 2, 2, 1, 2, 4, 3, 2, 2, 3, 1, 1, 2, 2, 3, 4, 2, 1, 2, ~
## $ Q2_Relatable  <int> 3, 4, 2, 2, 3, 2, 1, 3, 4, 3, 5, 3, 4, 3, 1, 2, 2, 5, ~
## ...
```

- The hope is that we can build a rule for predicting viewer interest from pilot surveys, thus helping the studios to make better programming decisions.

PCR Example: TV Shows

- It might seem like there is a lot of data here—6241 pilot viewings—but there are only 40 shows and 20 survey questions.
- To relate survey results to show performance, we need to first calculate the average survey question response by show.
- This leads to a 40×20 design matrix X, and we can fit PCA on this design.

PCR Example: TV Shows



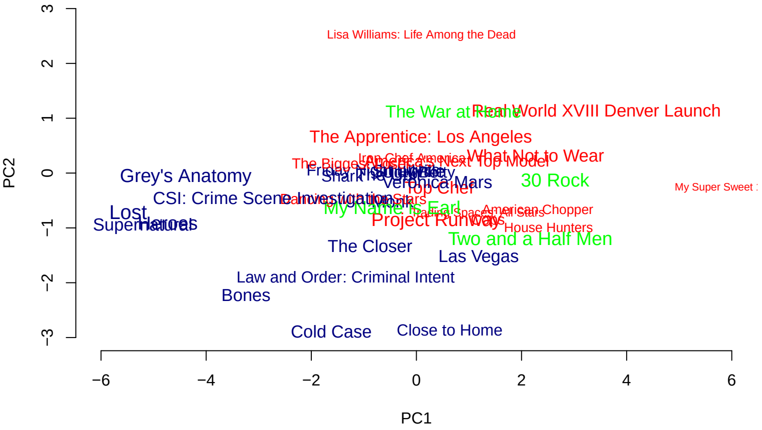
PCR: Example

```
round(PCApilot$rotation[,1:3],1)
```

```
##                   PC1  PC2  PC3
## Q1_Attentive     -0.3  0.0  0.0
## Q1_Excited       -0.3  0.1 -0.1
## Q1_Happy         -0.1  0.2 -0.5
## Q1_Engaged       -0.3  0.0  0.0
## Q1_Curious       -0.3  0.0  0.1
## Q1_Motivated     -0.2  0.3  0.0
## Q1_Comforted     -0.1  0.4 -0.1
## Q1_Annoyed        0.2  0.3  0.1
## Q1_Indifferent    0.2  0.4  0.1
## Q2_Relatable     -0.1  0.3 -0.1
## Q2_Funny          0.1  0.2 -0.5
## Q2_Confusing     -0.1  0.3  0.2
## Q2_Predictable    0.2  0.3  0.0
## Q2_Entertaining -0.3 -0.1 -0.3
## Q2_Fantasy       -0.1  0.2  0.1
## Q2_Original      -0.3  0.1 -0.2
## Q2_Believable    -0.1  0.1  0.1
## Q2_Boring         0.2  0.4  0.1
## Q2_Dramatic      -0.2  0.0  0.4
## Q2_Suspenseful   -0.3  0.0  0.3
```

PCR: Example

```r
zpilot <- predict(PCApilot)
```



PCR: Example

```
library(gamlr)
PE <- shows$PE
zdf <- as.data.frame(zpilot)
summary(PEglm <- glm(PE ~ ., data=zdf[,1:2]))
```

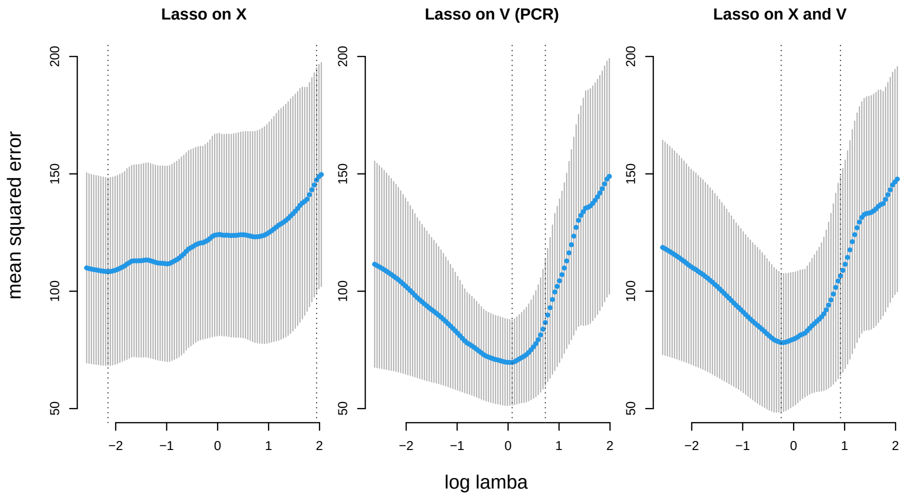
```
##
## Call:
## glm(formula = PE ~ ., data = zdf[, 1:2])
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -17.7970   -6.6583   -0.7242    6.7524   17.9895
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  72.6831     1.4370  50.580  < 2e-16 ***
## PC1          -2.6401     0.5202  -5.075 1.12e-05 ***
## PC2          -1.5029     0.6349  -2.367   0.0233 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 82.59648)
##
##     Null deviance: 5646.5  on 39  degrees of freedom
## Residual deviance: 3056.1  on 37  degrees of freedom
## AIC: 294.96
##
## Number of Fisher Scoring iterations: 2
```

PCR: Example

```
cvlassoPCR <- cv.gamlr(x=zpilot, y=PE, nfold=20) # nfold=20 for leave-two-out CV...
coef(cvlassoPCR)
```

```
## 21 x 1 sparse Matrix of class "dgCMatrix"
##                 seg28
## intercept 72.6830750
## PC1       -1.8881753
## PC2       -0.5852612
## PC3       -0.7950954
## PC4        .
## PC5        .
## PC6        .
## PC7       -3.1873761
## PC8        .
## PC9        .
## PC10       .
## PC11       .
## PC12       .
## PC13       .
## PC14       .
## PC15       .
## PC16      10.0701768
## PC17       .
## PC18       .
## PC19       .
## PC20       .
```

PCR: Example



Lasso on X
Lasso on V (PCR)
Lasso on X and V
mean squared error
50
100
150
200
−2
−1
0
1
2
log lamba

Partial Least Squares

- In the previous two examples, there was a clear low-dimensional factor structure in X: ideology in Congress and like-versus-dislike in the TV pilot survey.
- For the TV pilots, these factors were also directly related to the y response of interest.
- Nature will not always be this nice. It is common to encounter X data that has been generated without a clear factor structure, or through some messy mix of underlying factors and idiosyncratic shocks.
- And even when there is factor structure in X, it will often be that y is not related to the dominant sources of variation in X.
- The response is not driven by the first few PCs, and it is inefficient to try to estimate f as a middle-man between y and X.

Partial Least Squares

- PCR will work only if the dominant directions of variation in X are related to y.
- However, the idea of combining inputs into a few factors (or indices) that affect y is an appealing framework.
- Is there a way to force factors f to be relevant to both X and y?

Partial Least Squares

- Yes;

Partial Least Squares

- Yes;
- It's known as supervised supervised factor modeling, and it is a useful big data technique.
- There is a big world of supervised factor modeling, and there are several algorithms for supervised adaptations of PCA.
- We'll consider the simple but powerful method of partial least squares (PLS)
- To understand PLS, we start with the more basic algorithm of marginal regression (MR).

Marginal Regression (MR)

- Idea,
  1. Run $y \Rightarrow$ on each X
  2. Use the coefficients to map from X to a univariate factor F
- This factor aggregates the first-order effect of each input variable on y.
- It will be dominated by X_j dimensions that both
  1. have a big effect on y
  2. move consistently in the same direction with each other (since their influence on the factor is additive).
- That is, marginal regression constructs a single factor that is connected both to y and to a dominant direction of variation in X.

Marginal Regression (MR)

- MR Algorithm

  1. Calculate $\delta = (\delta_1, ..., \delta_n)$ where $\delta_j = cor(X_j, y)/sd(X_j)$ is the OLS coefficient in a simple univariate regression for y on X_j.
  2. Set $f_i = X_i'\delta = \sum_j X_{ij}\delta_j$ for each observation i.
  3. Fit the "forward" univariate linear regression $y_i = \alpha + \beta f_i + \epsilon_i$
  4. Given a new X,we can predict $\hat{y} = \alpha + \beta X'\delta$

Marginal Regression (MR)

- One big advantage of MR is computational efficiency.
- We can use MapReduce:
  - In the Map step, you produce $(x_i j, y_i)$ pairs that are indexed by the dimension key j;
  - the Reduce step then runs univariate OLS for y on x_j and returns δ_j.
- It works in high dimensions even if $p >> n$.
- MR is a strategy for supervised learning in ultra-high dimensions.

Partial Least Squares

- Partial least squares (PLS) is an extension of marginal regression.

Partial Least Squares

- Partial least squares (PLS) is an extension of marginal regression.
- Instead of stopping after running the single MR, you iterate:
  - Take the residuals from the first MR and repeat a second MR to predict these residuals.
  - You can then take the residuals from the second MR and repeat, continuing until you reach the minimum of p and n.

Partial Least Squares

- PLS Algorithm

  1. Begin by running MR algorithm as before for y on x.
  2. Store the MR factor as f_1, the 1st PLS direction, and PLS(1) forward regression fitted values as $\hat{y}_1 = \alpha + \beta_1 f_1$)
  3. Then, for k = 2, ... K, calculate the following:
     - Residuals
     - Loadings
     - Fitted values
  4. This yields PLS rotations $\delta = (\delta_1, ..., \delta_k)$ and factors $F = (f_1, ..., f_k)$.

Partial Least Squares

- This yields PLS rotations $\delta = (\delta_1, ..., \delta_k)$ and factors $F = (f_1, ..., f_k$.
- The PLS algorithm involves a number of steps but is really very simple.
- We are just running marginal regression on the residuals after each PLS(k) fit and updating the fitted values.
- This general procedure of taking a simple algorithm and repeatedly applying it to residuals from previous fits. (This is boosting!!)

Partial Least Squares

- If $p < n$ and you do PLS with $K = p$, then the fitted $\hat{y}_i^K$ will be the same as what you would get running OLS for y on X.
- The PLS coefficients on each X_i, available as $\sum_k \beta_k \delta_{kj}$, also match the OLS coefficients.
- Thus, PLS provides a path of models between MR and OLS.
- Remember whenever you are boosting, there is a potential for overfit.

Text Regression: Example (Gentzkow and Shapiro)

```r
#load packages
library(textir)
#load data
data(congress109)
congress109Counts[c("Barack Obama","John Boehner"),995:998]
```

```
## 2 x 4 sparse Matrix of class "dgCMatrix"
##              stem.cel natural.ga hurricane.katrina trade.agreement
## Barack Obama        .          1                20               7
## John Boehner        .          .                14               .
```

```r
congress109Ideology[1:4,1:5]
```

```
##                            name party state chamber  repshare
## Chris Cannon       Chris Cannon     R    UT       H 0.7900621
## Michael Conaway Michael Conaway     R    TX       H 0.7836028
## Spencer Bachus   Spencer Bachus     R    AL       H 0.7812933
## Mac Thornberry   Mac Thornberry     R    TX       H 0.7776520
```

Text Regression: Example (Gentzkow and Shapiro)

- We used LASSO and got

```r
head(sort(round(B[B!=0],4)),10)
```

```
##    congressional.black.caucu                       family.value
##                      -0.0839                            -0.0443
##        issue.facing.american                voter.registration
##                      -0.0324                            -0.0298
##      minority.owned.business                 strong.opposition
##                      -0.0284                            -0.0264
##                  civil.right             universal.health.care
##                      -0.0259                            -0.0254
## congressional.hispanic.caucu               ohio.electoral.vote
##                      -0.0187                            -0.0183
```

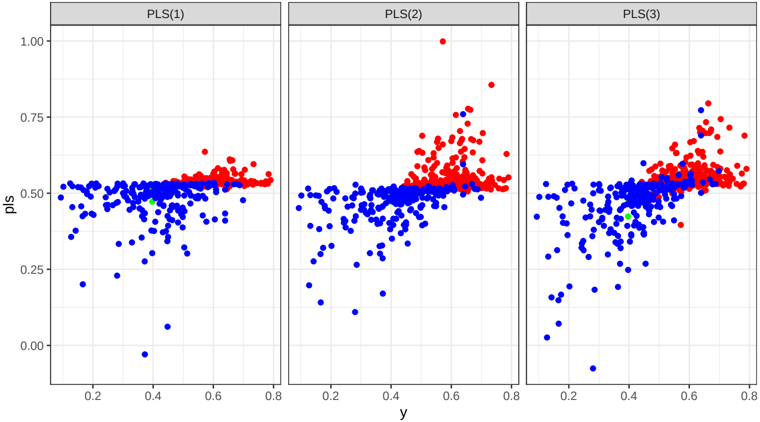
```r
tail(sort(round(B[B!=0],4)),10)
```

```
##         illegal.alien        percent.growth   illegal.immigration
##                0.0079                0.0083                0.0087
##            global.war          look.forward             war.terror
##                0.0098                0.0099                0.0114
##      private.property        action.lawsuit          human.embryo
##                0.0133                0.0142                0.0226
## million.illegal.alien
##                0.0328
```

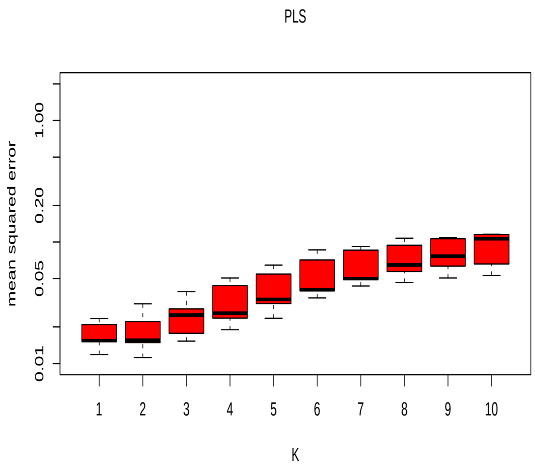
Text Regression: Example (Gentzkow and Shapiro)

```
slant <- pls(f, y, K=3)
```

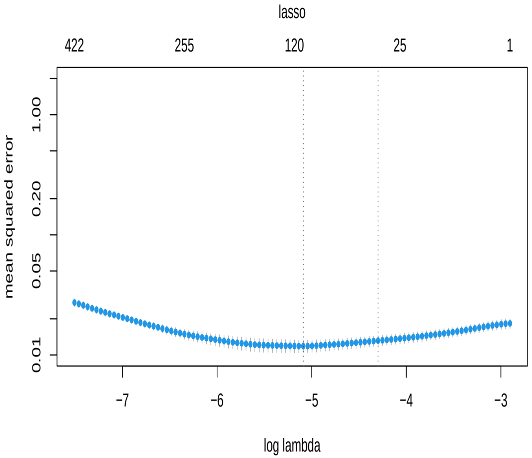
```
## Directions 1, 2, 3, done.
```



Text Regression: Example (Gentzkow and Shapiro)



(a) MSE PLS



(b) MSE Lasso

Review & Next Steps

- Factor Models: Example PCA
- Next class:
  - Presentation Rafael
  - Word Embedings
- Questions? Questions about software?

Further Readings

- Gentzkow, M., & Shapiro, J. M. (2010). What drives media slant? Evidence from US daily newspapers. Econometrica, 78(1), 35-71.
- Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning (Vol. 1, No. 10). New York: Springer series in statistics.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: springer.
- Taddy, M. (2019). Business data science: Combining machine learning and economics to optimize, automate, and accelerate business decisions. McGraw Hill Professional.