ECON2300 - Introductory Econometrics

Tutorial 12: Prediction with Many Regressors and Big Data

Tutor: Francisco Tavares Garcia

Report 2 is available!

ECON 2300: INTRODUCTORY ECONOMETRICS
Coordinator: Professor Rodney Strachan
Research Project 2
Due: 4 pm, 6 November

Submission of your report

Your report must be single-spaced and in 12 Font size. You should give your answer to each of the following questions following a similar format of the solutions to the tutorial problem sets. When you are required to use R, you must show your R command and R `outputs` (screenshots or figures generated from R). You will lose **2 points** whenever you fail to provide R commands and `outputs`. For each question, when you are asked to discuss or interpret, your answer has to be brief and compact. You will lose **2 points** if your answer is needlessly wordy. You must upload your assignment on the course webpage (Blackboard) in PDF format. (Do not submit a hard copy.)

This project has two research questions. You are required to investigate both of them.

Problem 1: money, Growth, and Inflation (30 marks)

Background

To examine the quantity theory of `money`, Brumm (2005) ["Money Growth, Output Growth, and Inflation: A Reexamination of the Modern Quantity Theory's Linchpin Prediction," *Southern Economic Journal*, 71(3), 661–667] specifies the inflation equation

$$\texttt{inflat} = \beta_1 + \beta_2\texttt{money} + \beta_3\texttt{output} + u$$

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Report 2
is available!

Problem 1: money, Growth, and Inflation (30 marks)

Background

To examine the quantity theory of `money`, Brumm (2005) ["Money Growth, Output Growth, and Inflation: A Reexamination of the Modern Quantity Theory's Linchpin Prediction," *Southern Economic Journal*, 71(3), 661–667] specifies the inflation equation

$$\texttt{inflat} = \beta_1 + \beta_2\texttt{money} + \beta_3\texttt{output} + u$$

where `inflat` is the growth rate of the general price level, `money` is the growth rate of the money supply, and `output` is the growth rate of national output. Economic theory suggests that $\beta_2 = 1$ and $\beta_3 = -1$. The dataset `brumm.dta` consists of 1995 data on 76 countries.

Research tasks

1. It is argued that `output` may be endogenous. Four instrumental variables are proposed, `initial` = initial level of real GDP, `school` = a measure of the population's educational attainment, `inv` = average investment share of GDP, and `poprate` = average population growth rate.
   (a) Give an intuitive explanation as to why `output` can be endogenous (3 marks)
   (b) Explain why the proposed IVs can be valid. (6 marks, i.e., 2 marks for understanding of valid IVs and 1 mark for convincing story for each IV)
2. Using the four IVs, obtain TSLS estimates of the inflation equation (4 marks), and test the economic theory using the IV estimates (4 marks).
3. Determine whether the IVs are strong or not (3 marks) and test if they are exogeneous (3 marks).
4. Present a short research note (less than a half page) of your findings. You are allowed to use your previous findings here again or to estimate again the model using a different set of IVs (7 marks).

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Problem 2: Demand for Democracy (70 marks)

Background:

Do citizens demand more democracy and political freedom as their incomes grow? That is, is democracy a normal good? To investigate this issue, you will explore the dataset `Income_Democracy.dta` which contains a panel data set from 195 countries for the years 1960, 1965, ... , 2000. A detailed description is given in `Income_Democracy_Description.pdf`.[1] The dataset contains an index of political freedom/democracy for each country in each year, together with data on the country's income and various demographic controls. (The income and demographic controls are lagged five years relative to the democracy index to allow time for democracy to adjust to changes in these variables.)

Research tasks:

1. Is the data set a balanced panel? Explain. (5 marks)
2. The index of political freedom/democracy is labeled `dem_ind`.
   (a) What is the value of `dem_ind` for the United States in 2000? What is the average of `dem_ind` for the United States over all years in the data set? (4 marks) Repeat this exercise for Libya (2 marks).
   (b) List five countries with an average value of `dem_ind` greater than 0.95; less than 0.10; and between 0.3 and 0.7. (5 marks)
3. The logarithm of per capita income is labeled `log_gdppc` .
   (a) Regress `dem_ind` on `log_gdppc` using standard errors that are clustered by country (3 marks).
   (b) How large is the estimated coefficient on `log_gdppc`? Is the coefficient statistically significant? (2 marks)
   (c) If per capita income in a country increases by 20%, by approximately how much is `dem_ind` predicted to increase? What is a 95% confidence interval for the prediction? Is the predicted increase in `dem_ind` large or small? Explain what you mean by large or small. (5 marks)
   (d) Why is it important to use clustered standard errors for the regression? Do the results change if you do not use clustered standard errors? (4 marks)

   (a) Suggest a variable that varies across countries but plausibly varies little–or not at all–over time and that could cause omitted variable bias in the regression in Question 3 above. (5 marks)
   (b) Estimate the regression in Q3, allowing for country fixed effects. How do your answers to Q3(b) and Q3(c) change? (5 marks)
   (c) Exclude the data for Azerbaijan and rerun the regression. Do the results change? Why or why not? (5 marks)
   (d) Suggest a variable that varies over time but plausibly varies little–or not at all–across countries and that could cause omitted variable bias in the regression in Q3. (5 marks)
   (e) Estimate the regression in Q3, allowing for time and country fixed effects. How do your answers to Q3(b) and Q3(c) change? (5 marks)
   (f) There are addition demographic controls in the data set. Should these variables be included in the regression? If so, how do the results change when they are included? (5 marks)
5. Based on your analysis, what conclusions do you draw about the effects of income on democracy? (10 marks)

SETutor is available!!!

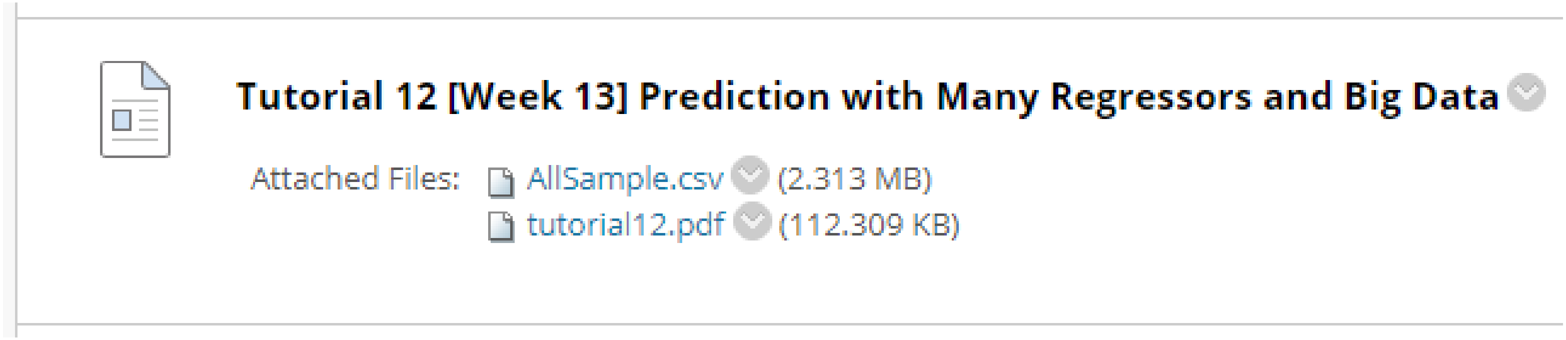
If you found these tutorials helpful, please answer the survey.

(If you didn’t, please let me know how to improve them through the survey too ☺)

This is very valuable for us tutors!

https://eval.uq.edu.au/eus.onlinesurveyportal/Home/Survey?surveyid=768118861

- Download the files for tutorial 12 from Blackboard,
- save them into a folder for this tutorial.



- Copy the code from Codeshare,
- https://codeshare.io/tut12
- Paste the code in a new script in RStudio,
- Save the script in the same folder as the data.

Based on SW E14.1

The data set `CASchools_EE14_inSample.csv` contains a subset of $n = 500$ schools from the data set used in the lecture. Included are data on test scores and 20 of the primitive predictor variables. See `CASchools_E141_Description.pdf` for a description of the variables. In this exercise, you will construct prediction models like those described in the lecture and use these models to predict test scores for 500 out-of-sample schools.

```r
rm(list = ls())
setwd("/Users/uqdkim7/Dropbox/Teaching/R tutorials/Data")
All <- read_csv("AllSample.csv") %>%
  dplyr::select(-c(x2))
attach(All)

InSample <- filter(All, InSample == 1)
OutOfSample <- filter(All, InSample == 0)

x.in <- as.matrix(InSample[,2:(ncol(All)-1)])
y.in <- as.matrix(InSample[,ncol(All)])
n.in <- nrow(x.in)
x.out <- as.matrix(OutOfSample[,2:(ncol(All)-1)])
y.out <- as.matrix(OutOfSample[,ncol(All)])
n.out <- nrow(x.out)
```

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Variables in ca_school_testcore dataset used in Chapter 14

Variable	Description
School Identifiers	
countyname	county name
districtname	district name
schoolname	school name
zipcode	zipcode of school
Test Scores	
testscore	test score (sum of math and english/language arts, 5th grade)
Predictors	
(1) str_s	student teacher(FTE) ratio (school)
(2) charter_s	charter school (0-1, school)
(3) frpm_frac_s	free or reduced price meals (fraction, school)
(4) enrollment_s	enrollment (school)
(5) ell_frac_s	english language learners (fraction, school)
(6) edi_s	ethnic diversity index (school)
(7) te_fte_s	number (fte) teachers (school)
(8) te_avgyr_s	average years teaching (school)
(9) ada_enrollment_ratio_d	avg. daily attendance divided by enrollment (district)
(10) te_salary_low_d	Teacher Salary: lowest salary offered (district)
(11) te_salary_avg_d	Teacher Salary: average (district)
(12) te_days_d	Teaching days (district)
(13) te_serdays_d	Teaching service days (district)
(14) age_frac_5_17_z	Population(1+) fraction age 5-17 years (zipcode)
(15) pop_1_older_z	Population total: 1 year and older
(16) ed_frac_hs_z	Population (25+), education: high school (zipcode)
(17) ed_frac_sc_z	Population (25+), education: some college or AA (zipcode)
(18) ed_frac_ba_z	Population (25+), education: bachelors degree (zipcode)
(19) ed_frac_grd_z	Population (25+), education: graduate or professional degree (zipcode)
(20) med_income_z	Population (15+), median income (zipcode)

(a) From the 20 primitive predictors, construct squares of all the predictors, along with all of the interactions. Collect the 20 primitive predictors, their squares, and all interactions into a set of k predictors. Verify that you have 230 predictors. Read the labels (descriptions) of the primitive predictors `charter_s`, `enrollment_s`, `str_s`, and `te_fte_s`. Drop the predictors `charter_s`^{2} and `enrollment_s` from the list of 230 predictors, leaving 228 predictors for the analysis. Why should these predictors be dropped from the original list of predictors?

Charter_s is a binary variable, so it's square has the same value as the original data. **Enrollment** is duplicate because it is equal to STR * number of teachers. As we have both of these information and the interaction term, keeping Enrollment is redundant.

(b) Compute the sample mean and standard deviation of each of the predictors, and use these to compute the standardized regressors. Compute the sample mean of `TestScore`, and subtract the sample mean from `TestScore` to compute its demeaned value.

It's already done.

	InSample	x1	x3	x4	x5	x6	x7	x8	x9	x10	x11
1	1	6.424627e-01	-0.169881080	0.23027714	-1.18838320	1.83500230	0.482724670	1.139654400	-1.7883239	-1.9634041	-1.3468236000
2	1	-3.481790e-02	-0.406277870	0.60199165	-0.58975804	0.96105295	-0.357052480	1.587800000	-0.8169367	-0.1875321	-0.2601993100
3	1	8.122372e-01	-0.855895160	1.28346820	-1.55005250	-0.20421283	-0.041741543	-1.047327300	-0.8169367	0.1676423	-0.4823016200
4	1	1.277572e+00	-0.651945050	1.65518270	-0.88907057	-0.20421283	0.527769090	0.141878600	0.6401442	0.8779911	-0.4823016200
5	1	-1.464257e-01	-0.902247430	0.41613439	-1.01378420	-1.07816220	-0.106295650	-1.494050100	0.6401442	0.5228167	1.3204324000
6	1	-3.049303e-01	0.261195240	0.97370613	1.09387530	0.08710362	2.236874800	0.299638480	0.6401442	0.8779911	-1.4399148000
7	1	9.387501e-01	1.874254800	0.35418198	0.28323701	-0.49552929	-1.124242200	0.406339790	-1.7883239	-1.9634041	-0.8485091900
8	1	1.044541e+00	0.085056439	-0.26534221	-0.09090372	-0.78684574	-0.802045400	-1.100757000	0.6401442	0.8779911	2.2908065000
9	1	1.101617e+00	0.571755470	-0.51315188	0.53266418	0.37842008	-1.697769500	-0.406961350	0.6401442	0.5228167	1.4545772000
10	1	-1.421371e+00	0.455874740	0.29222953	0.28323701	0.37842008	1.423780100	-0.264851000	-0.3312430	-0.5427065	-0.1248977800

(c) Using OLS regress the demeaned value of `TestScore` on the standardized regressors.

i Did you include an intercept in the regression? Why or why not?

ii Compute the standard error of the regression.

```
ols <- lm(y ~.-1, data = dplyr::select(InSample, -c(InSample)))
summary(ols)
```

```
Call:
lm(formula = y ~ . - 1, data = dplyr::select(InSample, -c(InSample)))

Residuals:
    Min      1Q  Median      3Q     Max 
-75.796 -16.060  -0.674  16.676  79.001 

Coefficients:
          Estimate Std. Error t value Pr(>|t|)  
x1        553.2561   581.9533   0.951 0.342607  
x3       -126.1663   556.5510  -0.227 0.820833  
x4        -79.3404   424.1324  -0.187 0.851749  
x5      -2763.2124  2239.9647  -1.234 0.218418  
x6        502.8866   357.4293   1.407 0.160583  
x7        724.8955   416.6178   1.740 0.082998 .
x8       -672.2828   464.0944  -1.449 0.148604  
x9       2142.4804  1412.0601   1.517 0.130359  
x10      -238.3989   483.6678  -0.493 0.622482  
x11      -617.1180   558.6186  -1.105 0.270257  
x12       601.8916   300.5682   2.003 0.046222 *
x13     -1003.7145   622.7196  -1.612 0.108160  
x14       248.7867   506.6023   0.491 0.623760  
x15     -2241.0936  1238.4717  -1.810 0.071467 .
```

- Because all the variables, including Y, are deviated from their means, the intercept is zero – so is omitted from (1).

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

(d) Using ridge regression with $\lambda_{Ridge} = 300$, regress the demeaned value of `TestScore` on the standardized regressors. Compare the OLS and ridge estimates of the standardized regression coefficients.

Ridge Regression

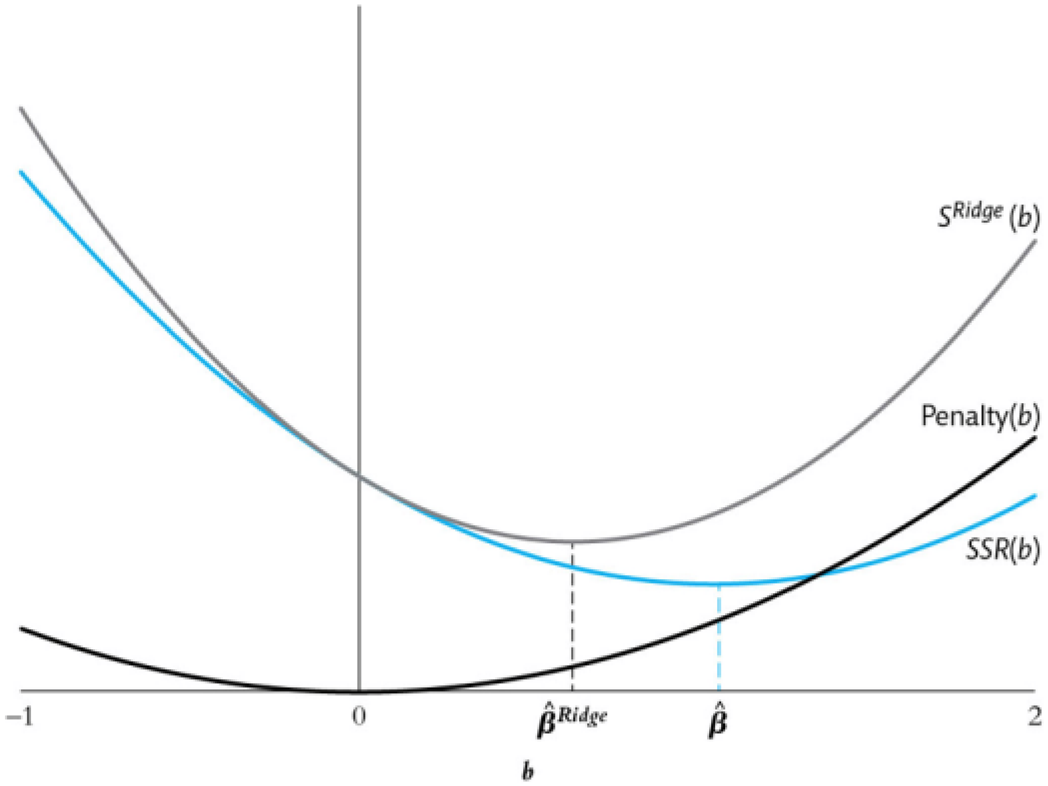
The *ridge regression* estimator minimizes the penalized sum of squares

$$S^{Ridge}(b; \lambda_{Ridge}) = \underbrace{\sum_{i=1}^{n}(Y_i - b_1 X_{1i} - \cdots - b_k X_{ki})^2}_{(1)} + \underbrace{\lambda_{Ridge} \sum_{j=1}^{k} b_j^2}_{(2)}$$

where $\lambda_{Ridge} \geq 0$ is called the *shrinkage parameter*.

- (1) is the usual sum of squared residuals.
- (2) is called a *penalty term* as it penalizes the estimator for choosing a large estimate of β.
- (1) + (2) is called the *penalized sum of squared residuals*.

Ridge Regression in a Picture



(d) Using ridge regression with $\lambda_{Ridge} = 300$, regress the demeaned value of `TestScore` on the standardized regressors. Compare the OLS and ridge estimates of the standardized regression coefficients.

```r
ridge <- glmnet(x.in, y.in, alpha = 0, lambda = 300/n.in, intercept = F, standardize = F)
coef(ridge)[1:12,]
```

```
> ridge <- glmnet(x.in, y.in, alpha = 0, lambda = 300/n.in, intercept = F, standardize = F)
> coef(ridge)[1:12,]
(Intercept)          x1          x3          x4          x5          x6          x7          x8
  0.0000000   6.4358662   3.1717732   4.5330827  -0.1651905   5.2542592   3.4278902   5.0746862
         x9         x10         x11         x12
  0.1903878  -1.4113269  -0.1231323   4.9048759
```

Notice that the ridge and Lasso objective functions used by the `glmnet` command are

$$\sum_{i=1}^{n}(Y_i - b_1X_{1i} - \cdots - b_kX_{ki})^2 + n\lambda_{Ridge}\sum_{j=1}^{k} b_j^2$$

and

$$\sum_{i=1}^{n}(Y_i - b_1X_{1i} - \cdots - b_kX_{ki})^2 + 2n\lambda_{Lasso}\sum_{j=1}^{k} |b_j|$$

, respepectively. (See https://web.stanford.edu/~hastie/glmnet/glmnet_alpha.html#intro).

(e) Using Lasso with $\lambda_{Lasso} = 100$, regress the demeaned value of `TestScore` on the standardized regressors. How many of the estimated Lasso coefficients are different from 0? Which predictors have a nonzero coefficient.

The Lasso

The Lasso estimator shrinks the estimate towards zero by penalizing large absolute values of the coefficients.

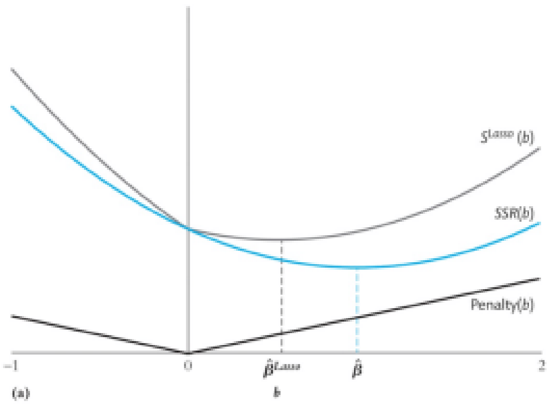
The Lasso estimator minimizes a penalized sum of squares, where the penalty term is the sum of the absolute values of the coefficients:

$$S^{Lasso}(b; \lambda_{Lasso}) = \sum_{i=1}^{n}(Y_i - b_1 X_{1i} - \cdots - b_k X_{ki})^2 + \lambda_{Lasso} \sum_{j=1}^{k} |b_j|$$

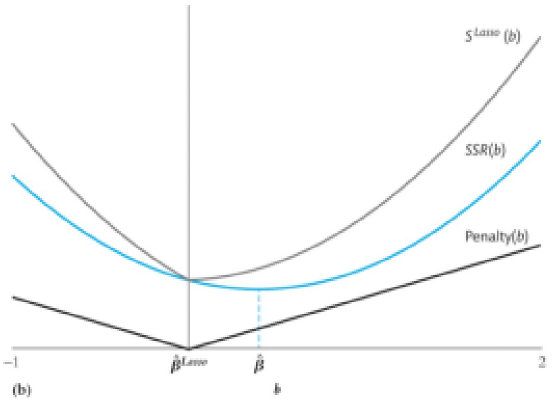
where $\lambda_{Lasso} \geq 0$ is called the *Lasso shrinkage parameter*.

This looks a lot like ridge estimation but it turns out to have very different properties...

Lasso in Pictures



Lasso shrinks large β less than ridge



Lasso shrinks small β all the way to 0

Thus, the Lasso estimator sets some many of the βs exactly to 0.

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

(e) Using Lasso with $\lambda_{Lasso} = 100$, regress the demeaned value of `TestScore` on the standardized regressors. How many of the estimated Lasso coefficients are different from 0? Which predictors have a nonzero coefficient.

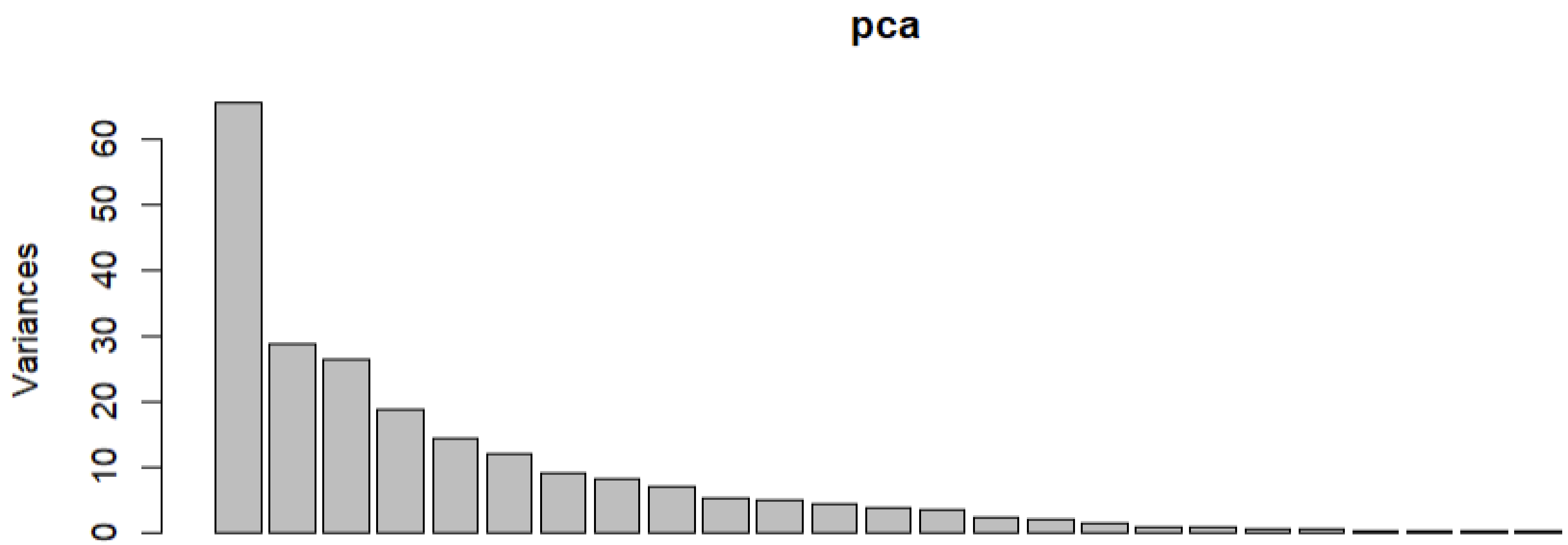
```r
lasso <- glmnet(x.in, y.in, alpha = 1, lambda = 100/(2*n.in),
                intercept = F, standardize = F)
lasso.coef <- predict(lasso, type = "coefficients")
lasso.coef
```

```r
lasso.coef[lasso.coef != 0]
```

```
> lasso.coef[lasso.coef != 0]
<sparse>[ <logic> ]: .M.sub.i.logical() maybe inefficient
 [1]  1.164132e+01 -7.128974e-01 -9.541942e-01  2.942584e+01  5.224814e+00  2.138165e+00
 [7] -2.212556e+01 -2.958260e+00 -4.754990e-01  9.123657e+00 -8.112619e-01  1.525031e+01
[13] -3.341380e+00 -1.010134e+01 -2.084372e+01  6.669142e+00  1.041021e+01 -1.252751e+01
[19] -4.119177e+00 -1.589255e+01  1.100392e+01  1.259295e+01  2.215345e-01  4.384704e+00
[25]  1.023928e+01 -4.145174e+00  5.923934e+00 -5.396947e+00  1.053925e-01 -2.268925e+01
[31] -2.129858e+00 -1.273743e+00 -4.529885e+00 -2.823698e+01  1.821586e+01  1.480679e-01
[37]  1.866649e+01 -6.768120e+00 -4.501677e+00  2.098599e+00 -1.507021e+01  3.828918e+00
[43]  8.133309e+00 -2.833963e+01 -1.615028e+01 -3.948481e+00  7.342094e+00  3.726820e+00
[49] -1.409893e+01 -5.284265e+00 -7.509761e+00  4.880184e+00 -1.624198e+01 -1.017998e+01
[55]  5.312105e+00  1.414978e+01  6.600040e+00  1.109643e+01  9.159329e+00 -8.920189e+00
[61]  4.541999e+00  1.689952e+01  6.143040e-01  1.025791e+01 -8.879919e+00 -1.136173e+01
[67]  6.615538e+00  2.927261e-01 -5.975462e-04  6.689085e+00 -9.170798e+00  5.048584e+00
[73] -1.801930e+00  4.099663e-01  1.369185e+00  3.153409e+00 -8.729957e+00  2.035807e+01
[79] -1.892360e+01 -7.772990e+00 -1.497437e+01  3.508745e+00  2.167623e+01 -7.102406e+00
[85]  4.291500e+00  2.430954e+01
> # 86 regressors with coefficient != 0
```

(f) Compute the scree plot for the 228 predictors. How much of the variance in the standardized regressors is captured by the first principal component? By the first two principal components? By the first 15 principal components?

```
pca <- prcomp(x.in, scale. = F)
screeplot(pca, type = c("barplot", "lines"), npcs = 25)
```



(g) Compute 15 principal components from the 228 predictors. Regress the demeaned value of `TextScore` on the 15 principal components.

```
pca.all <- predict(pca, newdata = x.in)
pcareg <- lm(y.in ~ pca.all[, 1:15] - 1)
summary(pcareg)
```

```
Residuals:
     Min       1Q   Median       3Q      Max 
-120.441  -26.154    0.316   25.397  158.020 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
pca.all[, 1:15]PC1    4.9200     0.2248  21.887  < 2e-16 ***
pca.all[, 1:15]PC2   -0.9670     0.3392  -2.851  0.00454 ** 
pca.all[, 1:15]PC3   -0.1709     0.3540  -0.483  0.62947    
pca.all[, 1:15]PC4    0.1111     0.4191   0.265  0.79112    
pca.all[, 1:15]PC5    0.6808     0.4796   1.420  0.15634    
pca.all[, 1:15]PC6   -0.2947     0.5229  -0.564  0.57333    
pca.all[, 1:15]PC7    3.9933     0.6004   6.651 7.88e-11 ***
pca.all[, 1:15]PC8   -4.0101     0.6268  -6.397 3.73e-10 ***
pca.all[, 1:15]PC9   -1.8237     0.6796  -2.684  0.00753 ** 
pca.all[, 1:15]PC10   0.4275     0.7947   0.538  0.59084    
pca.all[, 1:15]PC11  -0.8756     0.8040  -1.089  0.27672    
pca.all[, 1:15]PC12   1.8040     0.8663   2.082  0.03784 *  
pca.all[, 1:15]PC13  -4.5741     0.9066  -5.045 6.41e-07 ***
pca.all[, 1:15]PC14   0.4788     0.9603   0.499  0.61825    
pca.all[, 1:15]PC15  -1.6010     1.1414  -1.403  0.16135    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 40.6 on 485 degrees of freedom
Multiple R-squared:  0.5593,	Adjusted R-squared:  0.5457 
F-statistic: 41.04 on 15 and 485 DF,  p-value: < 2.2e-16
```

(h) Estimate λ_{Ridge}, λ_{Lasso}[1], and the number of principal components using 10-fold cross validation from the in-sample data set.

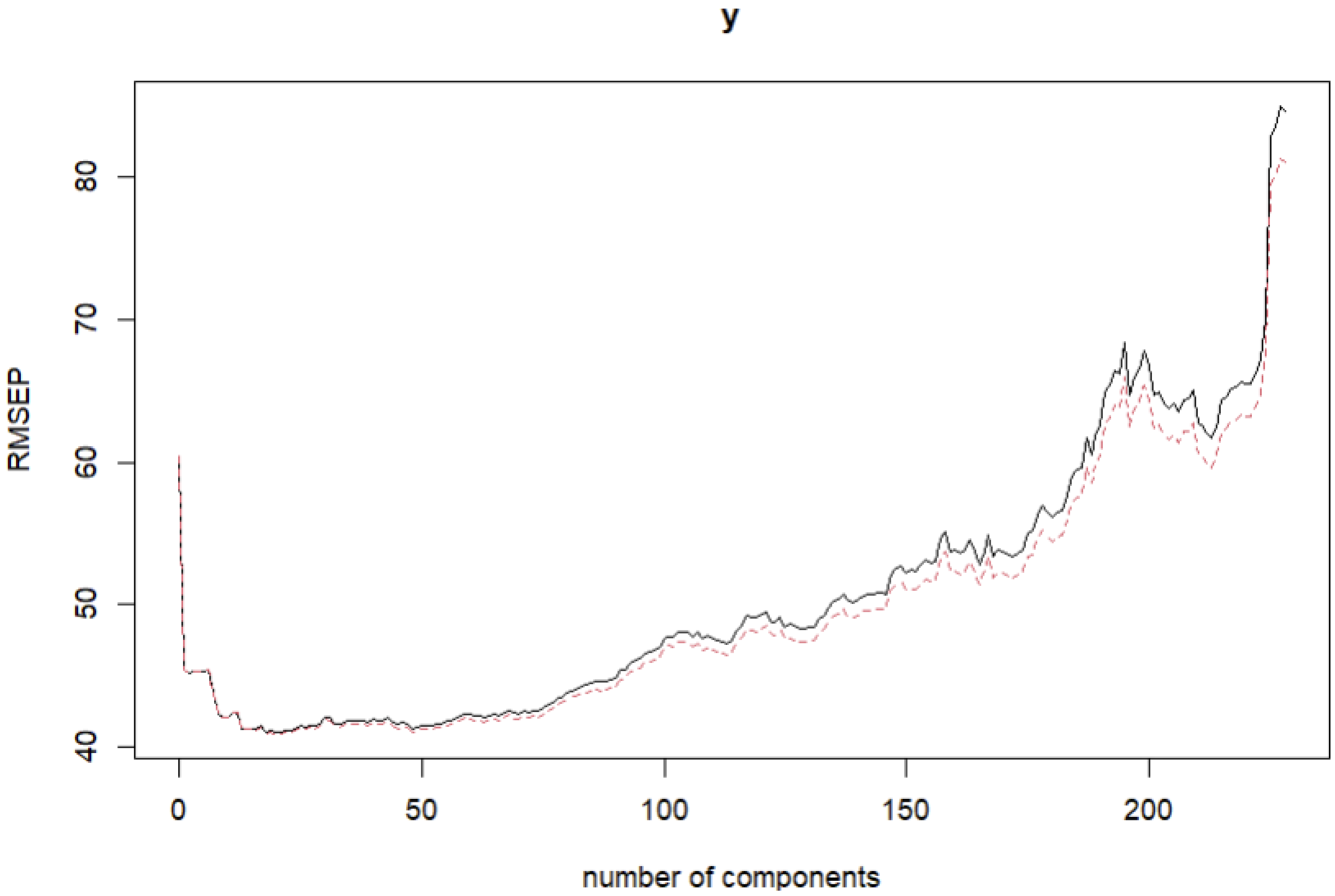
```
## (h)
set.seed(1)
cv.ridge <- cv.glmnet(x.in, y.in, alpha = 0, nfolds = 10,
                      intercept = F, standardize = F)
bestlam.ridge <- cv.ridge$lambda.min
bestlam.ridge

cv.lasso <- cv.glmnet(x.in, y.in, alpha = 1, nfolds = 10,
                      intercept = F, standardize = F)
bestlam.lasso <- cv.lasso$lambda.min
bestlam.lasso

cv.pca <- pcr(y ~.-1, data = dplyr::select(InSample, -c(InSample)),
              scale = F, center = F, validation = "CV")
summary(cv.pca)
# minimum RMSPE (Root Mean Square Error of Prediction) when p = 20
validationplot(cv.pca, val.type = "RMSEP")

bestp <- 20
```

(h) Estimate λ_{Ridge}, λ_{Lasso}[1], and the number of principal components using 10-fold cross validation from the in-sample data set.



(i) The data set `CASchools_EE14_OutOfSample.csv` contains data from another $n = 500$ schools. Use this data to conduct the following analyses.

i Predict the average test score for each of these 500 schools using the OLS, ridge, Lasso, and principal components prediction models that you estimated in (c), (d), (e), and (g). Compute the root mean square prediction error for each of these methods.

```
> ols_pred <- predict(ols, newdata = data.frame(x.out))
> sqrt(mean((ols_pred - y.out)^2))
[1] 81.12307
>
> ridge_pred <- predict(ridge, newx = x.out)
> sqrt(mean((ridge_pred - y.out)^2))
[1] 41.79715
>
> lasso_pred <- predict(lasso, newx = x.out)
> sqrt(mean((lasso_pred - y.out)^2))
[1] 41.88049
>
> pca.out <- prcomp(x.out, scale. = F)
> pca.all.out <- predict(pca.out, newdata = data.frame(x.out))
> pcareg.out <- lm(y.out ~ pca.all.out[, 1:15] - 1)
> pca_pred <- predict(pcareg.out, newdata = data.frame(x.out))
> sqrt(mean((pca_pred - y.out)^2))
[1] 38.34659
```

ii Construct four scatter plots like those in Figure 14.8. What do you learn from the plots?

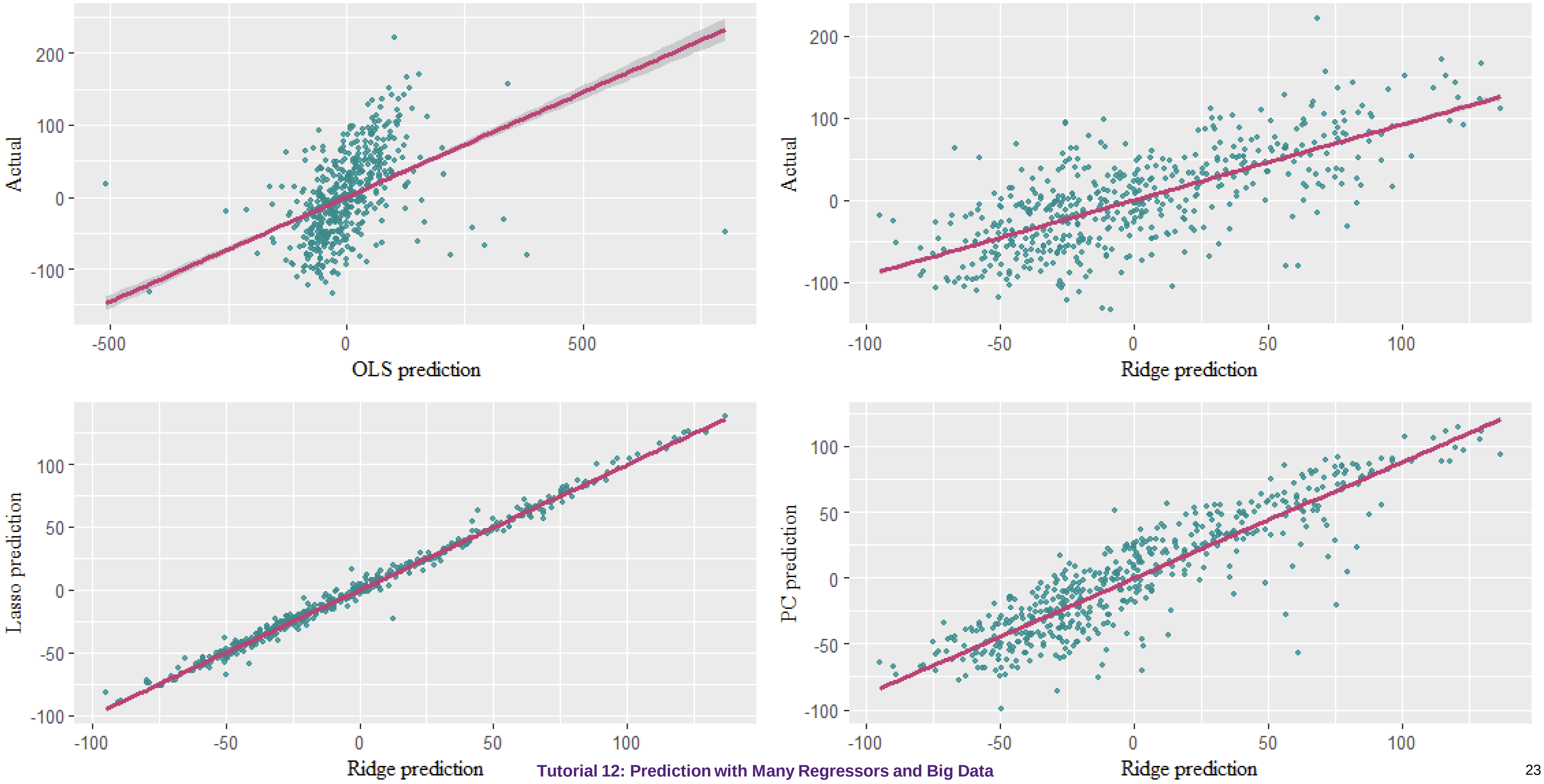
```
data_pred <- data.frame(y = y.out, ols = ols_pred, ridge = c(ridge_pred),
                        lasso = c(lasso_pred), pc = pca_pred)

scatter1 <- ggplot(data_pred, aes(x = ols, y = y)) +
  geom_point(alpha = .75, size = 1, color = "cyan4") +
  labs(x = "OLS prediction", y = "Actual") +
  geom_smooth(method = "lm", level = 0.5, color = "violetred3") +
  theme(axis.title = element_text(family = "serif"),
        plot.title = element_text(hjust = 0.5, family = "serif", face = "bold"))
scatter2 <- ggplot(data_pred, aes(x = ridge, y = y)) +
  geom_point(alpha = .75, size = 1, color = "cyan4") +
  labs(x = "Ridge prediction", y = "Actual") +
  geom_smooth(method = "lm", level = 0.5, color = "violetred3") +
  theme(axis.title = element_text(family = "serif"),
        plot.title = element_text(hjust = 0.5, family = "serif", face = "bold"))
scatter3 <- ggplot(data_pred, aes(ridge, lasso)) +
  geom_point(alpha = .75, size = 1, color = "cyan4") +
  labs(x = "Ridge prediction", y = "Lasso prediction") +
  geom_smooth(method = "lm", level = 0.5, color = "violetred3") +
  theme(axis.title = element_text(family = "serif"),
        plot.title = element_text(hjust = 0.5, family = "serif", face = "bold"))
scatter4 <- ggplot(data_pred, aes(ridge, pc)) +
  geom_point(alpha = .75, size = 1, color = "cyan4") +
  labs(x = "Ridge prediction", y = "PC prediction") +
  geom_smooth(method = "lm", level = 0.5, color = "violetred3") +
  theme(axis.title = element_text(family = "serif"),
        plot.title = element_text(hjust = 0.5, family = "serif", face = "bold"))

figMatrix <- ggarrange(scatter1, scatter2, scatter3, scatter4,
                       ncol = 2, nrow = 2)
annotate_figure(figMatrix, top = text_grob("Scatterplots for OOS Predictions",
                                           face = "bold", size = 12, family = "serif"))
```

ii Construct four scatter plots like those in Figure 14.8. What do you learn from the plots?

Scatterplots for OOS Predictions



(j) Use the estimated values of λ_{Ridge}, λ_{Lasso}, and the number of principal components from (h) to construct predictions of test scores for the out-of-sample schools. Are these predictions more accurate than the predictions you computed in (i)? Is the difference in line with what you expected from the cross-validation calculations in (h)?

```
best.ridge <- glmnet(x.out, y.out, alpha = 0, lambda = bestlam.ridge,
                     intercept = F, standardize = F)
best.lasso <- glmnet(x.out, y.out, alpha = 1, lambda = bestlam.lasso,
                     intercept = F, standardize = F)
```

```
best.ridge_pred <- predict(best.ridge, newx = x.out)
sqrt(mean((best.ridge_pred - y.out)^2))
```

```
## [1] 37.27344
```

```
best.lasso_pred <- predict(best.lasso, newx = x.out)
sqrt(mean((best.lasso_pred - y.out)^2))
```

```
## [1] 36.85517
```

```
pca <- prcomp(x.out, scale. = F)
pca.all <- predict(pca, newdata = x.out)
best.pca_pred <- lm(y.out ~ pca.all[, 1:bestp] - 1)
summary(best.pca_pred)
```

(j) Use the estimated values of λ_{Ridge}, λ_{Lasso}, and the number of principal components from (h) to construct predictions of test scores for the out-of-sample schools. Are these predictions more accurate than the predictions you computed in (i)? Is the difference in line with what you expected from the cross-validation calculations in (h)?

```
Call:
lm(formula = y.out ~ pca.all[, 1:bestp] - 1)

Residuals:
    Min      1Q  Median      3Q     Max 
-149.05  -23.82   -0.05   23.88  160.89 

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)    
pca.all[, 1:bestp]PC1   -4.4529     0.2164 -20.573  < 2e-16 ***
pca.all[, 1:bestp]PC2   -1.0796     0.3112  -3.469 0.000569 ***
pca.all[, 1:bestp]PC3   -2.4985     0.3328  -7.506 2.98e-13 ***
pca.all[, 1:bestp]PC4   -0.5034     0.4085  -1.232 0.218440    
pca.all[, 1:bestp]PC5    1.0069     0.4791   2.102 0.036113 *  
pca.all[, 1:bestp]PC6    1.1019     0.5275   2.089 0.037253 *  
pca.all[, 1:bestp]PC7   -4.3935     0.5546  -7.922 1.64e-14 ***
pca.all[, 1:bestp]PC8    1.0469     0.6386   1.639 0.101795    
pca.all[, 1:bestp]PC9   -5.9887     0.6694  -8.946  < 2e-16 ***
pca.all[, 1:bestp]PC10   0.1265     0.7187   0.176 0.860310    
pca.all[, 1:bestp]PC11  -1.9308     0.7766  -2.486 0.013254 *  
pca.all[, 1:bestp]PC12   1.9183     0.7943   2.415 0.016101 *  
pca.all[, 1:bestp]PC13  -0.5138     0.8751  -0.587 0.557350    
pca.all[, 1:bestp]PC14   1.9095     0.8979   2.127 0.033963 *  
pca.all[, 1:bestp]PC15  -1.9931     1.0688  -1.865 0.062816 .  
pca.all[, 1:bestp]PC16  -0.2773     1.1710  -0.237 0.812913    
pca.all[, 1:bestp]PC17   1.6155     1.4017   1.153 0.249686    
pca.all[, 1:bestp]PC18  -1.4672     1.6770  -0.875 0.382066    
pca.all[, 1:bestp]PC19  -0.1939     2.0065  -0.097 0.923037    
pca.all[, 1:bestp]PC20  -1.9590     2.1123  -0.927 0.354166    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 39.01 on 480 degrees of freedom
Multiple R-squared:  0.5829,	Adjusted R-squared:  0.5655 
F-statistic: 33.54 on 20 and 480 DF,  p-value: < 2.2e-16
```

The University of Queensland
AUSTRALIA
CREATE CHANGE

Thank you

Francisco Tavares Garcia

Academic Tutor | School of Economics

tavaresgarcia.github.io

Reference

Stock, J. H., & Watson, M. W. (2019). Introduction to Econometrics, Global Edition, 4th edition. Pearson Education Limited.