Right

Wrong

Abbie Pearson

June Suh

Jessica Warren

Dec. 6, 2016

STA4203

**Final Project**

This project uses the datasets data100.csv and data1000.csv for training, and test1000.csv for testing. All three datasets contain y as the dependent variable and the variables x1-x1000 as predictors.

The following Root Mean Square Error (RMSE) will be used:

where n is the number of observations and RSS is the residual sum of squares

**1. Using the data100.csv data for training the models:**

* a) Fit an OLS regression. Report the R2 of the obtained model.

Code:

proc import out=data100

datafile="/home/aep120/data100.csv"

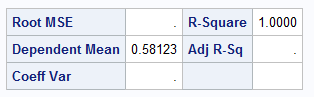
dbms = csv replace;

run;

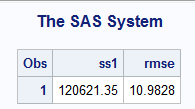
proc reg data=data100 outest = data100a;

model y=x1-x1000;

Run;



is 1.0000

* b) Compute the RMSE of the model from a) on the test set test1000.csv with four decimal places.
* Code:
* proc import out=test1000
* datafile="/home/aep120/test1000.csv"
* dbms = csv replace;
* run;
* proc score data=test1000 score=data100a residual type=parms out=parta;
* var y x1-x1000;
* run;
* proc univariate data=parta;
* var model1;
* output out=grmse uss=ss1;
* run;
* data grmse;
* set grmse;
* rmse=sqrt(ss1/1000);
* Run;
* 
* RMSE is 10.9828
* c) Perform variable selection by forward selection with significance level 0.01. Report the obtained model equation and the R2.

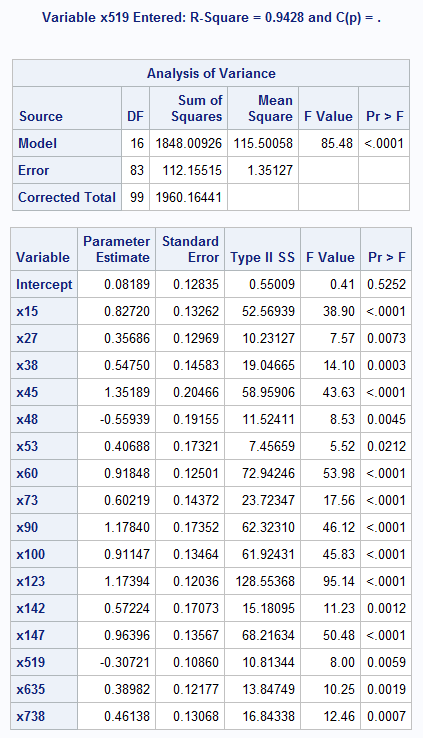
Code:

proc reg data=data100;

model y=x1-x1000/selection=forward slentry=0.01;

Run;

* Proc reg data=data100 outest=data100c;
* Model y= x15 x27 x38 x45 x48 x53 x60 x73 x90 x100 x123 x142 x147 x519 x635 x738;
* Run;
* Quit;

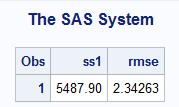


= 0.9428

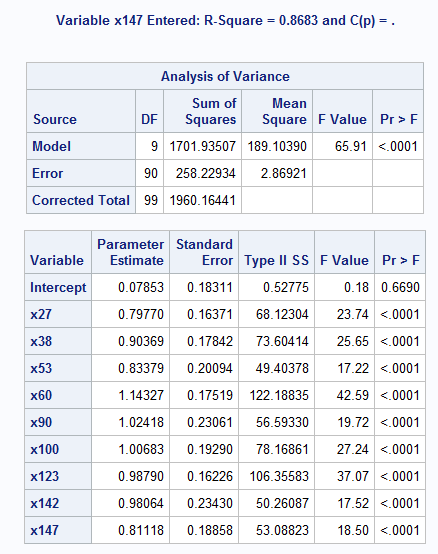
Model equation:

+

* d) Compute the RMSE of the model from c) on the test set test1000.csv with four decimal places.
* Code:
* proc score data=test1000 score=data100c residual type=parms out=partd;
* var y x15 x27 x38 x45 x48 x53 x60 x73 x90 x100 x123 x142 x147 x519 x635 x738;
* run;
* proc univariate data=partd;
* var model1;
* output out=grmse2 uss=ss1;
* run;
* data grmse2;
* set grmse2;
* rmse=sqrt(ss1/1000);
* run;
* proc print data=grmse2;



RMSE is 2.3426

* e) Perform variable selection by forward selection with significance level 0.001. Report the obtained model equation and the R2.
* Code:
* proc reg data=data100;
* model y=x1-x1000/method=forward slentry=0.001;
* Run;
* proc reg data=data100 outest=data100d;
* model y=x27 x38 x53 x60 x90 x100 x123 x142 x147;
* Run;
* 
* = 0.8683
* Model equation:

+

* f) Compute the RMSE of the model from e) on the test set test1000.csv with four decimal places.
* Code:

proc score data=test1000 score=data100d residual type=parms out=partd;

var y x27 x38 x53 x60 x90 x100 x123 x142 x147;

run;

proc univariate data=partd;

var model1;

output out=grmse2 uss=ss1;

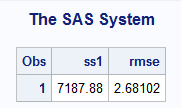
run;

data grmse2;

set grmse2;

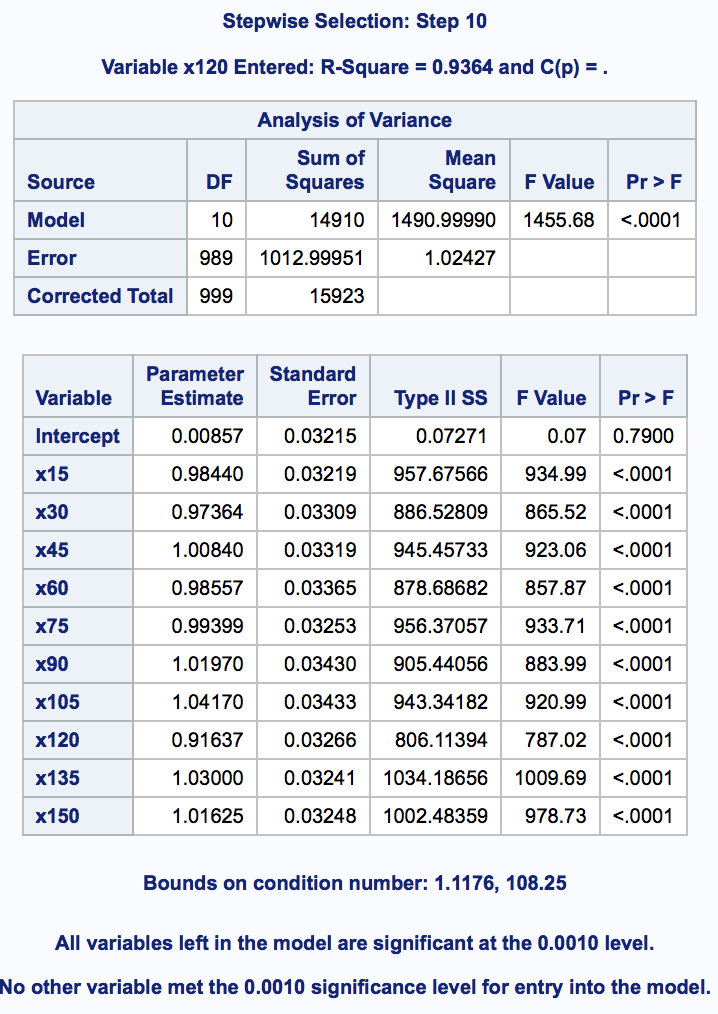
rmse=sqrt(ss1/1000);

run;



RMSE is 2.6810

**2. Using the data1000.csv data for training the models:**

* a) Perform variable selection by stepwise selection with significance level 0.001 for both entering and staying in the model. Report the obtained model equation and the R2.
* Code:
* proc import out=data1000
* datafile="/home/jes13j0/Homework/STA 4203/data1000.csv"
* dbms = csv replace;
* run;
* proc reg data=data1000;
* model y=x1-x1000 / selection=stepwise slstay=0.001 slentry=0.001;
* Run;
* proc reg data=data1000 outest=data1000a;
* model y = x15 x30 x45 x60 x75 x90 x105 x120 x135 x150;
* run;
* 
* is 0.9364.
* Model equation:
* b) Compute the RMSE of the model from a) on the test set test1000.csv with four decimal places.
* Code:

proc score data=test1000 score=data1000a residual type=parms out=scorea;

* var y x15 x30 x45 x60 x75 x90 x105 x120 x135 x150;
* run;
* proc univariate data=scorea noprint;
* var model1;
* output out=grmse uss=ss1;
* run;
* data grmse;
* set grmse;
* rmse=sqrt(ss1/1000);
* run;
* proc print data=grmse;
* Run;
* 
* RMSE is 1.0212
* c) Can one perform variable selection using the Adjusted R2 criterion? If so, report the selected variables. If it cannot be done, state the reason why.
* Result:
* No, there are so many variable it would not give viable output. R2 value decreases as variables are removed so even if the best model needs only a few of the predictors, it would show that it is significantly worse than a model with all the predictors. R 2 by itself is not a good criterion because it would always choose the largest possible model. Not to mention the fact that it simply cannot be done without massive amounts of computing time.
* d) Partial least squares with 10 factors. Report the RMSE of the obtained model on the test set test1000.csv with four decimal places.
* Code:

data train1;

set data1000 test1000 (keep=x1-x1000);

proc pls data=train1 nfac=10;

model y= x1-x1000;

output out=outpls predicted = pred;

run;

data testPred;

do obsnum=1001 to 2000;

set outpls point=obsnum;

output;

end;

stop;

run;

data test1000p;

set test1000(keep=y);

set testPred(keep=pred);

run;

data test1000p;

set test1000p;

res=y-pred;

run;

proc univariate data=test1000p noprint;

var res;

output out=teststat uss=ss1;

run;

data teststat;

set teststat;

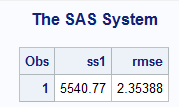
rmse=sqrt(ss1/1000);

run;

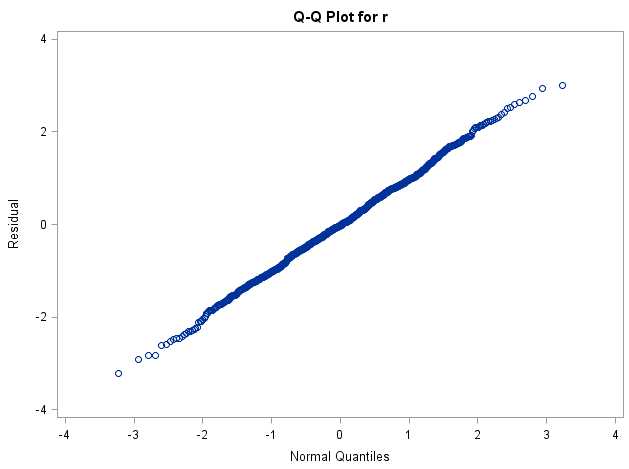
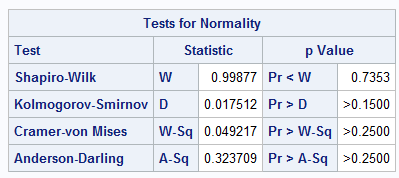
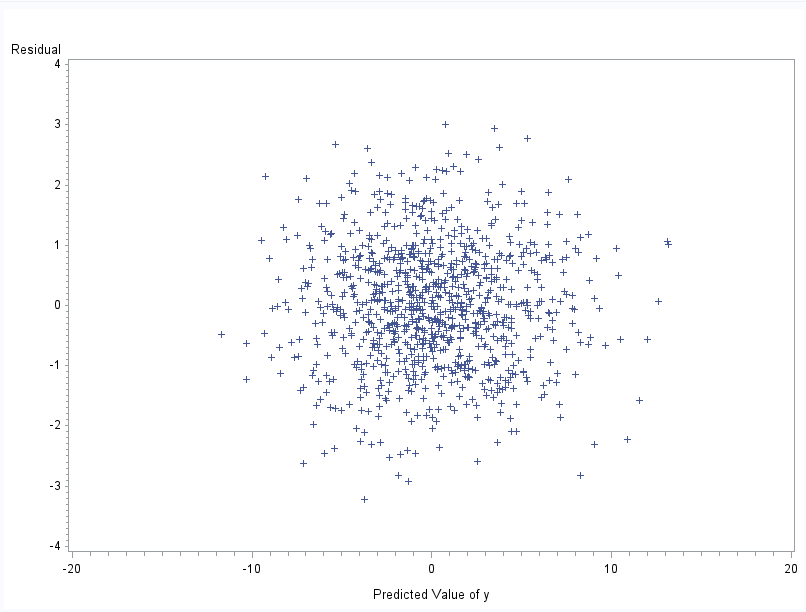
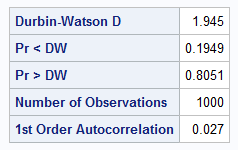
proc print data=teststat; run;



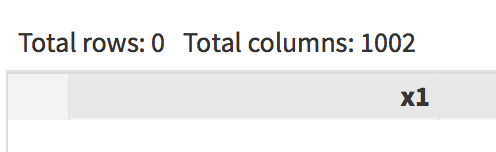
RMSE is 1.4153

* e) Ridge Regression with ridge coefficient 0.001. Report the RMSE of the obtained model on the test set test1000.csv with four decimal places.
* Code:
* proc reg data=data1000 outest=mdl ridge=0.001;
* model y =x1-x1000;
* run;
* quit;
* proc print data = mdl; run;
* data ridge;
* obsnum = 2;
* set mdl point = obsnum;
* output;
* stop;
* run;
* proc score data = test1000 score = ridge out = ridgeP residual type = ridge;
* var x1-x1000 y;
* run;
* proc univariate data=ridgep;
* var model1;
* output out=grmse3 uss=ss1;
* run;
* data grmse3;
* set grmse3;
* rmse=sqrt(ss1/1000);
* run;
* proc print data = grmse3; run;
* 
* RMSE is 2.3539

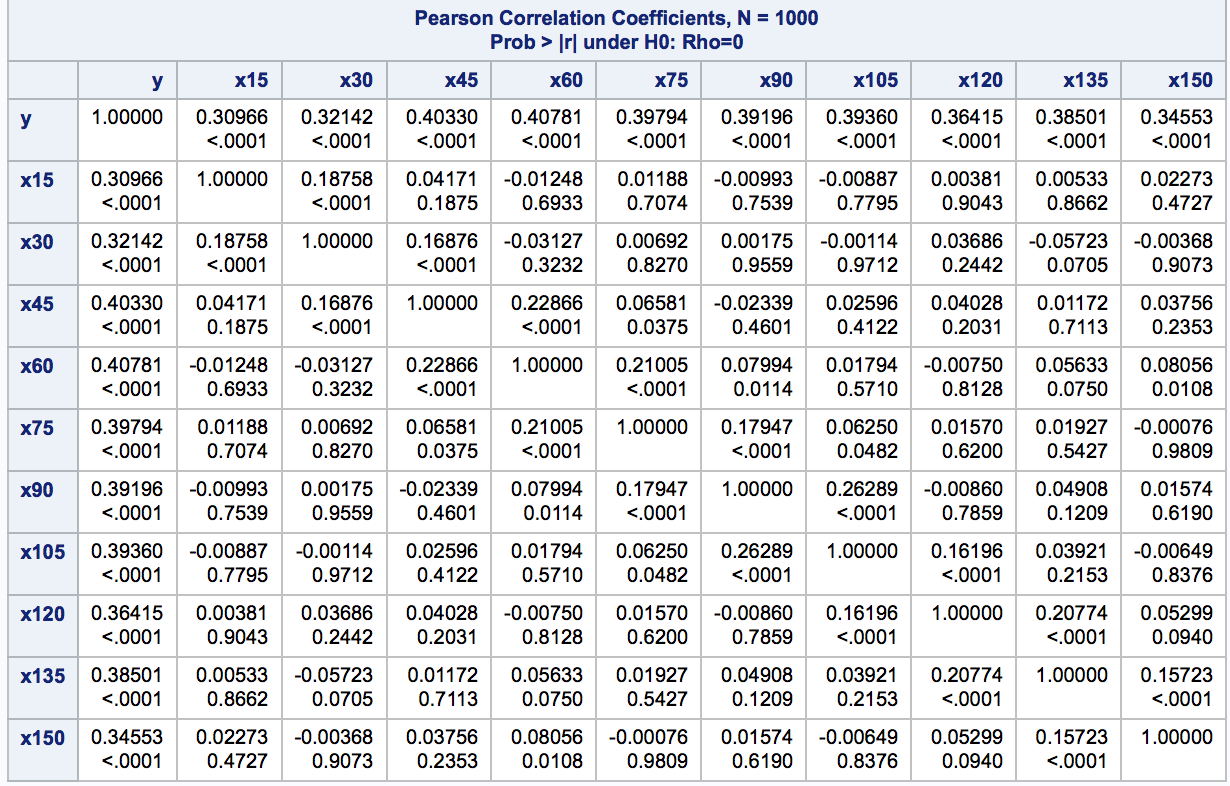
**Using the model from a):**

* e) Draw the QQ plot of the residuals and compute the tests for normality.
* Code:
* proc reg data=data1000;
* model y = x15 x30 x45 x60 x75 x90 x105 x120 x135 x150;
* output out=data1000e predicted=p residual=r;
* run;
* proc univariate data=data1000e normaltest;
* var r;
* run;
* 
* 
* f) Plot the residuals vs predicted and state whether the errors seem to have constant variance or not.
* Code:
* proc gplot data=data1000e;
* plot r\*p;
* run;
* 
* g) Test for correlated errors.
* Code:
* proc reg data=data1000;
* model y= x15 x30 x45 x60 x75 x90 x105 x120 x135 x150/dwprob;
* run;
* 
* Since Pr < DW is over 0.05 the conclusion is that there is no reason to suspect that there are correlated errors.
* h) Find any outliers and report their number.
* Code:

proc reg data=data1000;

* model y = x15 x30 x45 x60 x75 x90 x105 x120 x135 x150;
* output out=outliers rstudent=rez;
* run;
* data quantiles;
* cuttoff=abs(tinv(0.05/(2\*1000),1000-12-1));
* run;
* proc print data=quantiles; run;
* data outliers;
* do i=1 to 1000 by 1;
* set outliers point=i;
* if (abs(rez)>4.07360) then output;
* end;
* stop;
* Run;
* 
* 

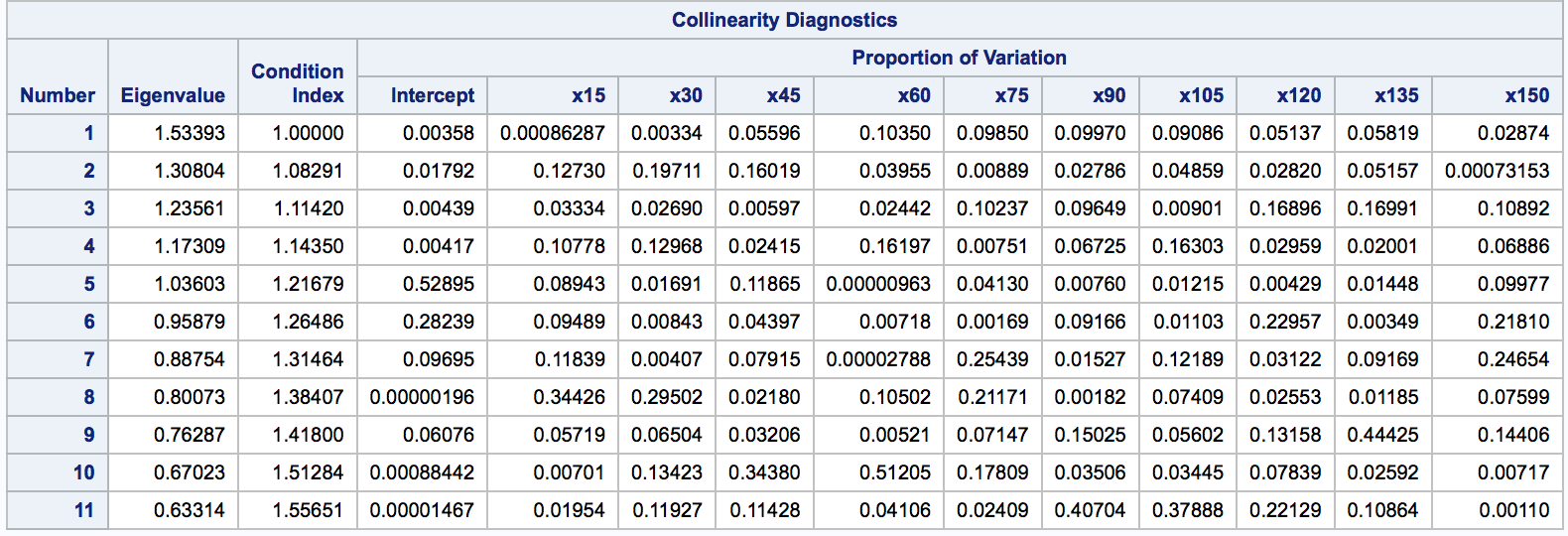
There are no outliers

* i) Compute the matrix of the correlations between the variables selected at a).
* Code:
* proc corr data=data1000;
* var y x15 x30 x45 x60 x75 x90 x105 x120 x135 x150;
* run;
* **He didn’t include the y variable**
* 
* j) Compute the condition index.
* Code:

proc reg data=data1000;

model y=x15 x30 x45 x60 x75 x90 x105 x120 x135 x150/collin;

Run;



Since condition number is 1.55651 (κ = 1.55651 < 30) it does not indicate collinearity.

* k) Can one use the Box-Cox method to find the best transformation for the response? Work only with the variables selected at a). If it cannot be done, state the reason why, otherwise report the obtained transformation.
* Result:

One cannot use the Box-Cox method, because there are invalid values. This happens because y (response) has negative values (invalid).

* l) Draw the partial regression plots.
* Code:
* proc reg data=data1000;
* model y= x15 x30 x45 x60 x75 x90 x105 x120 x135 x150/partial;
* run;

