Assignment 4

Ethan Tenison

4/13/2020

# Q1: Model Selection in the abstract (20 points)

The decennial census is a massive undertaking that involves hiring thousands of short-term employees and, locally, volunteering opportunities. The Census’ 2016 Planning Database Block Group Data contains 344 variables describing 220,357 block groups. The variable Mail\_Return\_Rate\_CEN\_2010 is the 2010 Census Mail Return Rate – suppose that when this rate is low in a block group, more people are needed to help with the count. We know the return rates, along with a lot of other characteristics of block groups, for 2010 but a lot has changed in ten years. Understanding the most important drivers of low return rates in 2010 could help with planning for 2020.

### (a) Describe how you might use forward vs. backward stepwise selection to find the best model of

###low return rates. Explain the advantages and disadvantages of one approach over the other. ##Explain how these approaches compare to best subset selection.

Using forward and backward stepwise selection will help us decide which features are most important in our model, especially when n < p. Both are computationally superior to best subset selection, where 2^p models are created. While FSS is computationally superior, there is still a chance that the selected model will not be the best one. Some casual predictors with low p-values at the beginning of the selection process might not be included . BSS does not have this problem because all predictors are added into the model first, and then substracted. The downside of BSS is that n must be greater than p.

### (b) How would you use model-level comparison metrics to determine when one model is better

### than another? Explain how a “penalty” works and how they manifest differently in different

### comparison metrics.

There are essentially two methods for comparing models, make adjustments to the training error to account for overfitting, and directly estimating test error through cross-validation.

Since R^2 and MSE increase and decrease monotonically as predictors are added, metrics such as Cp, AIC, BIC, and adjusted R^2 are used. Each of them has a penalty term which increases with the number of predictors.

**Cp** = (RSS - 2d), where d is the number of predictors, and is an estimate of the error associated with each response measurement. THis is done to adjust for the decline in RSS that occurs when more predictors are added.

**AIC** = (RSS - 2d), which uses maximum likelihood and is proportional to *Cp*

**BIC** = (RSS - log(n)d), which uses a log instead of 2d, resulting in a larger penality for a high number of predictors.

**Adjusted**  = , the result being that adding predictors that are just noise will only slightly decrease the RSS.

# Q2: Model Selection in data (40 points)

### (a) Use both forward and backward stepwise selection to find the best models containing 10 or

### fewer predictors. List the predictors in the order in which they enter the model for both

### methods.

library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

library(leaps)  
library(MASS)

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

# Fit the full model   
#full.model <- lm(Mail\_Return\_Rate\_CEN\_2010 ~., data = train)  
# Stepwise regression model  
#step.model <- stepAIC(full.model, direction = "forward", steps = 10,   
# trace = FALSE)  
#summary(step.model)  
  
#models <- regsubsets(Mail\_Return\_Rate\_CEN\_2010~., data = train, nvmax = 10, method = "forward")  
#summary(models)  
  
library(olsrr)

##   
## Attaching package: 'olsrr'

## The following object is masked from 'package:MASS':  
##   
## cement

## The following object is masked from 'package:datasets':  
##   
## rivers

model<-lm(Mail\_Return\_Rate\_CEN\_2010~., data = train)  
  
FWDfit.p<-ols\_step\_forward\_p(model,penter=.0002)  
  
#This gives you the short summary of the models at each step  
print("Forward")

## [1] "Forward"

FWDfit.p

##   
## Selection Summary   
## ---------------------------------------------------------------------------------------------------  
## Variable Adj.   
## Step Entered R-Square R-Square C(p) AIC RMSE   
## ---------------------------------------------------------------------------------------------------  
## 1 Med\_HHD\_Inc\_BG\_ACS\_10\_14 0.3780 0.3767 3373.8125 3341.7398 7.2426   
## 2 Renter\_Occp\_HU\_CEN\_2010 0.5419 0.5400 2358.3523 3193.5459 6.2218   
## 3 Sngl\_Prns\_HHD\_CEN\_2010 0.6700 0.6680 1564.8862 3034.4116 5.2857   
## 4 Med\_House\_Value\_TR\_ACSMOE\_10\_14 0.6899 0.6873 1226.0878 2878.5027 4.8781   
## 5 Med\_House\_Value\_TR\_ACS\_10\_14 0.7024 0.6992 1160.0969 2860.9229 4.7843   
## 6 RPLCMNT\_FRMS\_CEN\_2010 0.7149 0.7112 1093.8870 2842.4396 4.6878   
## 7 Hispanic\_CEN\_2010 0.7237 0.7196 1047.5430 2829.3714 4.6194   
## 8 FRST\_FRMS\_CEN\_2010 0.7400 0.7356 960.6080 2802.3463 4.4860   
## 9 Males\_CEN\_2010 0.7815 0.7773 735.7990 2721.2009 4.1168   
## 10 One\_Health\_Ins\_ACS\_10\_14 0.7911 0.7866 685.4222 2701.7551 4.0297   
## ---------------------------------------------------------------------------------------------------

BWDfit.p <-ols\_step\_backward\_p(model, prem = .000001)  
  
#This gives you the short summary of the models at each step  
print("Backward")

## [1] "Backward"

print(BWDfit.p)

##   
##   
## Elimination Summary   
## ---------------------------------------------------------------------------------------------------  
## Variable Adj.   
## Step Removed R-Square R-Square C(p) AIC RMSE   
## ---------------------------------------------------------------------------------------------------  
## 1 MLT\_U2\_9\_STRC\_ACSMOE\_10\_14 0.9269 0.8786 158.0002 1998.5244 2.5722   
## 2 Pop\_5\_17\_CEN\_2010 0.9269 0.8791 156.0006 1996.5251 2.5668   
## 3 avg\_Agg\_HH\_INC\_ACSMOE\_10\_14 0.9269 0.8796 154.0011 1994.5260 2.5615   
## 4 Pov\_Univ\_ACSMOE\_10\_14 0.9269 0.8801 152.0018 1992.5271 2.5562   
## 5 Prs\_Blw\_Pov\_Lev\_ACSMOE\_10\_14 0.9269 0.8806 150.0031 1990.5293 2.5509   
## 6 Aggregate\_HH\_INC\_ACSMOE\_10\_14 0.9269 0.8811 148.0064 1988.5348 2.5457   
## 7 ENG\_VW\_OTHER\_ACS\_10\_14 0.9269 0.8816 146.0099 1986.5407 2.5405   
## 8 Rel\_Child\_Under\_6\_ACSMOE\_10\_14 0.9269 0.882 144.0147 1984.5487 2.5354   
## 9 Not\_HS\_Grad\_ACS\_10\_14 0.9269 0.8825 142.0193 1982.5563 2.5303   
## 10 Pop\_18\_24\_ACS\_10\_14 0.9269 0.883 140.0243 1980.5647 2.5252   
## 11 NH\_Asian\_alone\_ACS\_10\_14 0.9269 0.8834 138.0363 1978.5848 2.5202   
## 12 Sngl\_Prns\_HHD\_ACS\_10\_14 0.9269 0.8839 136.0484 1976.6050 2.5152   
## 13 Rel\_Family\_HHD\_ACS\_10\_14 0.9269 0.8844 134.0673 1974.6366 2.5103   
## 14 Othr\_Lang\_ACSMOE\_10\_14 0.9269 0.8848 132.0893 1972.6734 2.5054   
## 15 Diff\_HU\_1yr\_Ago\_ACS\_10\_14 0.9269 0.8853 130.1139 1970.7146 2.5006   
## 16 Females\_ACSMOE\_10\_14 0.9269 0.8857 128.1388 1968.7561 2.4958   
## 17 Pop\_25\_44\_CEN\_2010 0.9269 0.8861 126.1742 1966.8154 2.4911   
## 18 Two\_Plus\_Health\_Ins\_ACS\_10\_14 0.9268 0.8866 124.2071 1964.8703 2.4864   
## 19 One\_Health\_Ins\_ACSMOE\_10\_14 0.9268 0.887 122.2439 1962.9318 2.4817   
## 20 Med\_House\_Value\_TR\_ACS\_10\_14 0.9268 0.8874 120.2782 1960.9890 2.4771   
## 21 Pop\_25\_44\_ACSMOE\_10\_14 0.9268 0.8878 118.3197 1959.0584 2.4725   
## 22 Mobile\_Homes\_ACSMOE\_10\_14 0.9268 0.8882 116.3621 1957.1292 2.4680   
## 23 Med\_House\_Value\_BG\_ACS\_10\_14 0.9268 0.8886 114.4205 1955.2267 2.4636   
## 24 Sngl\_Prns\_HHD\_ACSMOE\_10\_14 0.9268 0.889 112.4748 1953.3172 2.4591   
## 25 NH\_White\_alone\_ACSMOE\_10\_14 0.9267 0.8894 110.5451 1951.4347 2.4548   
## 26 Single\_Unit\_ACSMOE\_10\_14 0.9267 0.8898 108.6174 1949.5552 2.4505   
## 27 HHD\_Moved\_in\_ACS\_10\_14 0.9267 0.8902 106.6863 1947.6701 2.4463   
## 28 ENG\_VW\_API\_ACS\_10\_14 0.9267 0.8906 104.7547 1945.7842 2.4420   
## 29 Tot\_Prns\_in\_HHD\_ACSMOE\_10\_14 0.9267 0.8909 102.8467 1943.9375 2.4379   
## 30 Recent\_Built\_HU\_ACSMOE\_10\_14 0.9266 0.8913 100.9336 1942.0823 2.4338   
## 31 Pop\_under\_5\_CEN\_2010 0.9266 0.8917 99.0495 1940.2753 2.4299   
## 32 Not\_HS\_Grad\_ACSMOE\_10\_14 0.9266 0.892 97.1564 1938.4532 2.4259   
## 33 LAND\_AREA 0.9265 0.8924 95.2406 1936.5933 2.4218   
## 34 NH\_AIAN\_alone\_CEN\_2010 0.9265 0.8927 93.3698 1934.8082 2.4180   
## 35 Med\_HHD\_Inc\_TR\_ACSMOE\_10\_14 0.9264 0.893 91.5393 1933.0900 2.4145   
## 36 avg\_Tot\_Prns\_in\_HHD\_CEN\_2010 0.9264 0.8933 89.7041 1931.3637 2.4109   
## 37 NH\_AIAN\_alone\_ACSMOE\_10\_14 0.9263 0.8936 87.9061 1929.6989 2.4075   
## 38 MrdCple\_Fmly\_HHD\_ACSMOE\_10\_14 0.9263 0.8939 86.1072 1928.0323 2.4041   
## 39 Pop\_under\_5\_ACSMOE\_10\_14 0.9262 0.8942 84.2873 1926.3308 2.4007   
## 40 Rel\_Child\_Under\_6\_CEN\_2010 0.9261 0.8945 82.5096 1924.6989 2.3975   
## 41 Pop\_18\_24\_CEN\_2010 0.9261 0.8948 80.6819 1922.9839 2.3940   
## 42 No\_Plumb\_ACSMOE\_10\_14 0.926 0.8951 78.9176 1921.3735 2.3909   
## 43 College\_ACSMOE\_10\_14 0.9259 0.8954 77.1471 1919.7525 2.3878   
## 44 Males\_ACSMOE\_10\_14 0.9258 0.8956 75.5209 1918.3689 2.3854   
## 45 No\_Health\_Ins\_ACSMOE\_10\_14 0.9257 0.8958 73.8642 1916.9343 2.3829   
## 46 NH\_SOR\_alone\_ACS\_10\_14 0.9256 0.896 72.2065 1915.4970 2.3804   
## 47 Crowd\_Occp\_U\_ACS\_10\_14 0.9255 0.8962 70.5835 1914.1161 2.3780   
## 48 Tot\_Population\_ACS\_10\_14 0.9254 0.8964 68.9493 1912.7158 2.3757   
## 49 Tot\_Prns\_in\_HHD\_ACS\_10\_14 0.9253 0.8967 67.1152 1910.9874 2.3723   
## 50 Female\_No\_HB\_CEN\_2010 0.9253 0.897 65.3867 1909.4317 2.3695   
## 51 Prs\_Blw\_Pov\_Lev\_ACS\_10\_14 0.9252 0.8972 63.6875 1907.9232 2.3669   
## 52 Othr\_Lang\_ACS\_10\_14 0.9251 0.8974 62.0353 1906.4909 2.3645   
## 53 Inst\_GQ\_CEN\_2010 0.925 0.8976 60.3608 1905.0213 2.3620   
## 54 Renter\_Occp\_HU\_ACS\_10\_14 0.9249 0.8978 58.6584 1903.5059 2.3594   
## 55 PUB\_ASST\_INC\_ACS\_10\_14 0.9247 0.898 57.0926 1902.2115 2.3575   
## 56 NH\_AIAN\_alone\_ACS\_10\_14 0.9246 0.8982 55.4867 1900.8511 2.3553   
## 57 Female\_No\_HB\_ACSMOE\_10\_14 0.9245 0.8984 53.8817 1899.4909 2.3532   
## 58 Pop\_45\_64\_ACSMOE\_10\_14 0.9244 0.8985 52.3098 1898.1832 2.3513   
## 59 Crowd\_Occp\_U\_ACSMOE\_10\_14 0.9241 0.8986 51.0398 1897.3610 2.3508   
## 60 Hispanic\_ACSMOE\_10\_14 0.924 0.8987 49.6350 1896.3188 2.3497   
## 61 Pop\_45\_64\_CEN\_2010 0.9237 0.8987 48.3597 1895.4818 2.3492   
## 62 Med\_House\_Value\_TR\_ACSMOE\_10\_14 0.9235 0.8987 47.1722 1894.7817 2.3491   
## 63 College\_ACS\_10\_14 0.9232 0.8988 45.9335 1893.9959 2.3488   
## 64 Rel\_Family\_HHD\_ACSMOE\_10\_14 0.923 0.8988 44.7023 1893.2183 2.3485   
## 65 NonFamily\_HHD\_ACSMOE\_10\_14 0.9228 0.8989 43.3430 1892.2341 2.3476   
## 66 Not\_MrdCple\_HHD\_ACSMOE\_10\_14 0.9226 0.899 41.9312 1891.1645 2.3465   
## 67 Pop\_5\_17\_ACSMOE\_10\_14 0.9223 0.8989 40.8583 1890.6265 2.3469   
## 68 Pop\_5yrs\_Over\_ACSMOE\_10\_14 0.9221 0.899 39.5718 1889.7479 2.3464   
## 69 HHD\_PPL\_Und\_18\_ACS\_10\_14 0.922 0.8991 38.1282 1888.6203 2.3451   
## 70 HHD\_PPL\_Und\_18\_ACSMOE\_10\_14 0.9216 0.899 37.1388 1888.1998 2.3460   
## 71 NH\_SOR\_alone\_CEN\_2010 0.9214 0.899 36.0284 1887.5851 2.3463   
## 72 Tot\_Vacant\_Units\_ACSMOE\_10\_14 0.9211 0.899 34.8147 1886.8056 2.3461   
## 73 Female\_No\_HB\_ACS\_10\_14 0.9208 0.8989 34.0205 1886.6700 2.3478   
## 74 HHD\_Moved\_in\_ACSMOE\_10\_14 0.9204 0.8987 33.3051 1886.6465 2.3499   
## 75 Pop\_25yrs\_Over\_ACSMOE\_10\_14 0.920 0.8986 32.3983 1886.3210 2.3510   
## 76 NH\_NHOPI\_alone\_ACSMOE\_10\_14 0.9197 0.8985 31.4826 1885.9748 2.3522   
## 77 NH\_NHOPI\_alone\_ACS\_10\_14 0.9197 0.8988 29.5061 1884.0104 2.3485   
## 78 Hispanic\_ACS\_10\_14 0.9193 0.8986 28.8337 1884.0259 2.3508   
## 79 NH\_White\_alone\_ACS\_10\_14 0.919 0.8985 27.8690 1883.5906 2.3517   
## 80 NH\_Asian\_alone\_ACSMOE\_10\_14 0.9186 0.8984 26.9069 1883.1530 2.3526   
## 81 Males\_ACS\_10\_14 0.9182 0.8982 26.3913 1883.3769 2.3555   
## 82 Aggregate\_HH\_INC\_ACS\_10\_14 0.9177 0.8979 25.9337 1883.6748 2.3586   
## 83 avg\_Agg\_HH\_INC\_ACS\_10\_14 0.9176 0.8981 24.3420 1882.2807 2.3567   
## 84 Tot\_GQ\_CEN\_2010 0.9172 0.8979 23.5319 1882.0417 2.3582   
## 85 Pov\_Univ\_ACS\_10\_14 0.9168 0.8978 22.8616 1882.0003 2.3604   
## 86 Pop\_1yr\_Over\_ACS\_10\_14 0.9165 0.8976 22.0025 1881.6733 2.3617   
## 87 Pop\_65plus\_ACSMOE\_10\_14 0.916 0.8974 21.4531 1881.7901 2.3644   
## 88 ENG\_VW\_OTHER\_ACSMOE\_10\_14 0.9154 0.8969 21.5704 1882.8599 2.3698   
## 89 ENG\_VW\_API\_ACSMOE\_10\_14 0.9151 0.8969 20.5405 1882.2586 2.3704   
## 90 ENG\_VW\_SPAN\_ACSMOE\_10\_14 0.9146 0.8966 20.1033 1882.5014 2.3734   
## 91 Diff\_HU\_1yr\_Ago\_ACSMOE\_10\_14 0.914 0.8962 19.9762 1883.1727 2.3778   
## 92 MrdCple\_Fmly\_HHD\_CEN\_2010 0.9132 0.8955 20.7747 1885.1312 2.3860   
## 93 Not\_MrdCple\_HHD\_CEN\_2010 0.9129 0.8955 19.7188 1884.4579 2.3864   
## 94 Sngl\_Prns\_HHD\_CEN\_2010 0.9127 0.8956 18.2218 1883.1629 2.3849   
## 95 Med\_HHD\_Inc\_BG\_ACS\_10\_14 0.912 0.8951 18.5739 1884.4430 2.3912   
## 96 NH\_Asian\_alone\_CEN\_2010 0.9112 0.8945 19.0948 1885.9287 2.3981   
## 97 Med\_HHD\_Inc\_BG\_ACSMOE\_10\_14 0.9046 0.8871 37.4972 1915.8360 2.4778   
## 98 NH\_SOR\_alone\_ACSMOE\_10\_14 0.9038 0.8864 38.0962 1917.1620 2.4845   
## 99 NH\_White\_alone\_CEN\_2010 0.9032 0.886 38.3183 1917.9837 2.4896   
## 100 NH\_NHOPI\_alone\_CEN\_2010 0.9025 0.8855 38.6039 1918.8654 2.4949   
## 101 Occp\_U\_NO\_PH\_SRVC\_ACSMOE\_10\_14 0.9017 0.885 38.9911 1919.8532 2.5006   
## 102 TEA\_Mail\_Out\_Mail\_Back\_CEN\_2010 0.9013 0.8848 38.3640 1919.5615 2.5023   
## 103 Occp\_U\_NO\_PH\_SRVC\_ACS\_10\_14 0.9007 0.8845 38.1913 1919.8237 2.5057   
## 104 Tot\_Population\_ACSMOE\_10\_14 0.9001 0.8841 38.1976 1920.2930 2.5098   
## 105 Aggr\_House\_Value\_ACSMOE\_10\_14 0.8995 0.8837 38.2920 1920.8545 2.5142   
## 106 Pop\_1yr\_Over\_ACSMOE\_10\_14 0.8986 0.883 39.2417 1922.4343 2.5218   
## 107 Two\_Plus\_Health\_Ins\_ACSMOE\_10\_14 0.8979 0.8825 39.5772 1923.2462 2.5271   
## 108 Tot\_Occp\_Units\_ACSMOE\_10\_14 0.8971 0.882 40.0889 1924.2482 2.5330   
## 109 Pop\_45\_64\_ACS\_10\_14 0.8962 0.8813 41.0152 1925.7174 2.5403   
## 110 Pop\_25\_44\_ACS\_10\_14 0.896 0.8814 39.6941 1924.5181 2.5392   
## 111 Pop\_18\_24\_ACSMOE\_10\_14 0.8956 0.8813 39.0091 1924.0642 2.5405   
## 112 Pop\_5\_17\_ACS\_10\_14 0.895 0.881 38.8001 1924.1605 2.5436   
## 113 No\_Health\_Ins\_ACS\_10\_14 0.8946 0.8808 38.2039 1923.7958 2.5452   
## 114 One\_Health\_Ins\_ACS\_10\_14 0.8943 0.8808 37.2845 1923.0501 2.5456   
## 115 Deletes\_CEN\_2010 0.8934 0.8801 38.2670 1924.4917 2.5529   
## 116 Renter\_Occp\_HU\_ACSMOE\_10\_14 0.8924 0.8793 39.4510 1926.1333 2.5610   
## 117 avg\_Tot\_Prns\_in\_HHD\_ACSMOE\_10\_14 0.8913 0.8785 40.9756 1928.1263 2.5702   
## 118 avg\_Tot\_Prns\_in\_HHD\_ACS\_10\_14 0.8906 0.878 41.3488 1928.7923 2.5752   
## 119 MLT\_U10p\_ACSMOE\_10\_14 0.8894 0.8771 43.0088 1930.8695 2.5848   
## 120 avg\_Agg\_House\_Value\_ACSMOE\_10\_14 0.8882 0.876 45.1491 1933.4320 2.5960   
## 121 Med\_HHD\_Inc\_TR\_ACS\_10\_14 0.8871 0.8752 46.6675 1935.2687 2.6049   
## 122 Tot\_Housing\_Units\_ACS\_10\_14 0.8855 0.8738 49.8037 1938.8040 2.6194   
## 123 MLT\_U2\_9\_STRC\_ACS\_10\_14 0.8848 0.8733 50.1654 1939.3237 2.6240   
## 124 NH\_Blk\_alone\_CEN\_2010 0.8835 0.8723 52.2645 1941.6596 2.6347   
## 125 NH\_Blk\_alone\_ACS\_10\_14 0.883 0.872 52.0952 1941.5809 2.6374   
## 126 NH\_Blk\_alone\_ACSMOE\_10\_14 0.8818 0.8711 53.9476 1943.5940 2.6471   
## 127 Rel\_Family\_HHD\_CEN\_2010 0.8801 0.8696 57.5878 1947.3977 2.6628   
## 128 Rel\_Child\_Under\_6\_ACS\_10\_14 0.8786 0.8684 60.2505 1950.1326 2.6749   
## 129 Pop\_under\_5\_ACS\_10\_14 0.8782 0.8683 59.6424 1949.5353 2.6760   
## 130 Renter\_Occp\_HU\_CEN\_2010 0.8766 0.8669 62.9008 1952.7902 2.6900   
## 131 Tot\_Housing\_Units\_CEN\_2010 0.8751 0.8656 65.8599 1955.6833 2.7029   
## 132 Recent\_Built\_HU\_ACS\_10\_14 0.8736 0.8644 68.7086 1958.4103 2.7153   
## 133 MrdCple\_Fmly\_HHD\_ACS\_10\_14 0.8724 0.8635 70.3441 1959.9180 2.7236   
## 134 Not\_MrdCple\_HHD\_ACS\_10\_14 0.8705 0.8618 74.7560 1964.0306 2.7409   
## 135 Single\_Unit\_ACS\_10\_14 0.8682 0.8597 80.3414 1969.1430 2.7617   
## 136 Mobile\_Homes\_ACS\_10\_14 0.8661 0.8579 85.0975 1973.3727 2.7796   
## 137 No\_Plumb\_ACS\_10\_14 0.8643 0.8563 88.9259 1976.6699 2.7944   
## 138 PUB\_ASST\_INC\_ACSMOE\_10\_14 0.8622 0.8545 93.6761 1980.7185 2.8120   
## 139 Med\_House\_Value\_BG\_ACSMOE\_10\_14 0.8585 0.8511 103.2049 1998.3238 2.8427   
## 140 ENG\_VW\_ACSMOE\_10\_14 0.8556 0.8484 110.8422 2004.5989 2.8685   
## 141 ENG\_VW\_SPAN\_ACS\_10\_14 0.8546 0.8477 112.2377 2005.4743 2.8750   
## 142 MLT\_U10p\_ACS\_10\_14 0.8509 0.8443 122.2115 2013.4525 2.9072   
## 143 Males\_CEN\_2010 0.8487 0.8424 127.2257 2017.1844 2.9242   
## 144 URBAN\_CLUSTER\_POP\_CEN\_2010 0.8467 0.8407 131.9447 2020.5992 2.9402   
## 145 URBANIZED\_AREA\_POP\_CEN\_2010 0.8463 0.8407 131.3764 2019.7437 2.9406   
## 146 RURAL\_POP\_CEN\_2010 0.8455 0.8403 131.7790 2019.6569 2.9438   
## 147 HHD\_PPL\_Und\_18\_CEN\_2010 0.8428 0.838 138.5297 2024.5489 2.9654   
## 148 ENG\_VW\_INDO\_EURO\_ACS\_10\_14 0.8379 0.8333 152.6766 2034.9634 3.0078   
## 149 ENG\_VW\_INDO\_EURO\_ACSMOE\_10\_14 0.832 0.8277 169.7995 2047.1855 3.0576   
## 150 Tot\_Housing\_Units\_ACSMOE\_10\_14 0.8229 0.8188 197.7799 2066.5143 3.1360   
## 151 Owner\_Occp\_HU\_ACSMOE\_10\_14 0.8184 0.8147 210.2939 2074.4461 3.1711   
## 152 Hispanic\_CEN\_2010 0.8116 0.8083 230.4822 2087.1693 3.2257   
## ---------------------------------------------------------------------------------------------------

print(BWDfit.p$model$coefficients)

## (Intercept) Aggr\_House\_Value\_ACS\_10\_14   
## 7.458566e+01 -2.308826e-08   
## MailBack\_Area\_Count\_CEN\_2010 Vacants\_CEN\_2010   
## -7.508239e-02 8.634077e-02   
## Census\_UAA\_CEN\_2010 FRST\_FRMS\_CEN\_2010   
## 7.226596e-02 1.040532e-01   
## RPLCMNT\_FRMS\_CEN\_2010 avg\_Agg\_House\_Value\_ACS\_10\_14   
## 8.574907e-02 1.939962e-05

### (b) Fit a Lasso model to the data. Use cross-validation to select a value of using the one-standarderror rule and plot the relationship between error and What is the test error?

Best = 0.01142745 Test MSE = 14.92172

library(glmnet)

## Loading required package: Matrix

## Loaded glmnet 3.0-2

library(plotmo)

## Loading required package: Formula

## Loading required package: plotrix

## Loading required package: TeachingDemos

data\_noID\_noMC <- na.omit(data\_noID\_noMC)  
  
grid <- 10^seq(10,-2,length=1000) #these functions automatically plot lambda, so we are going to force it to plot over a range of lambda from 10^10 to 10^-2  
  
x <- model.matrix(Mail\_Return\_Rate\_CEN\_2010~.,data\_noID\_noMC)  
y <-data\_noID\_noMC$Mail\_Return\_Rate\_CEN\_2010  
train <- sample(1:nrow(x), nrow(x)\*.80)  
test <- (-train)  
y.test <- y[test]  
  
# Checks  
dim (x[train,])

## [1] 377 160

length(y[train])

## [1] 377

length(y.test)

## [1] 95

lasso.mod <-glmnet(x[train,], y[train],alpha=1,lambda = grid)  
# glmnet() function standardizes the variables by default so that they are on the same scale.  
# If alpha=0 then a ridge regression model is fit, and if alpha=1 then a lasso model is fit.  
  
dim(coef(lasso.mod ))

## [1] 161 1000

summary(lasso.mod)

## Length Class Mode   
## a0 1000 -none- numeric  
## beta 160000 dgCMatrix S4   
## df 1000 -none- numeric  
## dim 2 -none- numeric  
## lambda 1000 -none- numeric  
## dev.ratio 1000 -none- numeric  
## nulldev 1 -none- numeric  
## npasses 1 -none- numeric  
## jerr 1 -none- numeric  
## offset 1 -none- logical  
## call 5 -none- call   
## nobs 1 -none- numeric

# par(mfrow=c(1,2))  
# plot\_glmnet(lasso.mod, xvar = "lambda", label = 5)  
#   
# plot\_glmnet(lasso.mod, xvar="dev",label=5)  
  
  
#how to choose best lambda  
set.seed(27)  
cv.out=cv.glmnet(x[train,],y[train],alpha=1)  
plot(cv.out, label=TRUE)

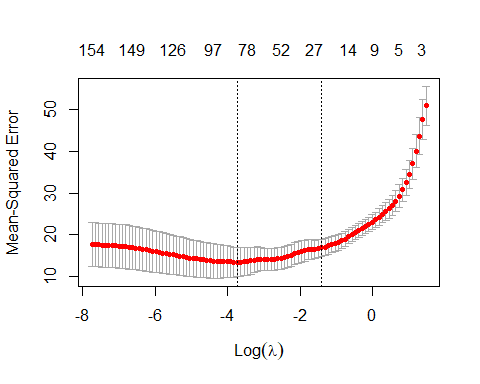
## Warning in plot.window(...): "label" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "label" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "label" is not a  
## graphical parameter  
  
## Warning in axis(side = side, at = at, labels = labels, ...): "label" is not a  
## graphical parameter

## Warning in box(...): "label" is not a graphical parameter

## Warning in title(...): "label" is not a graphical parameter



coef(cv.out)

## 161 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 7.946509e+01  
## (Intercept) .   
## LAND\_AREA .   
## URBANIZED\_AREA\_POP\_CEN\_2010 .   
## URBAN\_CLUSTER\_POP\_CEN\_2010 .   
## RURAL\_POP\_CEN\_2010 .   
## Tot\_Population\_ACS\_10\_14 .   
## Tot\_Population\_ACSMOE\_10\_14 .   
## Males\_CEN\_2010 .   
## Males\_ACS\_10\_14 .   
## Males\_ACSMOE\_10\_14 .   
## Females\_ACSMOE\_10\_14 .   
## Pop\_under\_5\_CEN\_2010 -4.344589e-03  
## Pop\_under\_5\_ACS\_10\_14 .   
## Pop\_under\_5\_ACSMOE\_10\_14 .   
## Pop\_5\_17\_CEN\_2010 .   
## Pop\_5\_17\_ACS\_10\_14 .   
## Pop\_5\_17\_ACSMOE\_10\_14 -2.375267e-03  
## Pop\_18\_24\_CEN\_2010 -2.716658e-03  
## Pop\_18\_24\_ACS\_10\_14 -1.933840e-03  
## Pop\_18\_24\_ACSMOE\_10\_14 -1.918687e-03  
## Pop\_25\_44\_CEN\_2010 .   
## Pop\_25\_44\_ACS\_10\_14 .   
## Pop\_25\_44\_ACSMOE\_10\_14 .   
## Pop\_45\_64\_CEN\_2010 .   
## Pop\_45\_64\_ACS\_10\_14 .   
## Pop\_45\_64\_ACSMOE\_10\_14 .   
## Pop\_65plus\_ACSMOE\_10\_14 1.177929e-02  
## Tot\_GQ\_CEN\_2010 .   
## Inst\_GQ\_CEN\_2010 .   
## Hispanic\_CEN\_2010 -1.485388e-03  
## Hispanic\_ACS\_10\_14 .   
## Hispanic\_ACSMOE\_10\_14 .   
## NH\_White\_alone\_CEN\_2010 .   
## NH\_White\_alone\_ACS\_10\_14 .   
## NH\_White\_alone\_ACSMOE\_10\_14 .   
## NH\_Blk\_alone\_CEN\_2010 .   
## NH\_Blk\_alone\_ACS\_10\_14 .   
## NH\_Blk\_alone\_ACSMOE\_10\_14 .   
## NH\_AIAN\_alone\_CEN\_2010 .   
## NH\_AIAN\_alone\_ACS\_10\_14 .   
## NH\_AIAN\_alone\_ACSMOE\_10\_14 .   
## NH\_Asian\_alone\_CEN\_2010 .   
## NH\_Asian\_alone\_ACS\_10\_14 .   
## NH\_Asian\_alone\_ACSMOE\_10\_14 .   
## NH\_NHOPI\_alone\_CEN\_2010 .   
## NH\_NHOPI\_alone\_ACS\_10\_14 .   
## NH\_NHOPI\_alone\_ACSMOE\_10\_14 .   
## NH\_SOR\_alone\_CEN\_2010 .   
## NH\_SOR\_alone\_ACS\_10\_14 .   
## NH\_SOR\_alone\_ACSMOE\_10\_14 .   
## Pop\_5yrs\_Over\_ACSMOE\_10\_14 .   
## Othr\_Lang\_ACS\_10\_14 .   
## Othr\_Lang\_ACSMOE\_10\_14 .   
## Pop\_25yrs\_Over\_ACSMOE\_10\_14 .   
## Not\_HS\_Grad\_ACS\_10\_14 .   
## Not\_HS\_Grad\_ACSMOE\_10\_14 .   
## College\_ACS\_10\_14 .   
## College\_ACSMOE\_10\_14 .   
## Pov\_Univ\_ACS\_10\_14 .   
## Pov\_Univ\_ACSMOE\_10\_14 .   
## Prs\_Blw\_Pov\_Lev\_ACS\_10\_14 .   
## Prs\_Blw\_Pov\_Lev\_ACSMOE\_10\_14 .   
## One\_Health\_Ins\_ACS\_10\_14 .   
## One\_Health\_Ins\_ACSMOE\_10\_14 .   
## Two\_Plus\_Health\_Ins\_ACS\_10\_14 .   
## Two\_Plus\_Health\_Ins\_ACSMOE\_10\_14 .   
## No\_Health\_Ins\_ACS\_10\_14 .   
## No\_Health\_Ins\_ACSMOE\_10\_14 -2.431591e-03  
## Pop\_1yr\_Over\_ACS\_10\_14 .   
## Pop\_1yr\_Over\_ACSMOE\_10\_14 .   
## Diff\_HU\_1yr\_Ago\_ACS\_10\_14 .   
## Diff\_HU\_1yr\_Ago\_ACSMOE\_10\_14 -4.543715e-04  
## ENG\_VW\_SPAN\_ACS\_10\_14 .   
## ENG\_VW\_SPAN\_ACSMOE\_10\_14 .   
## ENG\_VW\_INDO\_EURO\_ACS\_10\_14 .   
## ENG\_VW\_INDO\_EURO\_ACSMOE\_10\_14 5.766095e-02  
## ENG\_VW\_API\_ACS\_10\_14 .   
## ENG\_VW\_API\_ACSMOE\_10\_14 .   
## ENG\_VW\_OTHER\_ACS\_10\_14 .   
## ENG\_VW\_OTHER\_ACSMOE\_10\_14 .   
## ENG\_VW\_ACSMOE\_10\_14 .   
## Rel\_Family\_HHD\_CEN\_2010 .   
## Rel\_Family\_HHD\_ACS\_10\_14 .   
## Rel\_Family\_HHD\_ACSMOE\_10\_14 .   
## MrdCple\_Fmly\_HHD\_CEN\_2010 .   
## MrdCple\_Fmly\_HHD\_ACS\_10\_14 .   
## MrdCple\_Fmly\_HHD\_ACSMOE\_10\_14 .   
## Not\_MrdCple\_HHD\_CEN\_2010 .   
## Not\_MrdCple\_HHD\_ACS\_10\_14 .   
## Not\_MrdCple\_HHD\_ACSMOE\_10\_14 .   
## Female\_No\_HB\_CEN\_2010 -7.902156e-03  
## Female\_No\_HB\_ACS\_10\_14 .   
## Female\_No\_HB\_ACSMOE\_10\_14 .   
## NonFamily\_HHD\_ACSMOE\_10\_14 .   
## Sngl\_Prns\_HHD\_CEN\_2010 .   
## Sngl\_Prns\_HHD\_ACS\_10\_14 .   
## Sngl\_Prns\_HHD\_ACSMOE\_10\_14 .   
## HHD\_PPL\_Und\_18\_CEN\_2010 .   
## HHD\_PPL\_Und\_18\_ACS\_10\_14 .   
## HHD\_PPL\_Und\_18\_ACSMOE\_10\_14 .   
## Tot\_Prns\_in\_HHD\_ACS\_10\_14 .   
## Tot\_Prns\_in\_HHD\_ACSMOE\_10\_14 .   
## Rel\_Child\_Under\_6\_CEN\_2010 .   
## Rel\_Child\_Under\_6\_ACS\_10\_14 .   
## Rel\_Child\_Under\_6\_ACSMOE\_10\_14 .   
## HHD\_Moved\_in\_ACS\_10\_14 .   
## HHD\_Moved\_in\_ACSMOE\_10\_14 .   
## PUB\_ASST\_INC\_ACS\_10\_14 .   
## PUB\_ASST\_INC\_ACSMOE\_10\_14 .   
## Med\_HHD\_Inc\_BG\_ACS\_10\_14 2.268412e-05  
## Med\_HHD\_Inc\_BG\_ACSMOE\_10\_14 7.694777e-06  
## Med\_HHD\_Inc\_TR\_ACS\_10\_14 .   
## Med\_HHD\_Inc\_TR\_ACSMOE\_10\_14 .   
## Aggregate\_HH\_INC\_ACS\_10\_14 .   
## Aggregate\_HH\_INC\_ACSMOE\_10\_14 .   
## Tot\_Housing\_Units\_CEN\_2010 .   
## Tot\_Housing\_Units\_ACS\_10\_14 .   
## Tot\_Housing\_Units\_ACSMOE\_10\_14 .   
## Tot\_Occp\_Units\_ACSMOE\_10\_14 .   
## Tot\_Vacant\_Units\_ACSMOE\_10\_14 .   
## Renter\_Occp\_HU\_CEN\_2010 -1.144343e-02  
## Renter\_Occp\_HU\_ACS\_10\_14 .   
## Renter\_Occp\_HU\_ACSMOE\_10\_14 .   
## Owner\_Occp\_HU\_ACSMOE\_10\_14 .   
## Single\_Unit\_ACS\_10\_14 .   
## Single\_Unit\_ACSMOE\_10\_14 .   
## MLT\_U2\_9\_STRC\_ACS\_10\_14 -6.409922e-03  
## MLT\_U2\_9\_STRC\_ACSMOE\_10\_14 .   
## MLT\_U10p\_ACS\_10\_14 .   
## MLT\_U10p\_ACSMOE\_10\_14 .   
## Mobile\_Homes\_ACS\_10\_14 .   
## Mobile\_Homes\_ACSMOE\_10\_14 .   
## Crowd\_Occp\_U\_ACS\_10\_14 .   
## Crowd\_Occp\_U\_ACSMOE\_10\_14 .   
## Occp\_U\_NO\_PH\_SRVC\_ACS\_10\_14 .   
## Occp\_U\_NO\_PH\_SRVC\_ACSMOE\_10\_14 .   
## No\_Plumb\_ACS\_10\_14 .   
## No\_Plumb\_ACSMOE\_10\_14 .   
## Recent\_Built\_HU\_ACS\_10\_14 .   
## Recent\_Built\_HU\_ACSMOE\_10\_14 .   
## Med\_House\_Value\_BG\_ACS\_10\_14 .   
## Med\_House\_Value\_BG\_ACSMOE\_10\_14 -5.454673e-06  
## Med\_House\_Value\_TR\_ACS\_10\_14 6.398369e-07  
## Med\_House\_Value\_TR\_ACSMOE\_10\_14 -1.561698e-05  
## Aggr\_House\_Value\_ACS\_10\_14 .   
## Aggr\_House\_Value\_ACSMOE\_10\_14 .   
## MailBack\_Area\_Count\_CEN\_2010 .   
## TEA\_Mail\_Out\_Mail\_Back\_CEN\_2010 .   
## Vacants\_CEN\_2010 -1.267842e-03  
## Deletes\_CEN\_2010 .   
## Census\_UAA\_CEN\_2010 .   
## FRST\_FRMS\_CEN\_2010 6.893292e-03  
## RPLCMNT\_FRMS\_CEN\_2010 3.151580e-02  
## avg\_Tot\_Prns\_in\_HHD\_CEN\_2010 .   
## avg\_Tot\_Prns\_in\_HHD\_ACS\_10\_14 -5.772938e-01  
## avg\_Tot\_Prns\_in\_HHD\_ACSMOE\_10\_14 .   
## avg\_Agg\_HH\_INC\_ACS\_10\_14 .   
## avg\_Agg\_HH\_INC\_ACSMOE\_10\_14 5.525152e-06  
## avg\_Agg\_House\_Value\_ACS\_10\_14 4.790949e-06  
## avg\_Agg\_House\_Value\_ACSMOE\_10\_14 .

bestlam=cv.out$lambda.min  
bestlam

## [1] 0.02373788

#test MSE associated with this value of ??  
lasso.pred=predict(lasso.mod,s=bestlam,newx=x[test,])  
mse= mean((lasso.pred-y.test)^2)

### (c) Examine the coefficients at the one-standard-error value of and comment on your findings.

### What are some important determinants of low census mail return rates?

Based on the top 5 predictors, Home value and Median household income are some of the largest determinants of low census mail return rates.

library(caret)  
  
out=glmnet(x,y,alpha=1,lambda=grid)  
lasso.coef=predict(out,type="coefficients",s=bestlam)[1:161,]  
  
lasso.coef\_no0 = data.frame(as.list(lasso.coef[lasso.coef!=0]))  
  
library(tidyr)

##   
## Attaching package: 'tidyr'

## The following objects are masked from 'package:Matrix':  
##   
## expand, pack, unpack

data\_long <- gather(lasso.coef\_no0, coefficient, value, "X.Intercept.":"avg\_Agg\_House\_Value\_ACSMOE\_10\_14", factor\_key=TRUE)  
data\_long <- mutate(data\_long, absolute\_value = abs(value))  
  
head(data\_long[order(data\_long$absolute\_value), ])

## coefficient value absolute\_value  
## 72 Aggr\_House\_Value\_ACSMOE\_10\_14 5.527176e-09 5.527176e-09  
## 71 Aggr\_House\_Value\_ACS\_10\_14 -2.228245e-08 2.228245e-08  
## 68 Med\_House\_Value\_BG\_ACS\_10\_14 5.340824e-07 5.340824e-07  
## 69 Med\_House\_Value\_BG\_ACSMOE\_10\_14 -7.083832e-06 7.083832e-06  
## 70 Med\_House\_Value\_TR\_ACSMOE\_10\_14 -1.039793e-05 1.039793e-05  
## 51 Med\_HHD\_Inc\_BG\_ACS\_10\_14 1.066567e-05 1.066567e-05

### (d) Based on your findings in (a-c), what can you say about where to direct resources to help

### complete the 2020 Census? Who needs to know this information?

I think it’s safe to say that low income neighborhoods are going to need more help completing the census. The Census Bureau needs to know this along with any organization assisting gain responses.

# Q3: High dimensional simulated data (50 points)

Surveys often ask multiple questions intended to triangulate on a single concept; when this happens the groups of questions have positive correlations between them. The following function will generate a simulated data set with a response variable simY for n observations as a function of g groups of questions, and p\_g responses per group. Variables labeled like g1q3 indicate the third question in group 1. There will also be vector of the “true” betas used to generate simY from a linear combination of each group’s questions (in this case just a sum).

library(MASS)  
data\_gen <- function (n, g, p\_g) {  
simX <<- data.frame(ID=seq(1:n))  
sum\_g <- data.frame(ID=seq(1:n))  
sapply(seq(1:g), function (i) {  
temp\_p <- qr.Q(qr(matrix(rnorm(p\_g^2), p\_g)))  
Sigma\_g <- abs(crossprod(temp\_p, temp\_p\*(p\_g:1)))  
simX\_tmp <- as.data.frame(mvrnorm(n=n, mu=runif(p\_g, min = 0, max = 10), Sigma=Sigma\_g))  
colnames(simX\_tmp) <- paste0("g", i, "q", seq(1:p\_g))  
simX <<- cbind(simX, simX\_tmp)  
sum\_g <<- cbind(sum\_g, rowSums(simX\_tmp))  
})  
colnames(sum\_g) <- c("ID",seq(1:g))  
betas <<- rnorm(g+1, 0, 1)\*10  
error <<- rnorm(n, 0, 5)  
simY <<- rowSums(t(t(sum\_g[2:ncol(sum\_g)]) \* betas[2:length(betas)])) + betas[1] + error  
simData <<- cbind(simX, simY)  
}

### (a) Generate 1000 observations with 3 groups of 3 questions each – we will call this the n1000-g3-

### q3-simData. Examine the correlation or covariance matrices; comment on the correlation or

### covariance within groups of questions vs. between groups of questions (hint: compare g1q1 to

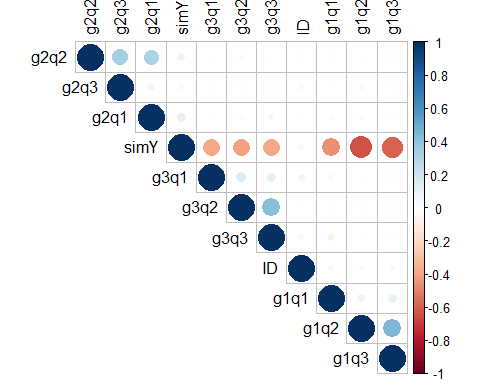
### al the g1 questions vs. g1q1 and all the g2 or g3 questions).

It appears that simy is negatively correlated with group questions 1 & 3, and has a slight positive correlation with group of questions 2. Questions within groups have a slight positive correlation.

set.seed(1000)  
simdata <- data\_gen(n=1000, g=3, p\_g = 3)  
  
res <- cor(simdata)  
  
library(corrplot)

## corrplot 0.84 loaded

corrplot::corrplot(res, type = "upper", order = "hclust",   
 tl.col = "black")



### (b) Use PCR to analyze the n1000-g3-q3-simData. Split the data into 80% training and 20% test.

How many components account for at least 50% of the total variation in X? How many account for at least 80% of the total variation in simY? Plot the cross-validation error as the number of components increase, choose a number of components to include and explain your choice. What is the test error?

The first three components account for atleast 50% of total variation in simY while 7 account for at least 80%. I chose 9 components to test because it explains the most variation, and the test error is 32.70264

require(pls)

## Loading required package: pls

##   
## Attaching package: 'pls'

## The following object is masked from 'package:corrplot':  
##   
## corrplot

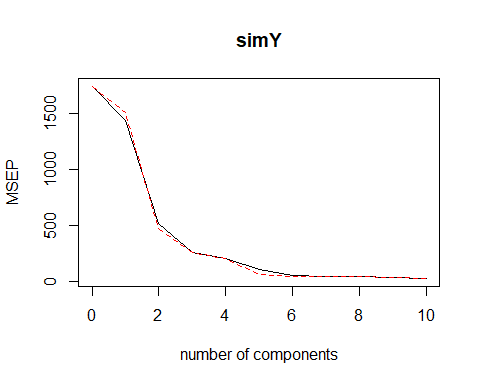
## The following object is masked from 'package:caret':  
##   
## R2

## The following object is masked from 'package:stats':  
##   
## loadings

library(caTools)  
set.seed (27)  
  
split = sample.split(simdata$simY, SplitRatio = 0.80)  
train = subset(simdata, split == TRUE)  
test = subset(simdata, split == FALSE)  
   
pcr\_model <- pls::pcr(simY~., data = train, scale = TRUE, validation = "CV")  
  
summary(pcr\_model)

## Data: X dimension: 800 10   
## Y dimension: 800 1  
## Fit method: svdpc  
## Number of components considered: 10  
##   
## VALIDATION: RMSEP  
## Cross-validated using 10 random segments.  
## (Intercept) 1 comps 2 comps 3 comps 4 comps 5 comps 6 comps  
## CV 41.79 37.94 22.66 16.02 14.3 10.216 6.891  
## adjCV 41.79 38.94 21.68 15.94 14.2 7.932 6.700  
## 7 comps 8 comps 9 comps 10 comps  
## CV 6.667 6.589 5.784 4.984  
## adjCV 6.650 6.582 5.572 4.979  
##   
## TRAINING: % variance explained  
## 1 comps 2 comps 3 comps 4 comps 5 comps 6 comps 7 comps 8 comps  
## X 15.69 30.95 45.43 56.13 65.64 75.02 83.71 89.56  
## simY 17.22 74.47 85.80 88.83 96.88 97.54 97.55 97.61  
## 9 comps 10 comps  
## X 94.86 100.00  
## simY 98.34 98.62

# Plot the cross validation MSE  
validationplot(pcr\_model, val.type="MSEP")



#test error   
y\_test <- test[, 11]  
pcr\_pred <- predict(pcr\_model, test, ncomp = 9)  
mean((pcr\_pred - y\_test)^2)

## [1] 26.86615

### (c) Use PLS to analyze the n1000-g3-q3-simData. How many components account for at least 50%

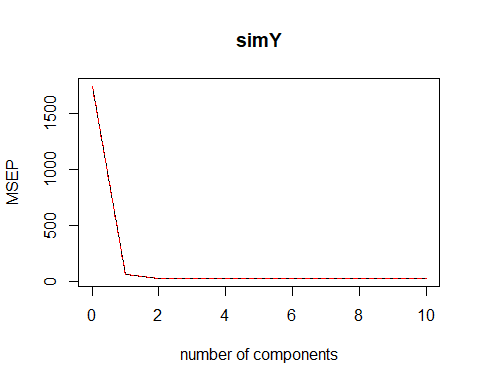
of the total variation in X? How many account for at least 80% of the total variation in simY? Plot the cross-validation error as the number of components increase, choose a number of components to include and explain your choice. What is the test error?

Four components account for atleast 50% and seven account for atleast 80%. I chose 4 components to test because there is not much change after that, and the test error was 25.87317.

set.seed (27)  
  
pls\_model <- pls::plsr(simY~., data = train, scale = TRUE, validation = "CV")  
  
summary(pls\_model)

## Data: X dimension: 800 10   
## Y dimension: 800 1  
## Fit method: kernelpls  
## Number of components considered: 10  
##   
## VALIDATION: RMSEP  
## Cross-validated using 10 random segments.  
## (Intercept) 1 comps 2 comps 3 comps 4 comps 5 comps 6 comps  
## CV 41.79 7.706 5.175 4.997 4.970 4.969 4.970  
## adjCV 41.79 7.668 5.164 4.993 4.967 4.966 4.966  
## 7 comps 8 comps 9 comps 10 comps  
## CV 4.970 4.970 4.970 4.970  
## adjCV 4.966 4.966 4.966 4.966  
##   
## TRAINING: % variance explained  
## 1 comps 2 comps 3 comps 4 comps 5 comps 6 comps 7 comps 8 comps  
## X 14.96 25.00 35.22 45.36 56.88 69.84 76.53 84.05  
## simY 96.71 98.51 98.60 98.62 98.62 98.62 98.62 98.62  
## 9 comps 10 comps  
## X 91.06 100.00  
## simY 98.62 98.62

# Plot the cross validation MSE  
validationplot(pls\_model, val.type="MSEP")



#test error   
y\_test <- test[, 11]  
pcr\_pred <- predict(pls\_model, test, ncomp = 4)  
mean((pcr\_pred - y\_test)^2)

## [1] 22.54284

### (d) Comment on the balance between predictive accuracy and interpretability in PCR and PLS.

Although PCR and PLS are similar, PLS performs better, and has higher accuracy, because it summarizes the predictors that are also associated with the outcome variable. These methods are difficult to interpret because the variables are transformed in orthogonal space and there is no feature selection, such as in Ridge and Lasso regression.

### e) Generate new data with 100 observations and 8 groups of 12 questions each. Use least squares

to model simY on all the predictors. Comment on the overall fit of the model, and whether or not you think the model is useful. Finally, generate new data with 100 observations and 12 groups of 12 questions each. Use least squares to model simY on all the predictors and comment on this process with respect to PCR and PLS.

The test error is huge! And it increases when more groups are included. It is somewhat interpretable though because it gives you pvalues for individual

set.seed (27)  
simdata\_812 <- data\_gen(n=100, g=8, p\_g = 12)  
split = sample.split(simdata\_812$simY, SplitRatio = 0.80)  
train = subset(simdata, split == TRUE)  
test = subset(simdata, split == FALSE)  
  
model812<-lm(simY~., data = train)  
  
summary(model812)

##   
## Call:  
## lm(formula = simY ~ ., data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -17.1220 -3.1631 0.0979 3.6206 14.9103   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.387e+00 1.772e+00 1.911 0.0563 .   
## ID 1.578e-04 6.079e-04 0.260 0.7953   
## g1q1 -1.164e+01 1.245e-01 -93.469 <2e-16 \*\*\*  
## g1q2 -1.178e+01 1.230e-01 -95.750 <2e-16 \*\*\*  
## g1q3 -1.196e+01 1.692e-01 -70.640 <2e-16 \*\*\*  
## g2q1 1.715e+00 1.310e-01 13.093 <2e-16 \*\*\*  
## g2q2 1.949e+00 1.508e-01 12.919 <2e-16 \*\*\*  
## g2q3 1.620e+00 1.267e-01 12.789 <2e-16 \*\*\*  
## g3q1 -8.052e+00 1.143e-01 -70.445 <2e-16 \*\*\*  
## g3q2 -8.154e+00 1.631e-01 -50.011 <2e-16 \*\*\*  
## g3q3 -7.949e+00 1.299e-01 -61.181 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4.931 on 789 degrees of freedom  
## Multiple R-squared: 0.9863, Adjusted R-squared: 0.9861   
## F-statistic: 5665 on 10 and 789 DF, p-value: < 2.2e-16

#test error   
lm812\_pred <- predict(model812, test)  
mean((lm812\_pred - y\_test)^2)

## [1] 3910.032

set.seed (27)  
simdata\_1212 <- data\_gen(n=100, g=12, p\_g = 12)  
split = sample.split(simdata\_1212$simY, SplitRatio = 0.80)  
train = subset(simdata, split == TRUE)  
test = subset(simdata, split == FALSE)  
  
model1212<-lm(simY~., data = train)  
  
summary(model1212)

##   
## Call:  
## lm(formula = simY ~ ., data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -13.7882 -3.1874 0.1955 3.5423 13.4346   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.079e+00 1.768e+00 1.741 0.082 .   
## ID 1.965e-04 6.048e-04 0.325 0.745   
## g1q1 -1.167e+01 1.249e-01 -93.409 <2e-16 \*\*\*  
## g1q2 -1.178e+01 1.247e-01 -94.489 <2e-16 \*\*\*  
## g1q3 -1.192e+01 1.664e-01 -71.630 <2e-16 \*\*\*  
## g2q1 1.792e+00 1.305e-01 13.731 <2e-16 \*\*\*  
## g2q2 1.783e+00 1.525e-01 11.692 <2e-16 \*\*\*  
## g2q3 1.657e+00 1.262e-01 13.130 <2e-16 \*\*\*  
## g3q1 -8.158e+00 1.156e-01 -70.543 <2e-16 \*\*\*  
## g3q2 -8.073e+00 1.683e-01 -47.959 <2e-16 \*\*\*  
## g3q3 -8.003e+00 1.328e-01 -60.263 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4.916 on 789 degrees of freedom  
## Multiple R-squared: 0.9862, Adjusted R-squared: 0.9861   
## F-statistic: 5656 on 10 and 789 DF, p-value: < 2.2e-16

#test error   
lm1212\_pred <- predict(model1212, test)  
mean((lm1212\_pred - y\_test)^2)

## [1] 3573.176