# HW6

David Schultheiss

10/12/2020

##Problem 1

```
library(readr)
crime <- read_csv("/Users/davidschultheiss/Downloads/CommunityCrime.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double()
## )
```

```
## See spec(...) for full column specifications.
```

a)

```
set.seed(1)
x= model.matrix(data= crime, ViolentCrimesPerPop ~ .)[ , -1]
y= crime$ViolentCrimesPerPop
train = sample(1:nrow(crime), .9*nrow(crime))

crime.test = crime[-train, ]
crime.training = crime[train, ]
```

b)

```
lm.fit = lm(data= crime.training, ViolentCrimesPerPop ~ . )
lm.predict = predict(lm.fit, newdata= crime.test, type = 'response')
mean((lm.predict-crime.test$ViolentCrimesPerPop)^2)
```

```
## [1] 0.03439512
```

MSE is .034.

c)

```
library(glmnet)
```

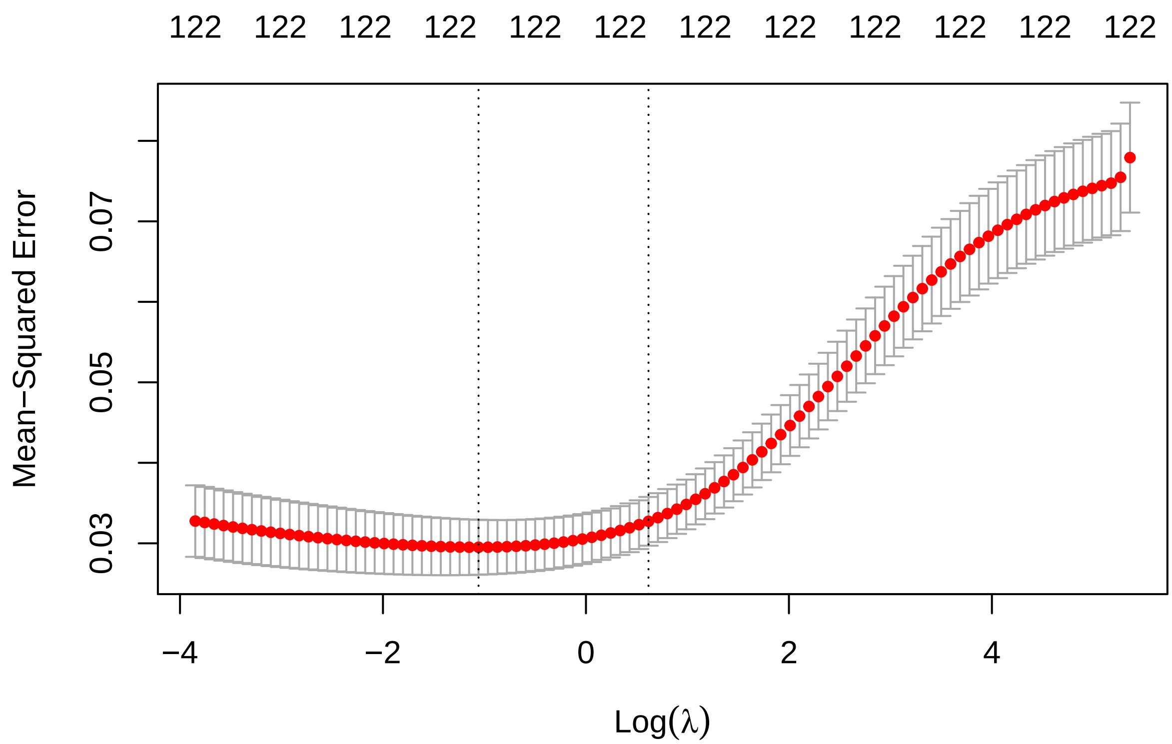
```
## Warning: package 'glmnet' was built under R version 3.6.2
```

```
## Loading required package: Matrix

## Loaded glmnet 4.0-2
```

```r
set.seed(1)
grid1 = 10^seq(10, -2, length= 100)
ridge.mod = glmnet(x[train, ], y[train], alpha= 0, lambda= grid1)

cv = cv.glmnet(x[train, ], y[train], alpha= 0)
plot(cv)
```




```r
best = cv$lambda.min
ridge.best = predict(ridge.mod, newx= x[-train, ], alpha= 0, s= best)
mean((ridge.best - crime.test$ViolentCrimesPerPop)^2)
```

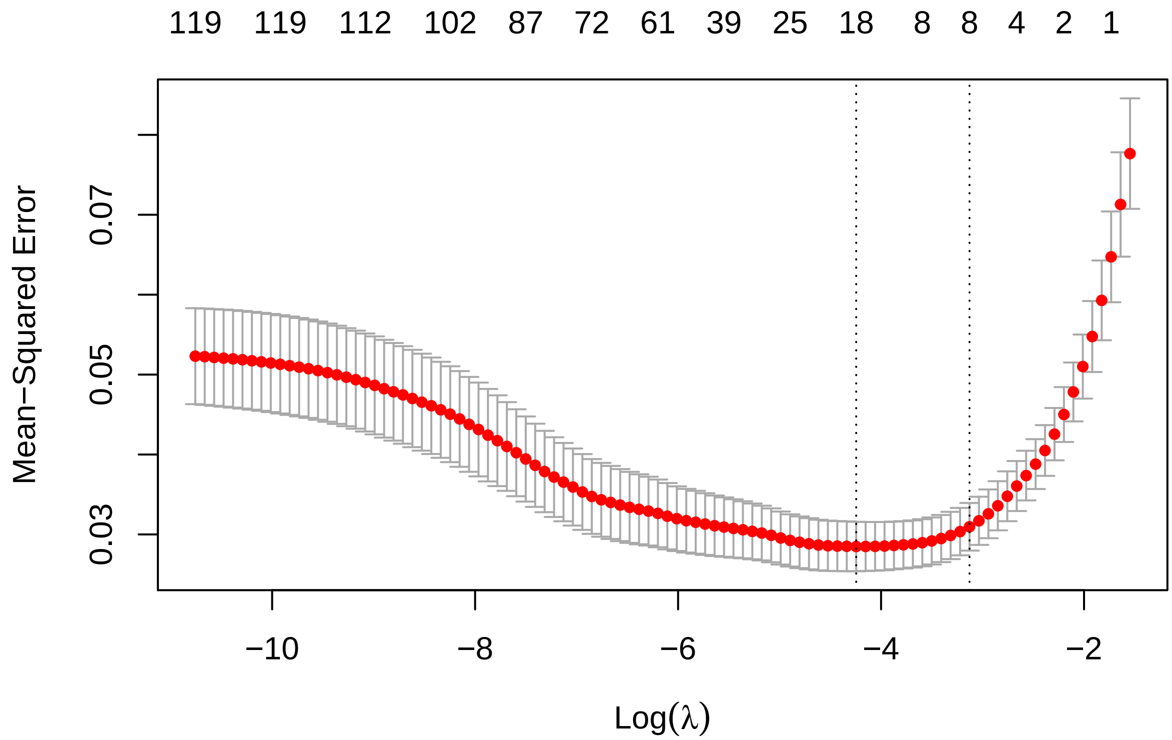
```
## [1] 0.02597019
```

MSE is .026.

d)

```r
set.seed(1)
lasso.mod = glmnet(x[train, ], y[train], alpha= 1, lambda= grid1)

cv = cv.glmnet(x[train, ], y[train], alpha=1)
plot(cv)
```




```
best = cv$lambda.min
lasso.best = predict(lasso.mod, newx= x[-train, ], alpha= 1, s= best)
mean((lasso.best - crime.test$ViolentCrimesPerPop)^2)
```

```
## [1] 0.024541
```

```
lasso.out = glmnet(x, y, alpha=1)
z= predict(lasso.out, type= 'coefficients', s= best)
length(which(z != 0))
```

```
## [1] 16
```

MSE is .025, and the model has 15 non-zero coefficients.

e)

```
library(pls)
```

```
## Warning: package 'pls' was built under R version 3.6.2
```

```
##
## Attaching package: 'pls'
```

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

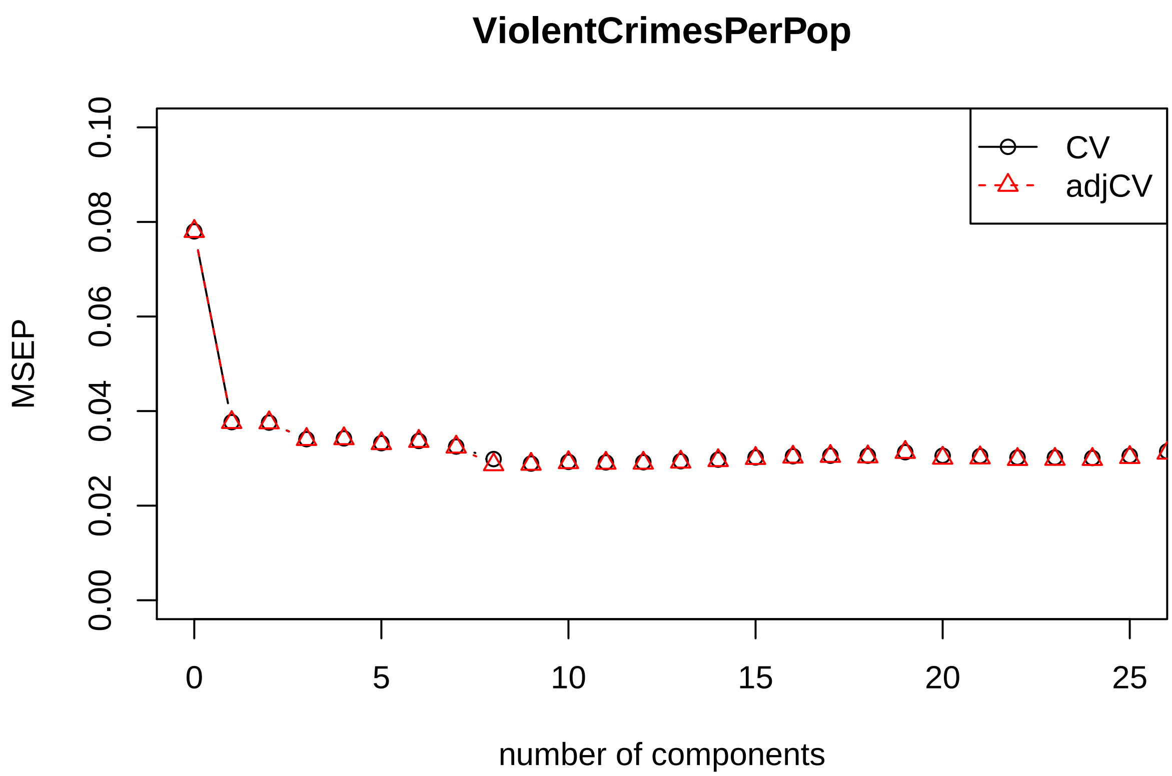
```r
pcr.fit = pcr(ViolentCrimesPerPop ~ ., data= crime, scale= T,
              subset= train, validation= 'CV')
summary(pcr.fit)
```

```
## Data:    X dimension: 287 122
##  Y dimension: 287 1
## Fit method: svdpc
## Number of components considered: 122
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV          0.2793   0.1941   0.1938   0.1846   0.1850   0.1823   0.1835
## adjCV       0.2793   0.1939   0.1938   0.1844   0.1849   0.1820   0.1833
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV      0.1803   0.1728   0.1700    0.1711    0.1707    0.1708    0.1715
## adjCV   0.1800   0.1693   0.1696    0.1706    0.1703    0.1703    0.1710
##        14 comps  15 comps  16 comps  17 comps  18 comps  19 comps  20 comps
## CV       0.1724    0.1738    0.1745    0.1748    0.1748    0.1769    0.1748
## adjCV    0.1719    0.1732    0.1740    0.1745    0.1742    0.1769    0.1734
##        21 comps  22 comps  23 comps  24 comps  25 comps  26 comps  27 comps
## CV       0.1746    0.1737    0.1737    0.1734    0.1747    0.1774    0.1777
## adjCV    0.1735    0.1726    0.1727    0.1726    0.1738    0.1764    0.1767
##        28 comps  29 comps  30 comps  31 comps  32 comps  33 comps  34 comps
## CV       0.1783    0.1789    0.1793    0.1789    0.1796    0.1799    0.1790
## adjCV    0.1773    0.1779    0.1784    0.1777    0.1787    0.1787    0.1779
##        35 comps  36 comps  37 comps  38 comps  39 comps  40 comps  41 comps
## CV       0.1782    0.1787    0.1782    0.1790    0.1799    0.1793    0.1783
## adjCV    0.1767    0.1773    0.1770    0.1779    0.1784    0.1781    0.1768
##        42 comps  43 comps  44 comps  45 comps  46 comps  47 comps  48 comps
## CV       0.1787    0.1792    0.1766    0.1769    0.1777    0.1781    0.1781
## adjCV    0.1774    0.1780    0.1745    0.1746    0.1758    0.1765    0.1767
##        49 comps  50 comps  51 comps  52 comps  53 comps  54 comps  55 comps
## CV       0.1783    0.1781    0.1792    0.1805    0.1830    0.1845    0.1863
## adjCV    0.1764    0.1763    0.1774    0.1787    0.1811    0.1825    0.1843
##        56 comps  57 comps  58 comps  59 comps  60 comps  61 comps  62 comps
## CV       0.1869    0.1886    0.1896    0.1909    0.1896    0.1891    0.1889
## adjCV    0.1850    0.1863    0.1873    0.1887    0.1879    0.1864    0.1864
##        63 comps  64 comps  65 comps  66 comps  67 comps  68 comps  69 comps
## CV       0.1895    0.1899    0.1904    0.1894    0.1907    0.1895    0.1908
## adjCV    0.1872    0.1877    0.1882    0.1876    0.1877    0.1869    0.1880
##        70 comps  71 comps  72 comps  73 comps  74 comps  75 comps  76 comps
## CV       0.1916    0.1918    0.1918    0.1923    0.1936    0.1941    0.1943
## adjCV    0.1889    0.1891    0.1892    0.1895    0.1908    0.1914    0.1917
##        77 comps  78 comps  79 comps  80 comps  81 comps  82 comps  83 comps
## CV       0.1979    0.2004    0.1985    0.1968    0.1978    0.1982    0.1992
## adjCV    0.1957    0.1984    0.1946    0.1931    0.1942    0.1946    0.1960
##        84 comps  85 comps  86 comps  87 comps  88 comps  89 comps  90 comps
## CV       0.1969    0.1969    0.1963    0.1981    0.2000    0.2012    0.2018
## adjCV    0.1934    0.1934    0.1928    0.1945    0.1963    0.1975    0.1982
##        91 comps  92 comps  93 comps  94 comps  95 comps  96 comps  97 comps
## CV       0.2035    0.2050    0.2046    0.2041    0.2025    0.2024    0.2015
## adjCV    0.1998    0.2012    0.2010    0.2003    0.1986    0.1988    0.1972
```

```
##        98 comps  99 comps  100 comps  101 comps  102 comps  103 comps
## CV       0.2027    0.2033     0.2041     0.2014     0.2019     0.2042
## adjCV    0.1985    0.1993     0.2002     0.1966     0.1974     0.1996
##        104 comps  105 comps  106 comps  107 comps  108 comps  109 comps
## CV        0.2045     0.2052     0.2081     0.2153     0.2132     0.2162
## adjCV     0.1999     0.2006     0.2034     0.2102     0.2081     0.2110
##        110 comps  111 comps  112 comps  113 comps  114 comps  115 comps
## CV        0.2159     0.2167     0.2188     0.2248     0.2251     0.2281
## adjCV     0.2107     0.2114     0.2134     0.2191     0.2194     0.2222
##        116 comps  117 comps  118 comps  119 comps  120 comps  121 comps
## CV        0.2308     0.2303     0.2285     0.2224     0.2246     0.2262
## adjCV     0.2248     0.2243     0.2225     0.2166     0.2186     0.2201
##        122 comps
## CV     1.145e+11
## adjCV  1.087e+11
##
## TRAINING: % variance explained
##                      1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## X                      21.93    38.05    48.02    55.41    61.52    65.78
## ViolentCrimesPerPop    51.93    52.29    57.10    57.42    58.94    58.97
##                      7 comps  8 comps  9 comps  10 comps  11 comps  12 comps
## X                      69.22    72.15    75.00     76.93     78.68     80.10
## ViolentCrimesPerPop    60.43    65.51    65.63     65.63     65.86     65.99
##                      13 comps  14 comps  15 comps  16 comps  17 comps  18 comps
## X                       81.46     82.63     83.66     84.53     85.40     86.24
## ViolentCrimesPerPop     66.00     66.05     66.05     66.06     66.06     66.22
##                      19 comps  20 comps  21 comps  22 comps  23 comps  24 comps
## X                       87.04     87.79     88.51     89.19     89.84     90.47
## ViolentCrimesPerPop     66.25     67.29     67.29     67.74     67.74     67.75
##                      25 comps  26 comps  27 comps  28 comps  29 comps  30 comps
## X                       91.05     91.60     92.08     92.55     92.99     93.42
## ViolentCrimesPerPop     67.87     67.87     67.87     67.88     67.90     67.90
##                      31 comps  32 comps  33 comps  34 comps  35 comps  36 comps
## X                       93.82     94.19     94.54     94.87     95.18     95.45
## ViolentCrimesPerPop     68.16     68.19     68.46     68.64     69.24     69.29
##                      37 comps  38 comps  39 comps  40 comps  41 comps  42 comps
## X                       95.73     95.97     96.19     96.40     96.60     96.78
## ViolentCrimesPerPop     69.29     69.36     69.83     69.83     70.29     70.63
##                      43 comps  44 comps  45 comps  46 comps  47 comps  48 comps
## X                       96.96     97.12     97.29     97.44     97.59     97.73
## ViolentCrimesPerPop     70.76     71.56     71.82     71.83     71.84     71.86
##                      49 comps  50 comps  51 comps  52 comps  53 comps  54 comps
## X                       97.87     98.00     98.12     98.23     98.34     98.44
## ViolentCrimesPerPop     72.22     72.23     72.26     72.28     72.29     72.29
##                      55 comps  56 comps  57 comps  58 comps  59 comps  60 comps
## X                       98.54     98.63     98.72     98.80     98.87     98.94
## ViolentCrimesPerPop     72.32     72.32     72.59     72.67     72.67     72.68
##                      61 comps  62 comps  63 comps  64 comps  65 comps  66 comps
## X                       99.00     99.06     99.12     99.17     99.23     99.28
## ViolentCrimesPerPop     73.56     73.64     73.70     73.79     73.94     73.97
##                      67 comps  68 comps  69 comps  70 comps  71 comps  72 comps
## X                       99.32     99.37     99.41     99.45     99.48     99.52
## ViolentCrimesPerPop     74.78     74.90     74.96     75.02     75.02     75.08
##                      73 comps  74 comps  75 comps  76 comps  77 comps  78 comps
```

```
## X                      99.55    99.58    99.61    99.64    99.66    99.69
## ViolentCrimesPerPop    75.21    75.22    75.24    75.29    75.33    75.35
##                     79 comps 80 comps 81 comps 82 comps 83 comps 84 comps
## X                      99.71    99.73    99.75    99.77    99.79    99.81
## ViolentCrimesPerPop    76.66    76.81    76.84    76.86    76.86    77.32
##                     85 comps 86 comps 87 comps 88 comps 89 comps 90 comps
## X                      99.82    99.83    99.85    99.86    99.87    99.88
## ViolentCrimesPerPop    77.45    77.53    77.55    77.68    77.68    77.68
##                     91 comps 92 comps 93 comps 94 comps 95 comps 96 comps
## X                      99.89    99.90    99.91    99.92    99.92    99.93
## ViolentCrimesPerPop    77.77    77.83    77.90    78.14    78.58    78.71
##                     97 comps 98 comps 99 comps 100 comps 101 comps
## X                      99.94    99.94    99.95     99.96     99.96
## ViolentCrimesPerPop    79.45    79.46    79.46     79.54     80.45
##                     102 comps 103 comps 104 comps 105 comps 106 comps
## X                       99.97     99.97     99.97     99.98     99.98
## ViolentCrimesPerPop     80.47     80.63     80.63     80.64     80.71
##                     107 comps 108 comps 109 comps 110 comps 111 comps
## X                       99.98     99.99     99.99     99.99     99.99
## ViolentCrimesPerPop     80.71     80.92     80.94     80.98     81.07
##                     112 comps 113 comps 114 comps 115 comps 116 comps
## X                       99.99    100.00    100.00    100.00    100.00
## ViolentCrimesPerPop     81.10     81.11     81.12     81.12     81.12
##                     117 comps 118 comps 119 comps 120 comps 121 comps
## X                      100.00    100.00    100.00    100.00    100.00
## ViolentCrimesPerPop     81.32     81.72     82.32     82.45     82.54
##                     122 comps
## X                      100.00
## ViolentCrimesPerPop     82.79
```

```r
validationplot(pcr.fit, type='b', legendpos= 'topright', val.type= 'MSEP',
               xlim= c(0,25), ylim= c(0,.1))
```




```
pcr.pred = predict(pcr.fit, crime.test, ncomp= 8)
mean((pcr.pred-crime.test$ViolentCrimesPerPop)^2)
```

```
## [1] 0.02548
```

MSE is .025 with M=8.

f)

```
pls.fit = plsr(ViolentCrimesPerPop ~ ., data= crime, scale= T,
               subset= train, validation= 'CV')
summary(pls.fit)
```

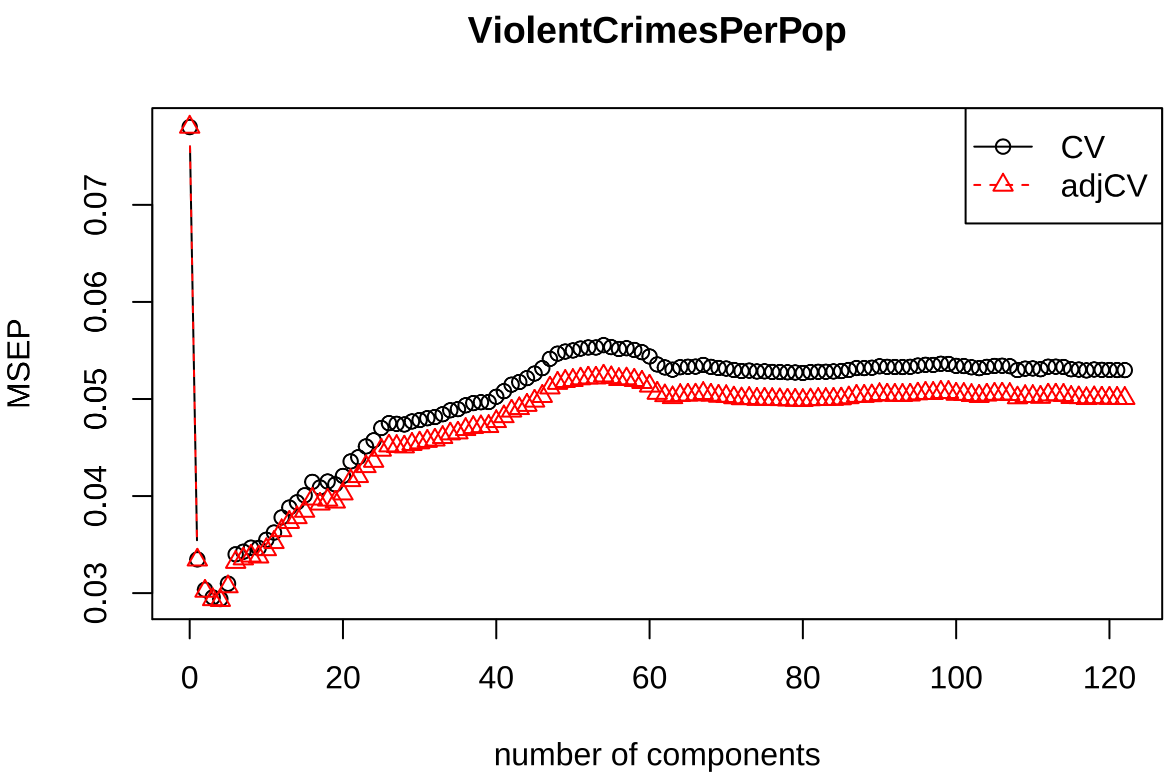
```
## Data:    X dimension: 287 122
##  Y dimension: 287 1
## Fit method: kernelpls
## Number of components considered: 122
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV          0.2793   0.1829   0.1742   0.1719   0.1715   0.1760   0.1844
## adjCV       0.2793   0.1828   0.1738   0.1713   0.1711   0.1751   0.1822
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV      0.1851   0.1862   0.1861    0.1884    0.1904    0.1944    0.1970
## adjCV   0.1831   0.1839   0.1836    0.1857    0.1877    0.1909    0.1931
##        14 comps  15 comps  16 comps  17 comps  18 comps  19 comps  20 comps
```

```
## CV       0.1984   0.2002   0.2036   0.2022   0.2037   0.2030   0.2051
## adjCV    0.1944   0.1961   0.1992   0.1979   0.1990   0.1985   0.2006
##        21 comps  22 comps  23 comps  24 comps  25 comps  26 comps  27 comps
## CV       0.2087   0.2098   0.2124   0.2139   0.2168   0.2180   0.2178
## adjCV    0.2039   0.2050   0.2074   0.2088   0.2115   0.2125   0.2124
##        28 comps  29 comps  30 comps  31 comps  32 comps  33 comps  34 comps
## CV       0.2176   0.2184   0.2187   0.2191   0.2194   0.2201   0.2210
## adjCV    0.2123   0.2129   0.2133   0.2137   0.2140   0.2146   0.2154
##        35 comps  36 comps  37 comps  38 comps  39 comps  40 comps  41 comps
## CV       0.2212   0.2221   0.2226   0.2228   0.2229   0.2241   0.2254
## adjCV    0.2157   0.2165   0.2169   0.2171   0.2172   0.2183   0.2195
##        42 comps  43 comps  44 comps  45 comps  46 comps  47 comps  48 comps
## CV       0.2269   0.2275   0.2283   0.2294   0.2306   0.2327   0.2338
## adjCV    0.2208   0.2214   0.2222   0.2231   0.2242   0.2261   0.2272
##        49 comps  50 comps  51 comps  52 comps  53 comps  54 comps  55 comps
## CV       0.2343   0.2345   0.2349   0.2352   0.2352   0.2357   0.2352
## adjCV    0.2277   0.2278   0.2282   0.2284   0.2284   0.2289   0.2285
##        56 comps  57 comps  58 comps  59 comps  60 comps  61 comps  62 comps
## CV       0.2348   0.2350   0.2346   0.2341   0.2332   0.2314   0.2308
## adjCV    0.2280   0.2282   0.2279   0.2274   0.2266   0.2250   0.2243
##        63 comps  64 comps  65 comps  66 comps  67 comps  68 comps  69 comps
## CV       0.2303   0.2308   0.2309   0.2309   0.2313   0.2309   0.2306
## adjCV    0.2239   0.2244   0.2245   0.2246   0.2249   0.2245   0.2243
##        70 comps  71 comps  72 comps  73 comps  74 comps  75 comps  76 comps
## CV       0.2305   0.2302   0.2299   0.2301   0.2299   0.2299   0.2298
## adjCV    0.2242   0.2239   0.2237   0.2238   0.2236   0.2236   0.2235
##        77 comps  78 comps  79 comps  80 comps  81 comps  82 comps  83 comps
## CV       0.2297   0.2297   0.2296   0.2295   0.2297   0.2298   0.2298
## adjCV    0.2234   0.2234   0.2233   0.2233   0.2234   0.2235   0.2235
##        84 comps  85 comps  86 comps  87 comps  88 comps  89 comps  90 comps
## CV       0.2299   0.2299   0.2302   0.2306   0.2306   0.2307   0.2310
## adjCV    0.2236   0.2236   0.2239   0.2243   0.2243   0.2243   0.2246
##        91 comps  92 comps  93 comps  94 comps  95 comps  96 comps  97 comps
## CV       0.2309   0.2309   0.2308   0.2309   0.2312   0.2313   0.2313
## adjCV    0.2245   0.2245   0.2245   0.2245   0.2248   0.2249   0.2249
##        98 comps  99 comps  100 comps  101 comps  102 comps  103 comps
## CV       0.2316   0.2315    0.2311     0.2311     0.2308     0.2306
## adjCV    0.2252   0.2251    0.2247     0.2247     0.2244     0.2242
##        104 comps  105 comps  106 comps  107 comps  108 comps  109 comps
## CV        0.2309     0.2311     0.2311     0.2310     0.2302     0.2305
## adjCV     0.2245     0.2247     0.2248     0.2247     0.2239     0.2242
##        110 comps  111 comps  112 comps  113 comps  114 comps  115 comps
## CV        0.2305     0.2304     0.2309     0.2309     0.2309     0.2303
## adjCV     0.2242     0.2240     0.2245     0.2245     0.2245     0.2240
##        116 comps  117 comps  118 comps  119 comps  120 comps  121 comps
## CV        0.2303     0.2301     0.2303     0.2302     0.2302     0.2302
## adjCV     0.2239     0.2237     0.2239     0.2238     0.2238     0.2238
##        122 comps
## CV        0.2301
## adjCV     0.2238
##
## TRAINING: % variance explained
##                      1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## X                       21.6    30.04    36.96    49.56    55.58    58.31
```

```
## ViolentCrimesPerPop     58.7    64.96    67.75    68.54    69.93    72.14
##                      7 comps  8 comps  9 comps  10 comps  11 comps  12 comps
## X                      63.69    66.90    69.13     71.05     74.31     75.29
## ViolentCrimesPerPop    72.78    73.81    74.79     75.45     75.79     76.80
##                      13 comps  14 comps  15 comps  16 comps  17 comps  18 comps
## X                       76.42     78.25     79.63     80.45     81.20     81.69
## ViolentCrimesPerPop     77.42     77.83     78.33     78.70     78.98     79.46
##                      19 comps  20 comps  21 comps  22 comps  23 comps  24 comps
## X                       82.80     83.92     84.66     85.57     86.14     86.79
## ViolentCrimesPerPop     79.72     79.94     80.15     80.28     80.44     80.57
##                      25 comps  26 comps  27 comps  28 comps  29 comps  30 comps
## X                       87.33     87.72     88.15     88.70     89.05     89.47
## ViolentCrimesPerPop     80.71     80.87     80.98     81.05     81.13     81.18
##                      31 comps  32 comps  33 comps  34 comps  35 comps  36 comps
## X                       89.93     90.37     90.76     91.06     91.42     91.84
## ViolentCrimesPerPop     81.21     81.24     81.27     81.31     81.35     81.39
##                      37 comps  38 comps  39 comps  40 comps  41 comps  42 comps
## X                       92.37     92.75     93.22     93.64     93.96     94.24
## ViolentCrimesPerPop     81.43     81.48     81.51     81.54     81.59     81.65
##                      43 comps  44 comps  45 comps  46 comps  47 comps  48 comps
## X                       94.66     94.98     95.21     95.40     95.61     95.90
## ViolentCrimesPerPop     81.69     81.74     81.79     81.84     81.89     81.92
##                      49 comps  50 comps  51 comps  52 comps  53 comps  54 comps
## X                       96.18     96.40     96.57     96.83     97.07     97.17
## ViolentCrimesPerPop     81.96     82.02     82.08     82.12     82.16     82.20
##                      55 comps  56 comps  57 comps  58 comps  59 comps  60 comps
## X                       97.28     97.36     97.51     97.68     97.79      97.9
## ViolentCrimesPerPop     82.24     82.30     82.33     82.35     82.38      82.4
##                      61 comps  62 comps  63 comps  64 comps  65 comps  66 comps
## X                       98.03     98.13     98.23     98.33     98.40     98.48
## ViolentCrimesPerPop     82.42     82.43     82.44     82.45     82.45     82.46
##                      67 comps  68 comps  69 comps  70 comps  71 comps  72 comps
## X                       98.56     98.63     98.73     98.79     98.88     98.96
## ViolentCrimesPerPop     82.47     82.47     82.48     82.48     82.49     82.49
##                      73 comps  74 comps  75 comps  76 comps  77 comps  78 comps
## X                       99.03     99.09     99.15     99.20     99.26     99.32
## ViolentCrimesPerPop     82.49     82.50     82.50     82.51     82.51     82.52
##                      79 comps  80 comps  81 comps  82 comps  83 comps  84 comps
## X                       99.36     99.41     99.45     99.48     99.51     99.54
## ViolentCrimesPerPop     82.52     82.53     82.53     82.53     82.53     82.54
##                      85 comps  86 comps  87 comps  88 comps  89 comps  90 comps
## X                       99.56     99.60     99.62     99.65     99.68     99.70
## ViolentCrimesPerPop     82.54     82.54     82.54     82.55     82.55     82.55
##                      91 comps  92 comps  93 comps  94 comps  95 comps  96 comps
## X                       99.72     99.75     99.76     99.78     99.80     99.82
## ViolentCrimesPerPop     82.56     82.56     82.56     82.57     82.57     82.57
##                      97 comps  98 comps  99 comps  100 comps  101 comps
## X                       99.84     99.85     99.86      99.87      99.88
## ViolentCrimesPerPop     82.57     82.58     82.58      82.58      82.58
##                      102 comps  103 comps  104 comps  105 comps  106 comps
## X                        99.89      99.91      99.92      99.92      99.93
## ViolentCrimesPerPop      82.59      82.59      82.59      82.60      82.60
##                      107 comps  108 comps  109 comps  110 comps  111 comps
## X                        99.94      99.94      99.95      99.95      99.96
```

```
## ViolentCrimesPerPop        82.61      82.62      82.64      82.65      82.67
##                        112 comps  113 comps  114 comps  115 comps  116 comps
## X                          99.96      99.97      99.97      99.98      99.98
## ViolentCrimesPerPop        82.70      82.71      82.73      82.75      82.78
##                        117 comps  118 comps  119 comps  120 comps  121 comps
## X                          99.98      99.99      99.99     100.00     100.00
## ViolentCrimesPerPop        82.79      82.79      82.79      82.79      82.79
##                        122 comps
## X                         100.00
## ViolentCrimesPerPop        82.79
```

```
validationplot(pls.fit, type='b', legendpos= 'topright', val.type= 'MSEP')
```




```
pls.pred = predict(pls.fit, crime.test, ncomp= 3)
mean((pls.pred-crime.test$ViolentCrimesPerPop)^2)
```

```
## [1] 0.02444648
```

MSE is .024 with M=3.

g) The error term is largest (.034) using the linear regression, and smallest (.024) using PLS. Our MSE values are low, so violent crime can be predicted fairly accurately.

##Problem 2

```
set.seed(1)
n <- 80
p <- 50
X <- matrix(0, nrow=n, ncol=p)

for(j in 1:p) {
  X[,j] <- runif(n=n, min=0, max=1) }
beta0 <- 2
betas <- rep(0,p)
betas[1:3] <- c(1,2,3)
betas <- matrix(betas,ncol=1)
Y <- beta0 + X %*% betas + rnorm(n,0,1)
```

a) The code assigns 80(n) observations of 50 predictors(p) and gives each observation a random number between 0 and 1. The true values of the betas are 1,2,3, and all 0s after that. The first 3 Ys are significantly connected to Xj for j=[1,3]

b)

```
lm.fit = lm(Y~X)
summary(lm.fit)
```

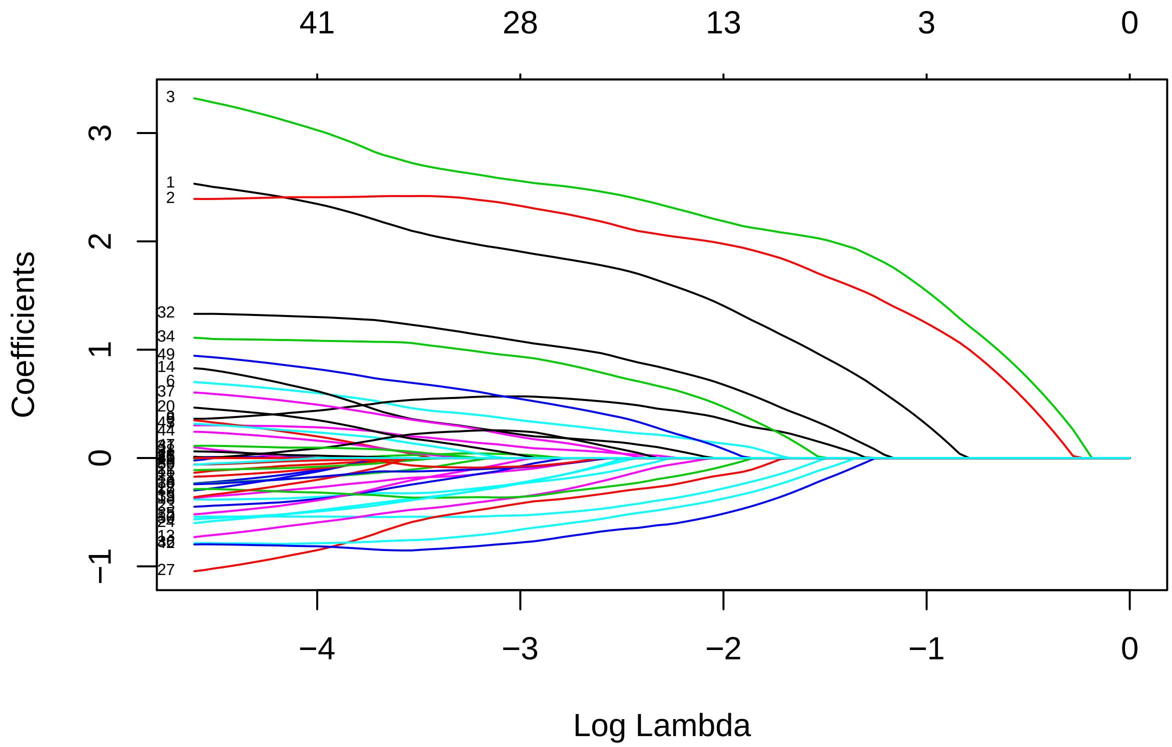
```
##
## Call:
## lm(formula = Y ~ X)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.08591 -0.50799  0.04112  0.39039  1.16323
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.188898   1.960629   0.606 0.548976
## X1           2.978243   0.665325   4.476 0.000108 ***
## X2           2.367923   0.673685   3.515 0.001466 **
## X3           3.817868   0.785191   4.862 3.72e-05 ***
## X4          -0.402415   0.502994  -0.800 0.430193
## X5           0.045005   0.583992   0.077 0.939102
## X6           0.873704   0.529626   1.650 0.109803
## X7           0.398627   0.540290   0.738 0.466563
## X8           0.307201   0.695992   0.441 0.662207
## X9           0.532805   0.578024   0.922 0.364255
## X10         -0.404113   0.635711  -0.636 0.529966
## X11         -0.062232   0.477435  -0.130 0.897193
## X12         -0.525872   0.497160  -1.058 0.298900
## X13         -0.956815   0.516955  -1.851 0.074399 .
## X14          1.027485   0.740168   1.388 0.175656
## X15         -0.330725   0.548093  -0.603 0.550926
## X16         -0.093900   0.505976  -0.186 0.854064
## X17         -0.493808   0.593929  -0.831 0.412527
## X18         -0.374466   0.584461  -0.641 0.526746
## X19         -0.536495   0.598690  -0.896 0.377571
## X20          0.704253   0.548730   1.283 0.209505
```

```
## X21          -0.265098   0.490391  -0.541 0.592921
## X22          -0.159632   0.510765  -0.313 0.756870
## X23          -0.603407   0.581655  -1.037 0.308124
## X24          -0.799255   0.504992  -1.583 0.124334
## X25          -0.757908   0.555598  -1.364 0.183021
## X26           0.223113   0.555057   0.402 0.690658
## X27          -1.374018   0.628133  -2.187 0.036918 *
## X28          -0.244653   0.592533  -0.413 0.682722
## X29          -0.185092   0.676265  -0.274 0.786257
## X30          -0.729229   0.487302  -1.496 0.145339
## X31           0.388400   0.587833   0.661 0.513999
## X32           1.440447   0.506018   2.847 0.008032 **
## X33          -0.735947   0.657497  -1.119 0.272190
## X34           1.137623   0.599581   1.897 0.067777 .
## X35          -0.279902   0.483165  -0.579 0.566854
## X36          -0.851994   0.546558  -1.559 0.129883
## X37           0.691944   0.550577   1.257 0.218868
## X38           0.142061   0.561277   0.253 0.801974
## X39           0.006078   0.544237   0.011 0.991167
## X40           0.184064   0.518447   0.355 0.725136
## X41          -0.207373   0.584457  -0.355 0.725297
## X42          -0.720693   0.592112  -1.217 0.233357
## X43           0.520515   0.505283   1.030 0.311457
## X44           0.355246   0.460136   0.772 0.446334
## X45          -0.265706   0.586495  -0.453 0.653891
## X46           0.174564   0.595110   0.293 0.771359
## X47           0.171293   0.504815   0.339 0.736815
## X48           0.089166   0.478919   0.186 0.853598
## X49           1.125379   0.598809   1.879 0.070278 .
## X50          -0.154330   0.462262  -0.334 0.740889
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8936 on 29 degrees of freedom
## Multiple R-squared:  0.8793, Adjusted R-squared:  0.6711
## F-statistic: 4.224 on 50 and 29 DF,  p-value: 4.377e-05
```

The estimates don't exactly match but are able to determine the first three as significant.

c)

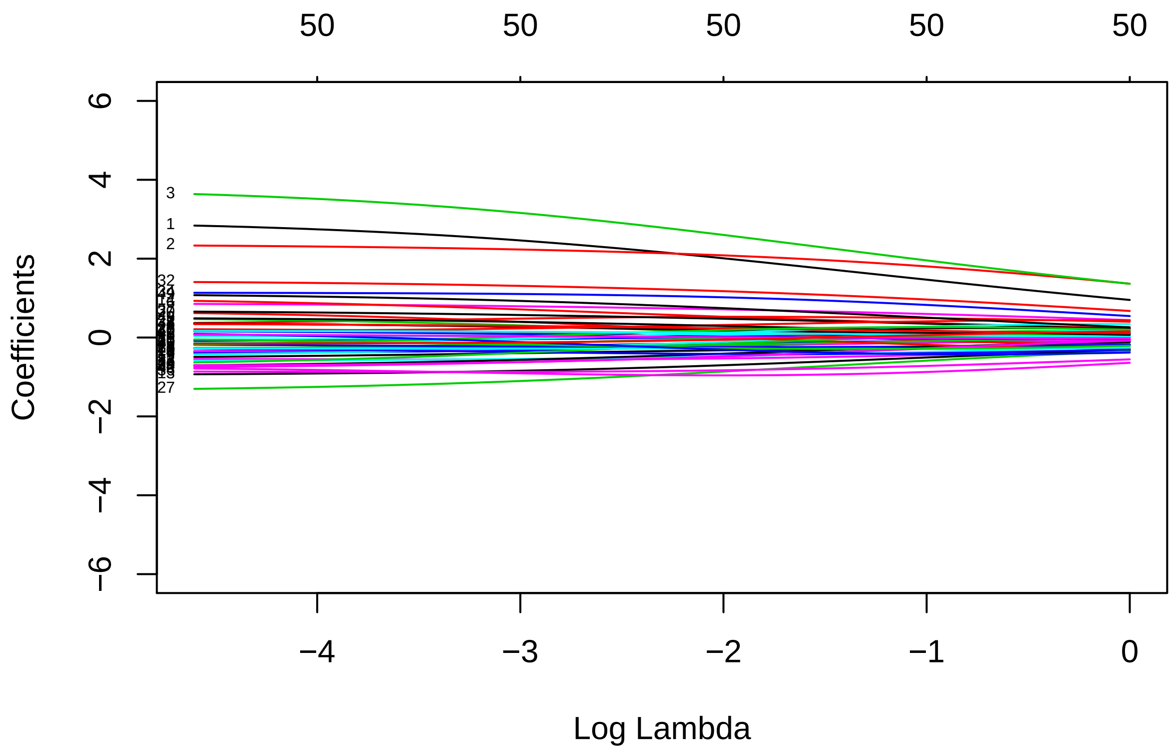
```r
grid1 = 10^seq(-2, 0, length= 100)
lasso.mod = glmnet(X, Y, alpha= 1, lambda= grid1)
plot(lasso.mod, xvar= 'lambda', label= T)
```




The first three coefficients are significantly higher. Three is also correctly estimated as higher than our 2 and 1 coefficients. On the other hand, 1 is a little further away from its true value.

d)

```
ridge.mod = glmnet(X, Y, alpha= 0, lambda= grid1)
plot(ridge.mod, xvar= 'lambda', label= T, ylim = c(-6, 6))
```

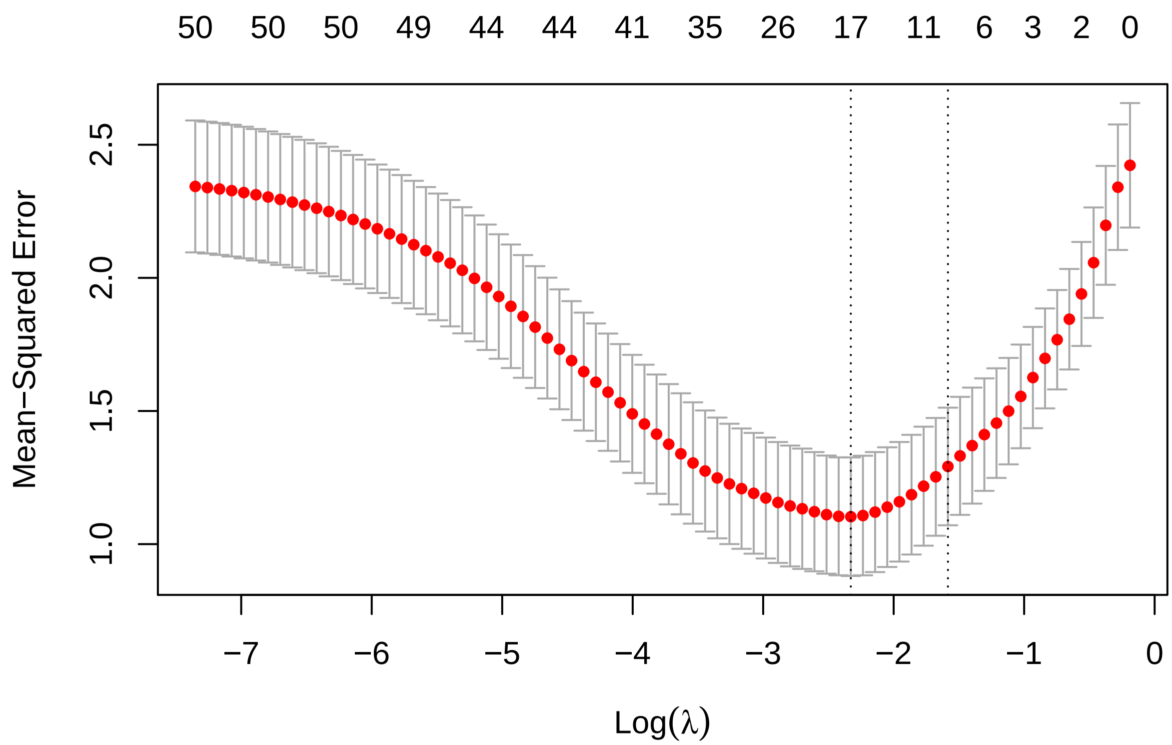



We

can see that Ridge regression is not performing any variable selection, so the coefficients of the insignificant variables approach 0 as lambda increases. With Lasso, the coeffiecients actually reach 0 as lambda increases.

e)

```r
cv = cv.glmnet(X, Y, alpha=1)
plot(cv)
```




```r
best = cv$lambda.min

lasso.out = glmnet(X, Y, alpha=1)
z= predict(lasso.out, type= 'coefficients', s= best)
```

It found all three of the variables significant. We also had 14 false positives.