Labs3

Jakub Kuciński

2022-05-29

Task 1

First let's calculate MSE on single coordinate:

$$\begin{aligned}
\mathbb{E}\|\hat{\beta}_i - \beta_i\|^2 &= \mathbb{E}\|((X'X + \lambda I)^{-1}X'Y)_i - \beta_i\|^2 = &(1)\\
&= \mathbb{E}\|((1+\lambda)^{-1}X'Y)_i - \beta_i\|^2 = &(2)\\
&= \mathbb{E}\|((1+\lambda)^{-1}X'(X\beta + \epsilon))_i - \beta_i\|^2 = &(3)\\
&= \mathbb{E}\|((1+\lambda)^{-1}X'(X\beta + \epsilon))_i - \beta_i\|^2 = &(4)\\
&= \mathbb{E}\|((1+\lambda)^{-1}(X'X\beta + X'\epsilon))_i - \beta_i\|^2 = &(5)\\
&= \mathbb{E}\|((1+\lambda)^{-1}(\beta + Z))_i - \beta_i\|^2 = &(6)\\
&= \mathbb{E}\left\|\frac{Z_i}{\lambda+1} - \frac{\lambda\beta_i}{\lambda+1}\right\|^2 = &(7)\\
&= \mathbb{E}\left[\frac{1}{(\lambda+1)^2}Z_i^2\right] - 2\mathbb{E}\left[\frac{\lambda}{(\lambda+1)^2}Z_i\beta_i\right] + \mathbb{E}\left[\frac{\lambda^2}{(\lambda+1)^2}\beta_i^2\right] = &(8)\\
&= \frac{1}{(\lambda+1)^2}\sigma^2 - 2\cdot 0 + \frac{\lambda^2}{(\lambda+1)^2}\beta_i^2 = &(9)\\
&= \frac{1}{(\lambda+1)^2}\sigma^2 + \frac{\lambda^2}{(\lambda+1)^2}\beta_i^2 &(10)
\end{aligned}$$

Now moving to the vector norm we get:

$$\mathbb{E}\|\hat{\beta} - \beta\|^2 = \frac{p}{(\lambda+1)^2}\sigma^2 + \frac{\lambda^2}{(\lambda+1)^2}\|\beta\|^2$$

We can find minimum of MSE by calculating the derivative of MSE with respect to λ:

$$\frac{\partial}{\partial\lambda}\mathbb{E}\|\hat{\beta} - \beta\|^2 = \frac{\partial}{\partial\lambda}\left(\frac{p}{(\lambda+1)^2}\sigma^2 + \frac{\lambda^2}{(\lambda+1)^2}\|\beta\|^2\right) = \frac{-2p}{(\lambda+1)^3}\sigma^2 + \frac{2\lambda}{(\lambda+1)^3}\|\beta\|^2 = 0$$

Now we can get the lambda that minimizes MSE:

$$\lambda = \frac{p\sigma^2}{\|\beta\|^2}$$

For $k = 20, 100, 200$ we get optimal values of lambda equal to: 3.877551, 0.7755102, 0.3877551, MSE for $\beta_i = 3.5$ equal to: 7.7839324, 2.6542476, 1.4756163 and MSE for $\beta_i = 0$ equal to:0.0420336, 0.317215, 0.5192474.

The expression for bias can also be easily obtained:

$$\text{Bias}(\hat{\beta}_i) = \mathbb{E}[\hat{\beta}_i - \beta_i] = \mathbb{E}\left[\frac{Z_i}{\lambda+1} - \frac{\lambda\beta_i}{\lambda+1}\right] = \mathbb{E}\left[\frac{Z_i}{\lambda+1}\right] - \mathbb{E}\left[\frac{\lambda\beta_i}{\lambda+1}\right] = 0 - \frac{\lambda\beta_i}{\lambda+1} = -\frac{\lambda}{\lambda+1}\beta_i$$

For $\beta_i = 0$ bias is actually zero. For $\beta_i = 3.5$ and optimal values of λ for $k = 20, 100, 200$ we obtain expected biases: -2.7824268, -1.5287356, -0.9779412.

The expression for variance is as follows:

$$\begin{aligned}
\text{Var}(\hat{\beta}_i) &= \frac{{\beta_i}^2 + 1}{(1+\lambda)^2} - (\text{Bias}(\hat{\beta}_i) + \beta_i)^2 = && (11)\\
&= \frac{{\beta_i}^2 + 1}{(1+\lambda)^2} - \left(-\frac{\lambda}{\lambda+1}\beta_i + \beta_i\right)^2 = && (12)\\
&= \frac{{\beta_i}^2 + 1}{(1+\lambda)^2} - \left(-\frac{1}{\lambda+1}\beta_i\right)^2 = && (13)\\
&= \frac{{\beta_i}^2 + 1}{(1+\lambda)^2} - \frac{\beta_i^2}{(1+\lambda)^2} = \frac{1}{(1+\lambda)^2} && (14)
\end{aligned}$$

The variance is only depended on the value of λ so it is equal for all values of β_i. For optimal values of λ for $k = 20, 100, 200$ we obtain the variances: 0.0420336, 0.317215, 0.5192474.

The expression for $\hat{\beta}_{OLS}$ is as follows:

$$\hat{\beta}_{OLS} = X'Y = X'(X\beta + \epsilon) = \beta + X'\epsilon$$

We can express $\hat{\beta}_{\text{Ridge}}$ in terms of $\hat{\beta}_{OLS}$:

$$\hat{\beta}_{\text{Ridge}} = \frac{\hat{\beta}_{OLS}}{1+\lambda}$$

```
##     k emp bias OLS b3.5 emp bias OLS b0 emp var OLS b3.5 emp var OLS b0
## 1  20      0.0081978418     0.001048655        1.0030045      0.9993975
## 2 100      0.0010479317     0.000107460        0.9989392      0.9991407
## 3 200     -0.0009291394    -0.002507764        0.9987880      1.0001663
##   emp mse OLS b3.5 emp mse OLS b0 emp bias Ridge b3.5 emp bias Ridge b0
## 1        1.0027956      0.9994117          -2.7807460      2.149962e-04
## 2        0.9991031      0.9990860          -1.5281454      6.052347e-05
## 3        0.9988974      1.0001549          -0.9786107     -1.807065e-03
##   emp var Ridge b3.5 emp var Ridge b0 emp mse Ridge b3.5 emp mse Ridge b0
## 1         0.04215987       0.04200825           7.774697       0.04200885
## 2         0.31687846       0.31694236           2.652159       0.31692502
## 3         0.51861807       0.51933377           1.476353       0.51932786
```

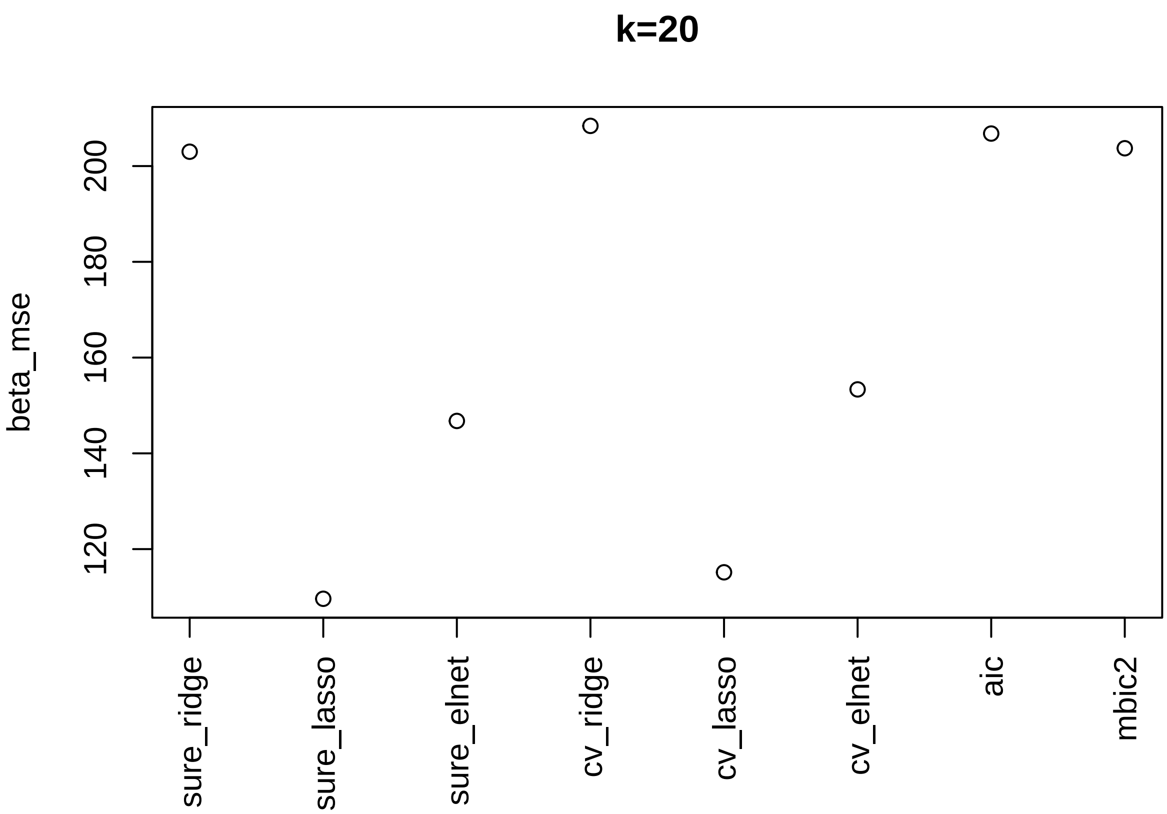
We've already shown that $\text{Bias}(\hat{\beta}_i) = -\frac{\lambda}{\lambda+1}\beta_i$ which is zero for OLS (as $\lambda = 0$), for ridge regression for $\beta_i = 0$ it is also zero, but for $\beta_i = 3.5$ and optimal values of λ for $k = 20, 100, 200$ the expected biases are: -2.7824268, -1.5287356, -0.9779412. Results from simulations are on line with those results. Experimental biases for OLS are around zero as OLS is not biased.

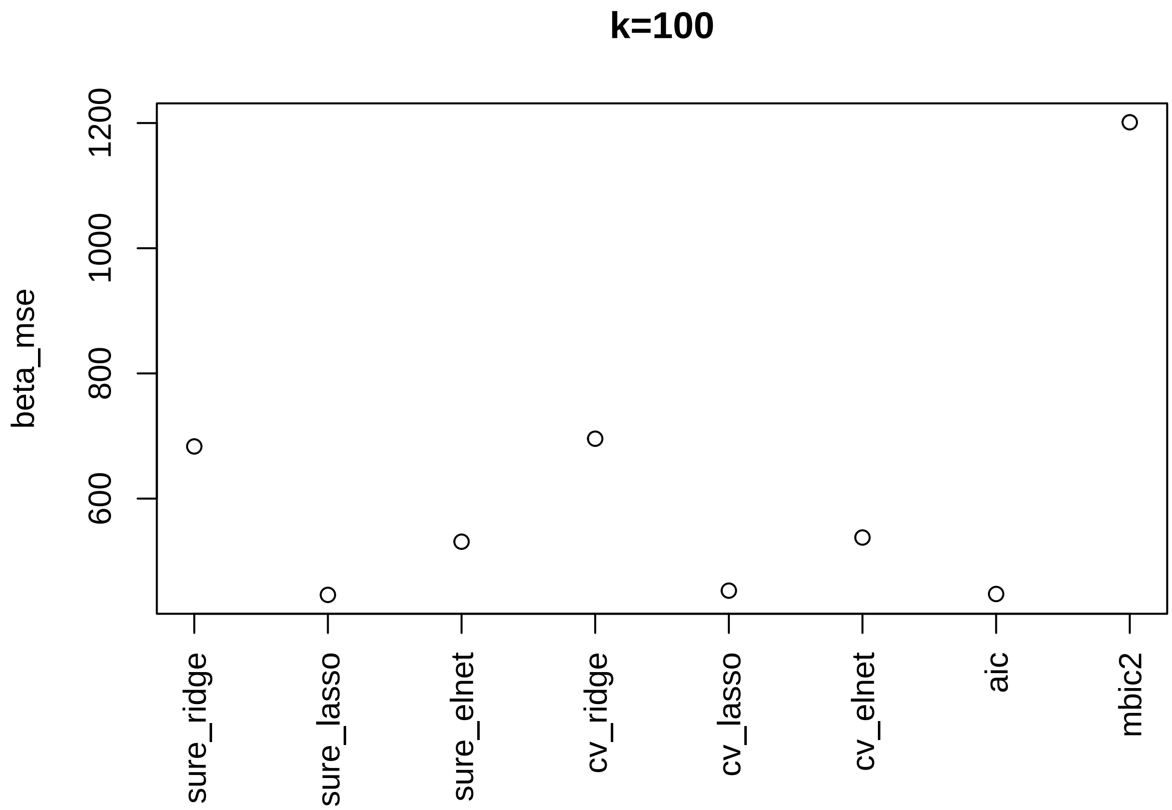
The expression for variance was $\text{Var}(\hat{\beta}_i) = \frac{1}{(1+\lambda)^2}$ so for OLS with $\lambda = 0$ we should get 1 which is the case in our experiments. Empirical values of variance for ridge are also similar to ones calculated theoretically.

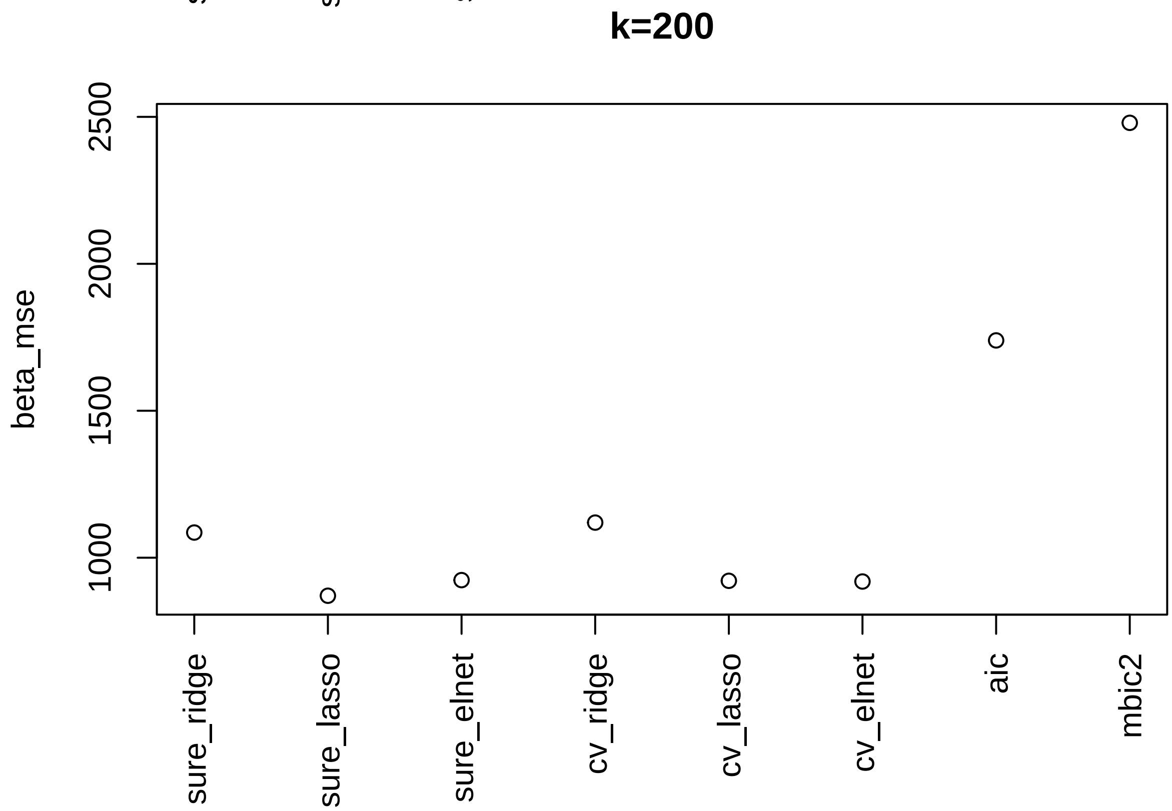
The expression for mean square error of the estimation of β was $\mathbb{E}\|\hat{\beta}_i - \beta_i\|^2 = \frac{1}{(\lambda+1)^2}\sigma^2 + \frac{\lambda^2}{(\lambda+1)^2}\beta_i^2$. For OLS $\lambda = 0$ so it is simply equal to $\sigma^2 = 1$. Empirical results are similar to this value. For $k = 20, 100, 200$ and optimal values of λ the MSE for $\beta_i = 3.5$ are: 7.7839324, 2.6542476, 1.4756163 and MSE for $\beta_i = 0$ are:0.0420336, 0.317215, 0.5192474. Experimental results are once again on line with theoretical estimations.

Task 2

```
##     k beta_mse_sure_ridge beta_mse_sure_lasso beta_mse_sure_elnet
## 1  20            202.9975            109.6286            146.7666
## 2 100            683.3379            446.2144            531.1675
## 3 200           1085.6934            870.6687            923.7988
##   beta_mse_crossval_ridge beta_mse_crossval_lasso beta_mse_crossval_elnet
## 1                208.3915                115.1555                153.3585
## 2                695.6789                452.9720                537.8816
## 3               1119.5004                921.4461                919.0318
##   beta_mse_ols beta_mse_aic beta_mse_mbic2
## 1     18401.42     206.7919      203.7185
## 2     17873.44     447.6229     1201.1993
## 3     17821.23    1739.3910     2479.6403
```







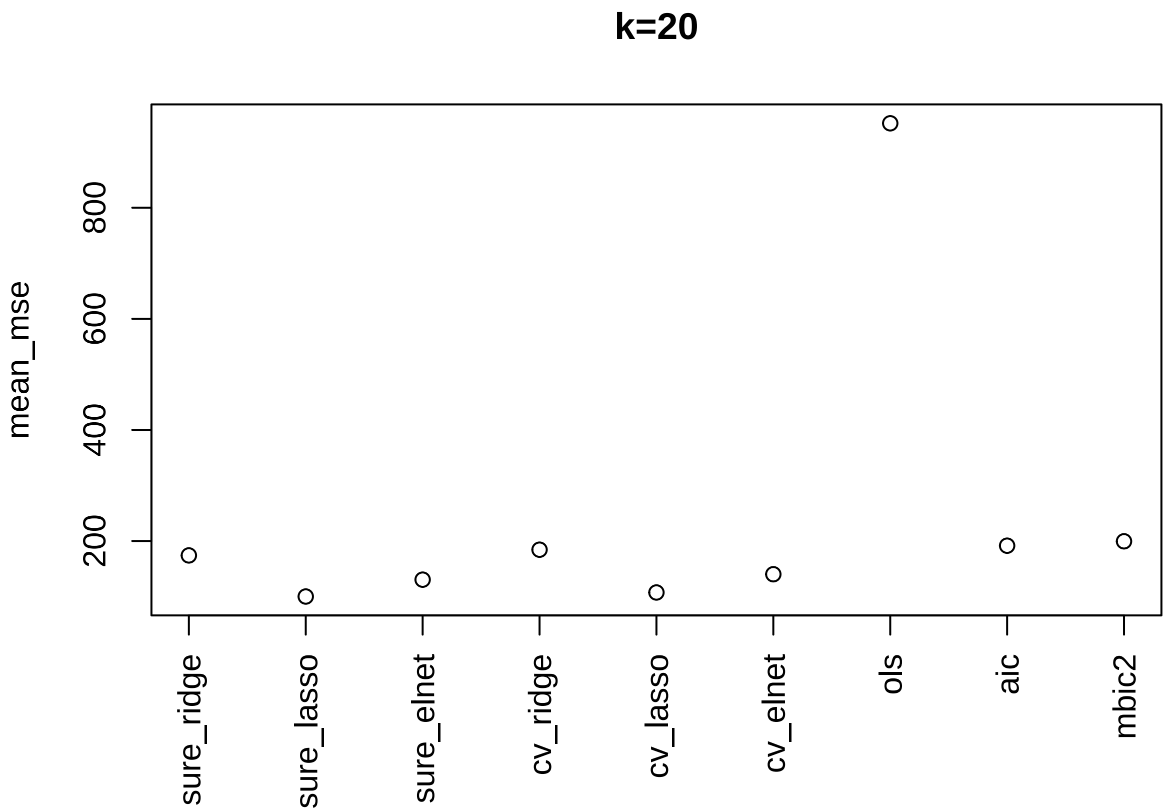
OLS error on beta is very large so I did not included it in beta error plots.

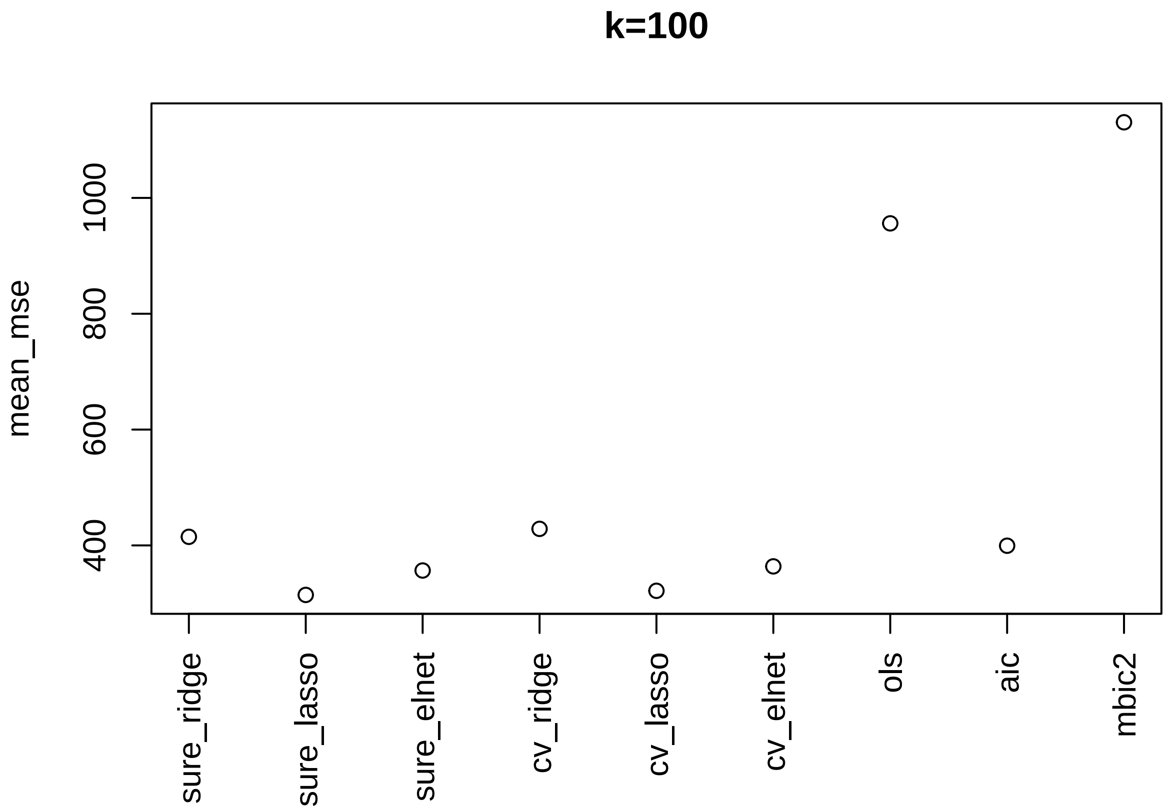
For k=20 we can see that lasso performs the best, medium performance is achieved by elastic net and worse by ridge, aic and mbic2. We can also notice that sure based methods performed slightly better than cv ones.

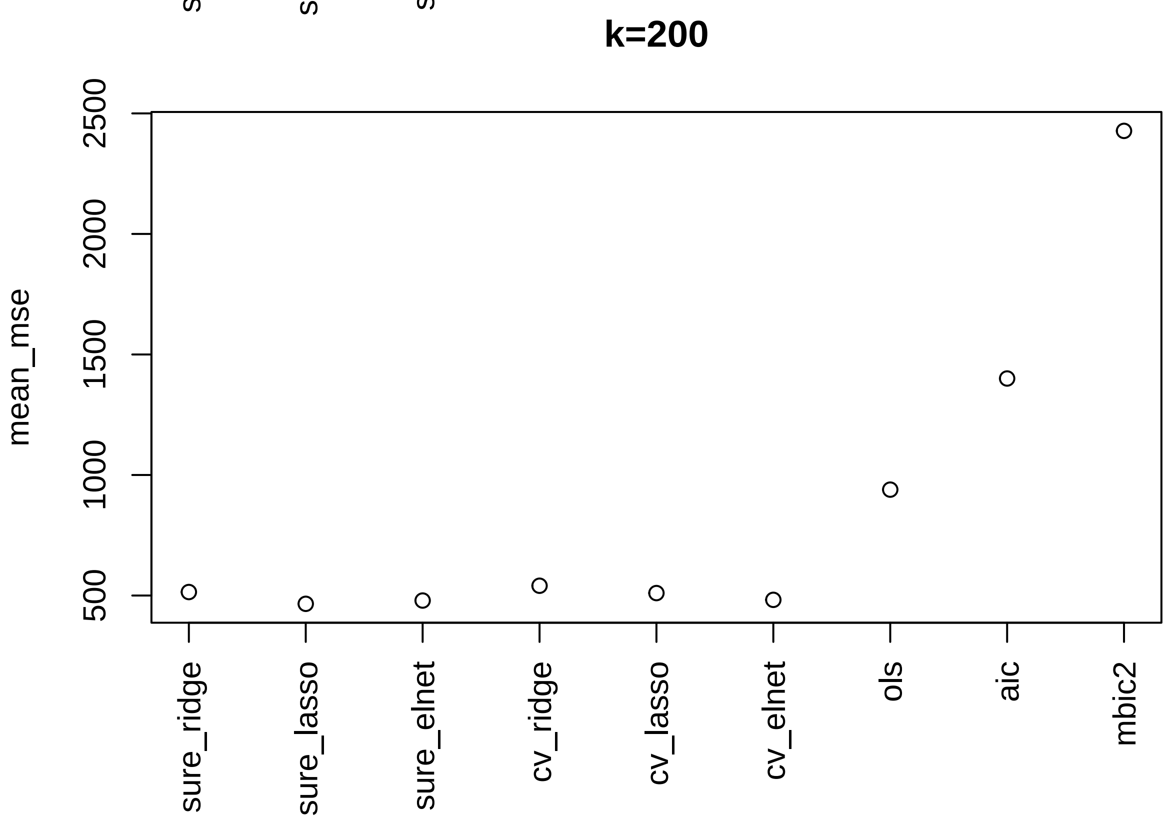
For k=100 we can see similar behavior however aic performed as well as lasso while mbic2 performed much worse.

For k=200 both aic and mbic2 got much bigger beta errors.

```
##     k mean_mse_sure_ridge mean_mse_sure_lasso mean_mse_sure_elnet
## 1  20            174.0405            100.0673            130.4272
## 2 100            414.8197            314.4976            356.6151
## 3 200            514.5409            465.6254            479.1545
##   mean_mse_crossval_ridge mean_mse_crossval_lasso mean_mse_crossval_elnet
## 1                184.3592                107.3219                140.1526
## 2                428.5623                321.6142                363.7711
## 3                540.5680                510.1541                482.0360
##   mean_mse_ols mean_mse_aic mean_mse_mbic2
## 1     951.8873     191.5663       199.4325
## 2     956.0505     399.4564      1130.5832
## 3     939.3034    1400.2854      2427.4107
```





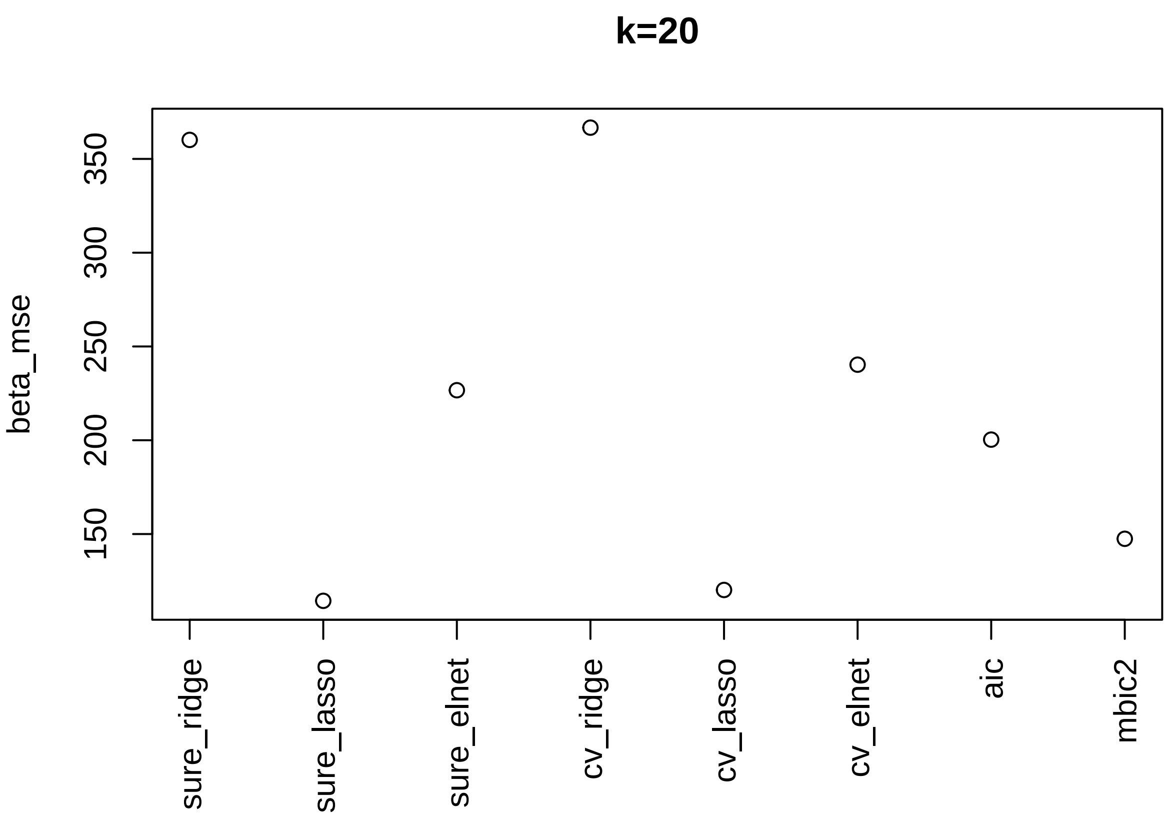


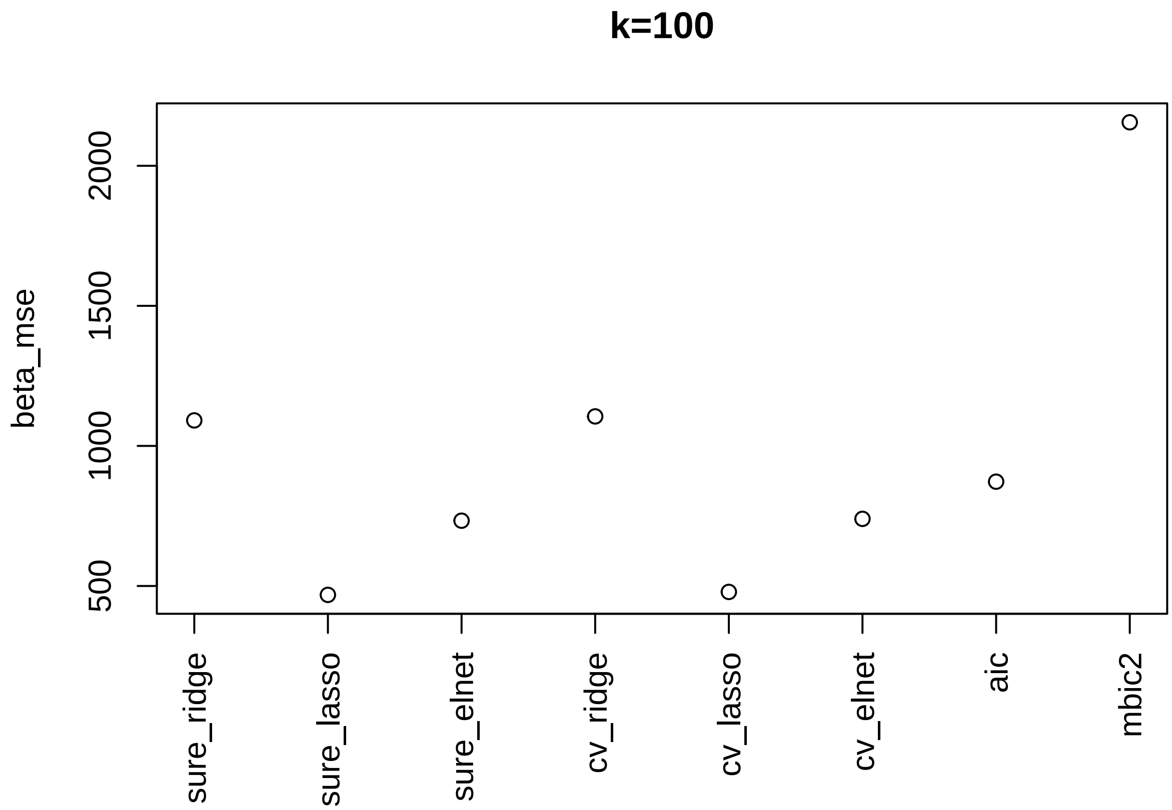
In terms of prediction error for k=20 ols performed much worse than all of the other methods. The relation between other models errors are similar as in the beta error. For k=100 mbic2 error increases and is even above ols error. For k=200 aic error raises above ols but mbic2 has still the highest error. For all k lasso with sure has the smallest error.

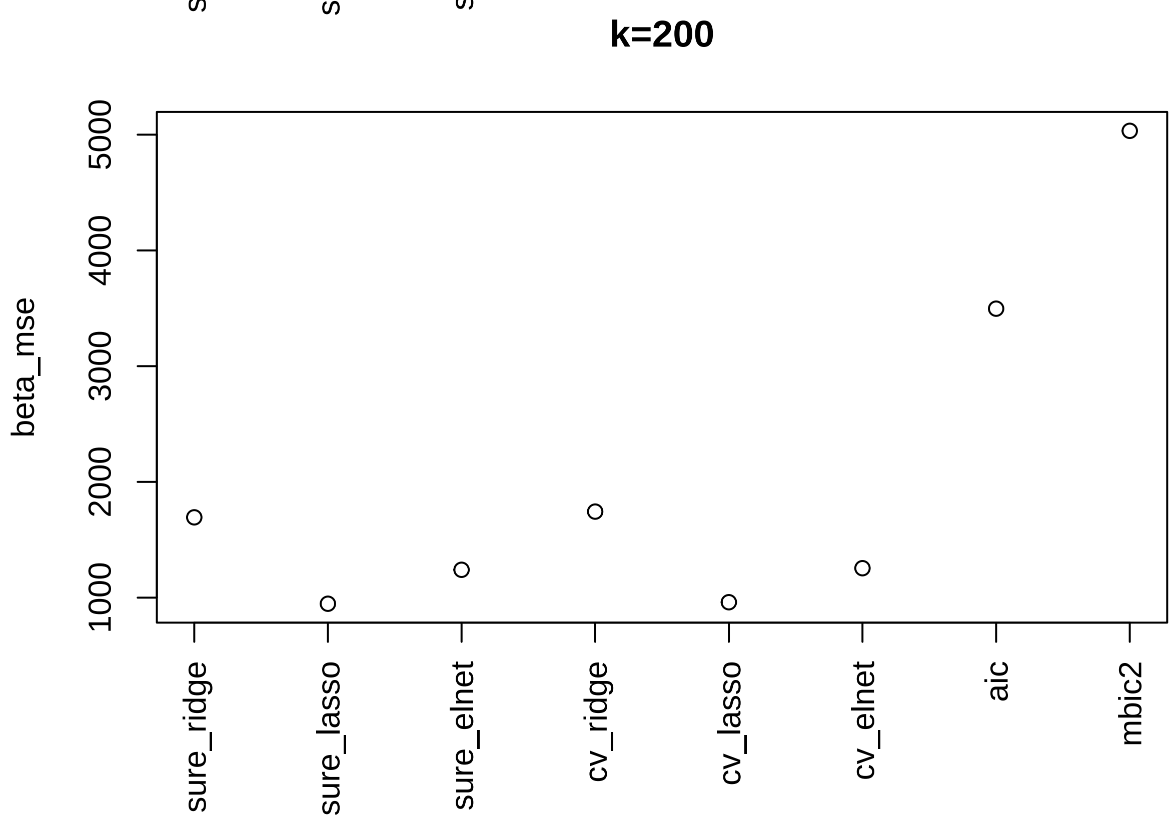
Task 3

In this task we increased nonzero betas from 3.5 to 5.0.

```
##     k beta_mse_sure_ridge beta_mse_sure_lasso beta_mse_sure_elnet
## 1  20            360.1333            114.3931            226.6717
## 2 100           1091.0891            468.2276            733.0912
## 3 200           1693.9760            947.4637           1240.5728
##   beta_mse_crossval_ridge beta_mse_crossval_lasso beta_mse_crossval_elnet
## 1                366.6655                120.1801                240.3007
## 2               1105.2997                478.7989                739.5884
## 3               1743.4023                960.5777               1254.5801
##   beta_mse_ols beta_mse_aic beta_mse_mbic2
## 1     18401.42     200.3473       147.4674
## 2     17873.44     872.3155      2155.2991
## 3     17821.23    3496.6302      5033.1452
```



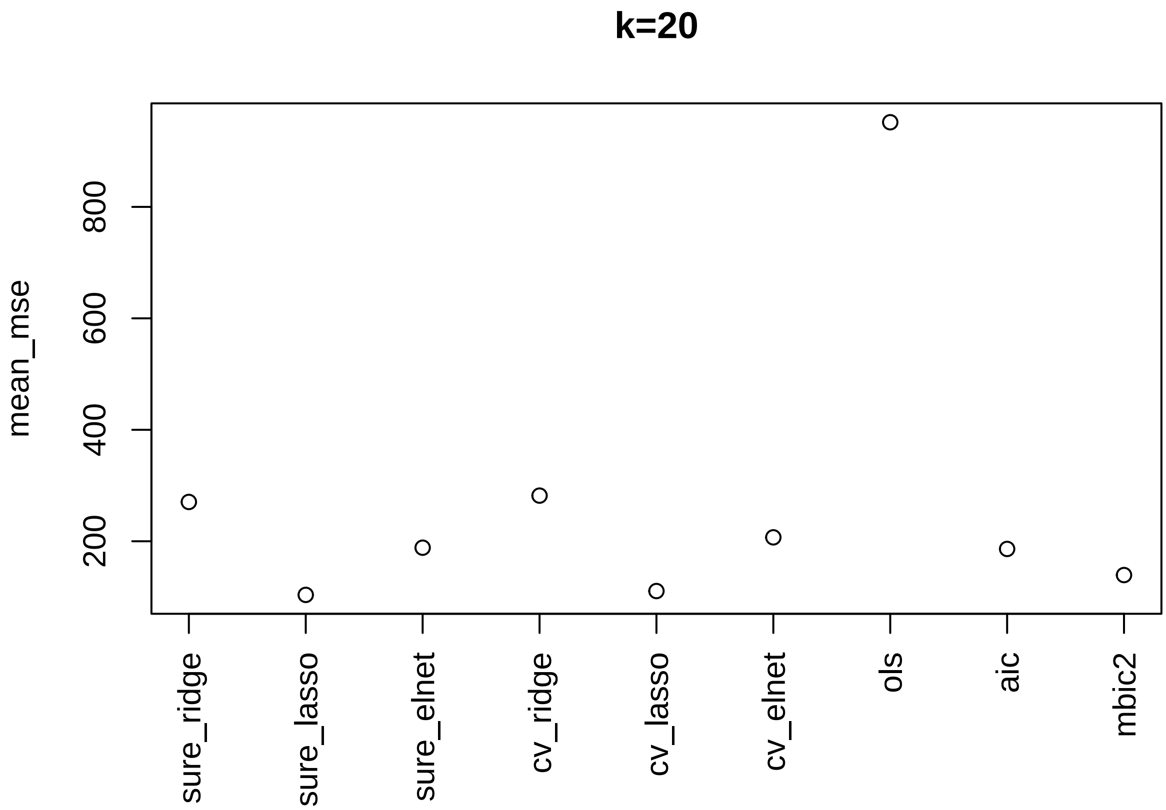


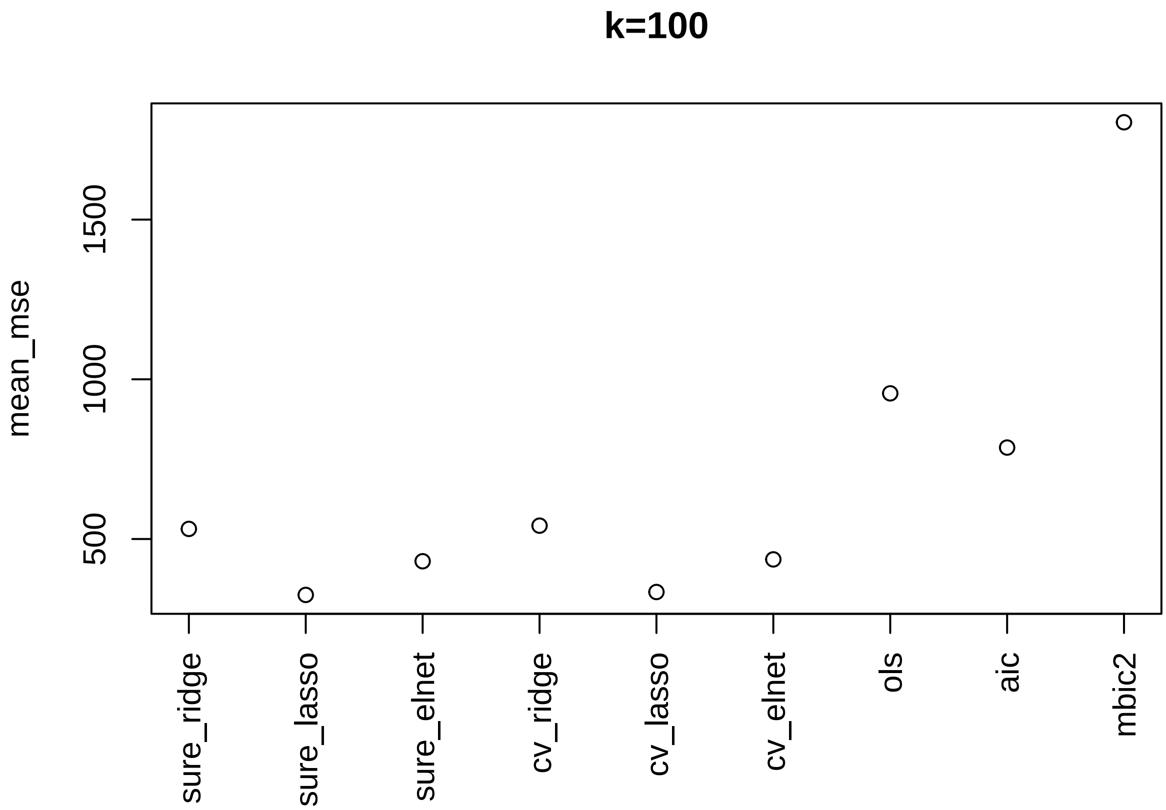


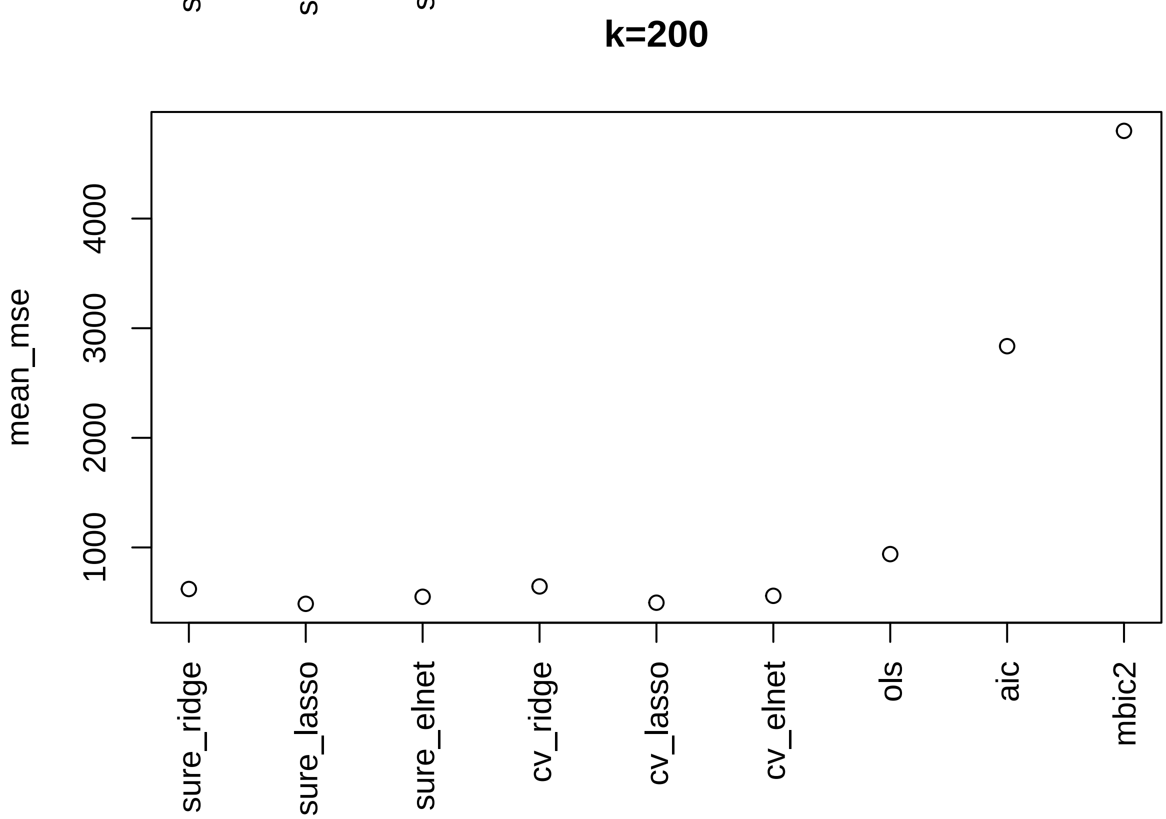
For beta error and k=20,100,200 the relations between lasso, ridge and elastic net errors stayed the same, but for k=20 aic and mbic2 errors decreased compared to other methods. OLS error remained significantly larger.

```
##      k mean_mse_sure_ridge mean_mse_sure_lasso mean_mse_sure_elnet
```

```
## 1  20            270.5786            103.8158            188.4807
## 2 100            531.6348            324.9733            430.3606
## 3 200            620.8363            485.8060            549.9181
##   mean_mse_crossval_ridge mean_mse_crossval_lasso mean_mse_crossval_elnet
## 1                281.9563                110.6898                207.0473
## 2                541.8537                334.0607                436.1528
## 3                644.2265                495.8843                558.7693
##   mean_mse_ols mean_mse_aic mean_mse_mbic2
## 1     951.8873     186.2318       139.3696
## 2     956.0505     786.3203      1804.7024
## 3     939.3034    2836.0270      4799.9305
```





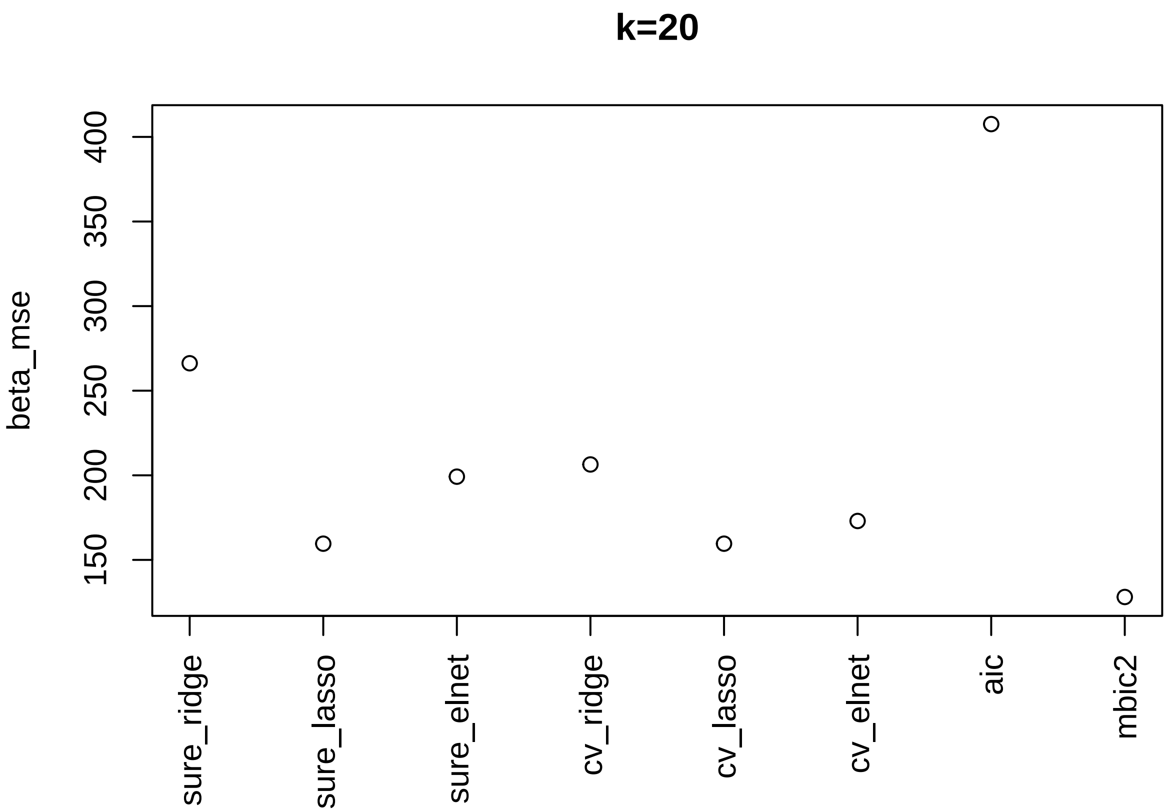


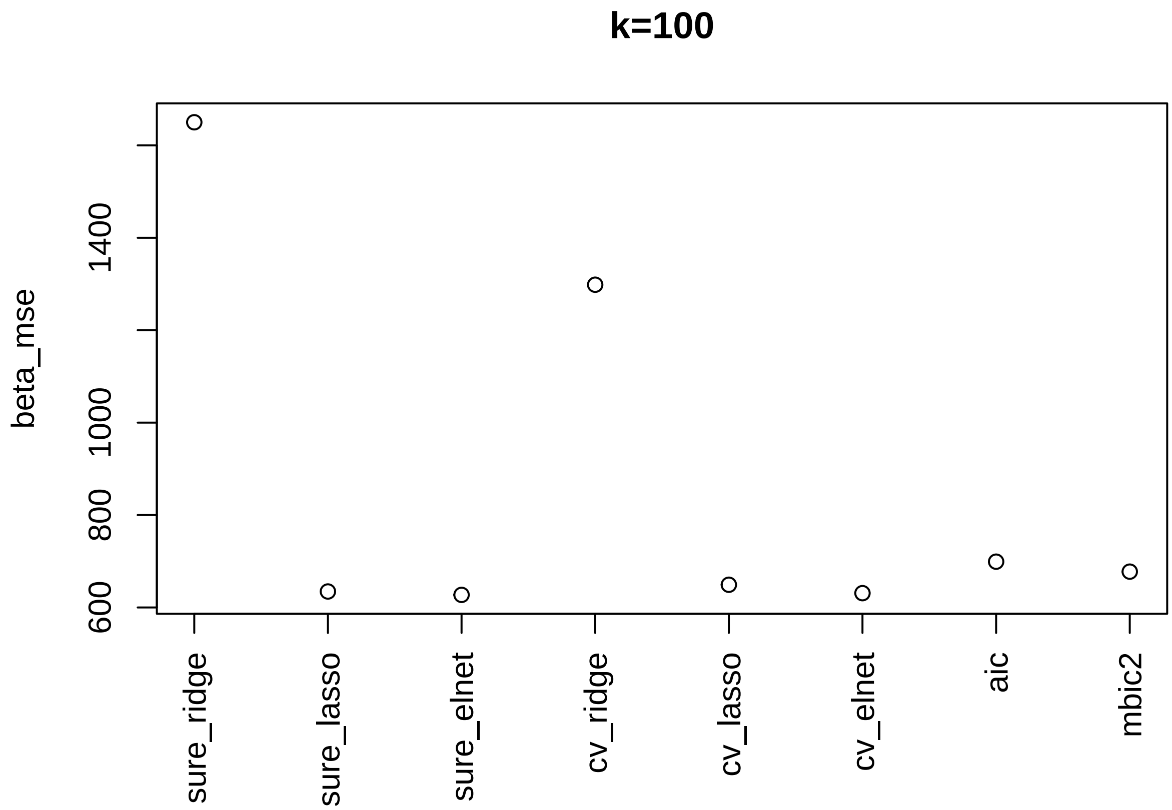
Similarly as for beta error, for prediction error and k=20,100,200 the relations between lasso, ridge and elastic net errors stayed the same, but for k=20 aic and mbic2 errors decreased compared to other methods.

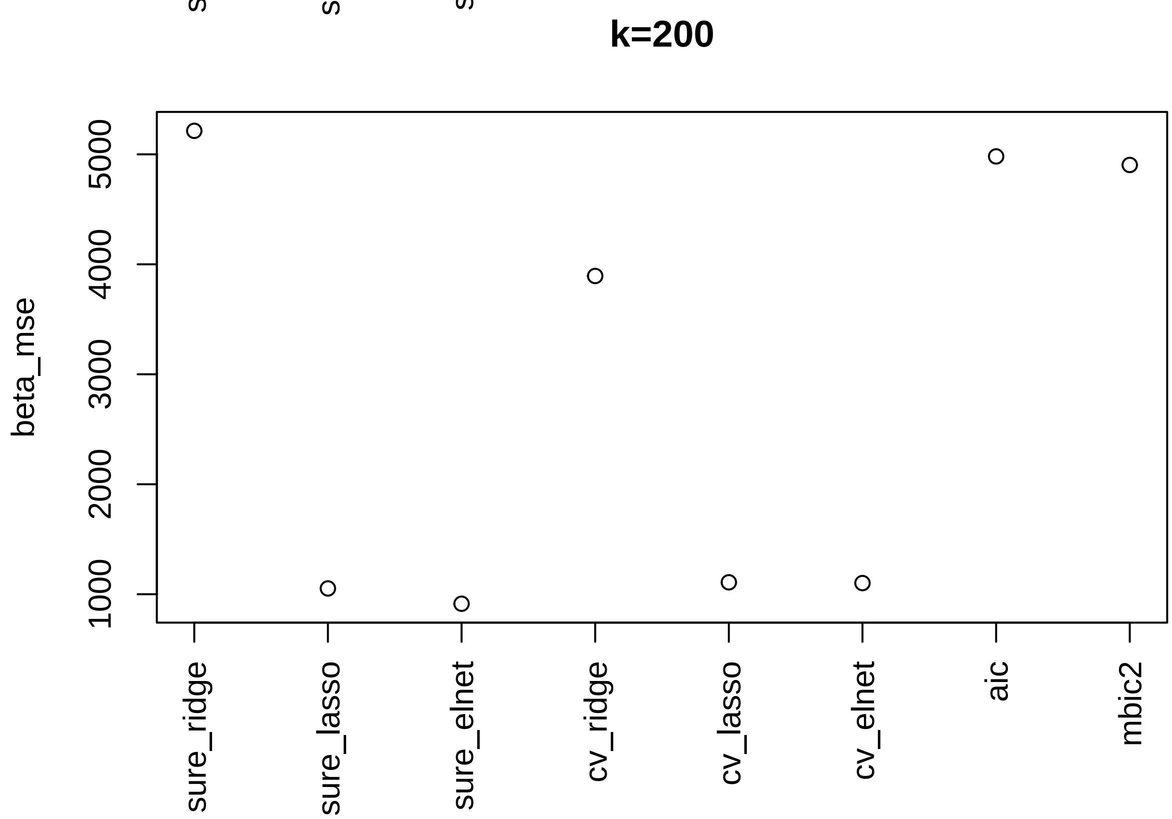
Task 4a

In 4a I considered setting with nonzero betas equal to 3.5. In 4b I considered setting with nonzero betas equal to 5.

```
##     k beta_mse_sure_ridge beta_mse_sure_lasso beta_mse_sure_elnet
## 1  20            266.2262            159.5779            199.1947
## 2 100           1649.9872            634.5406            627.2172
## 3 200           5213.5040           1052.5240            913.4341
##   beta_mse_crossval_ridge beta_mse_crossval_lasso beta_mse_crossval_elnet
## 1                206.4105                159.5779                172.9972
## 2               1298.3959                649.0981                630.9548
## 3               3893.8367               1108.2143               1101.2479
##   beta_mse_ols beta_mse_aic beta_mse_mbic2
## 1     37656.24     407.5788       128.0803
## 2     39828.78     699.0792       677.4920
## 3     37808.33    4980.9906      4903.2222
```

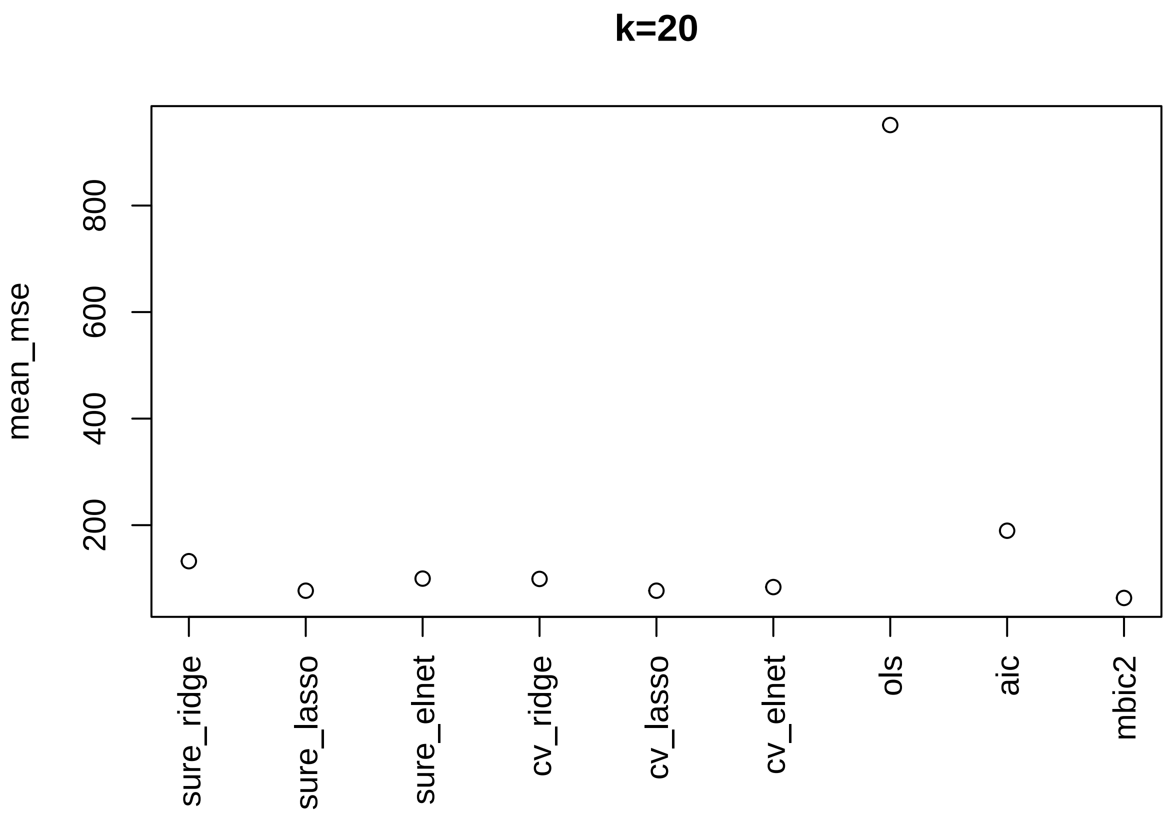


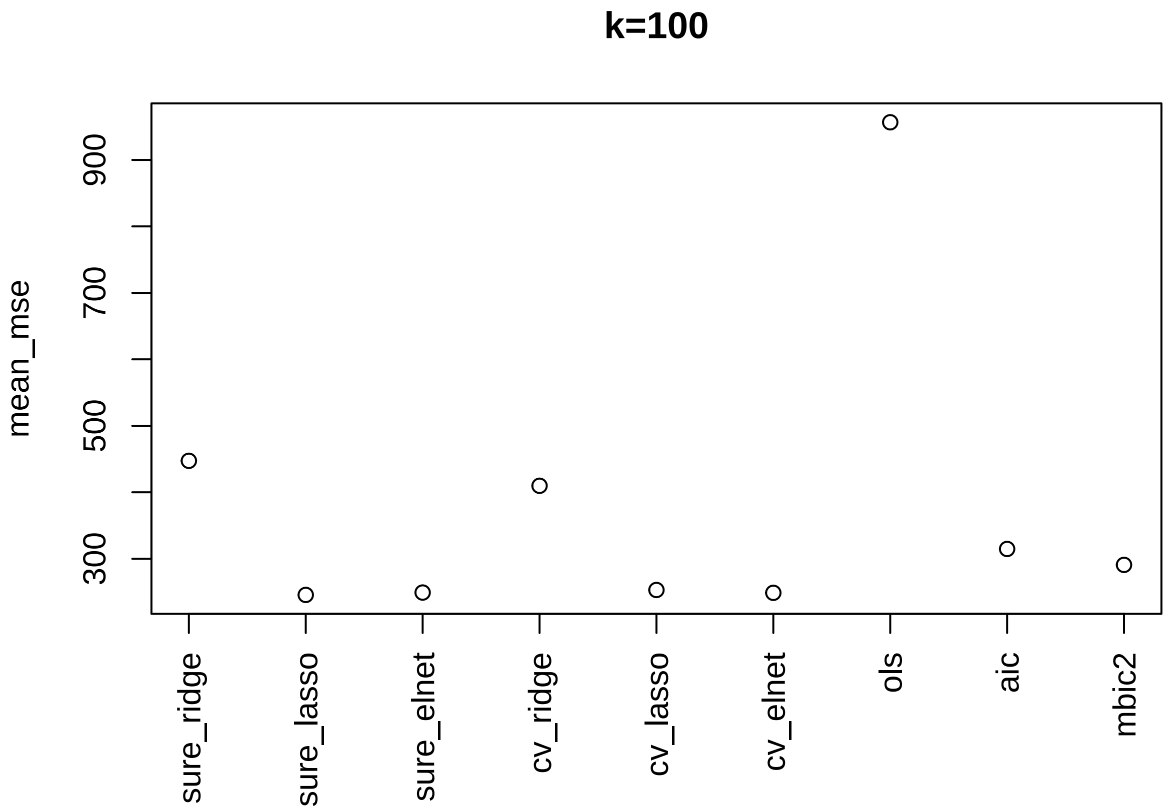


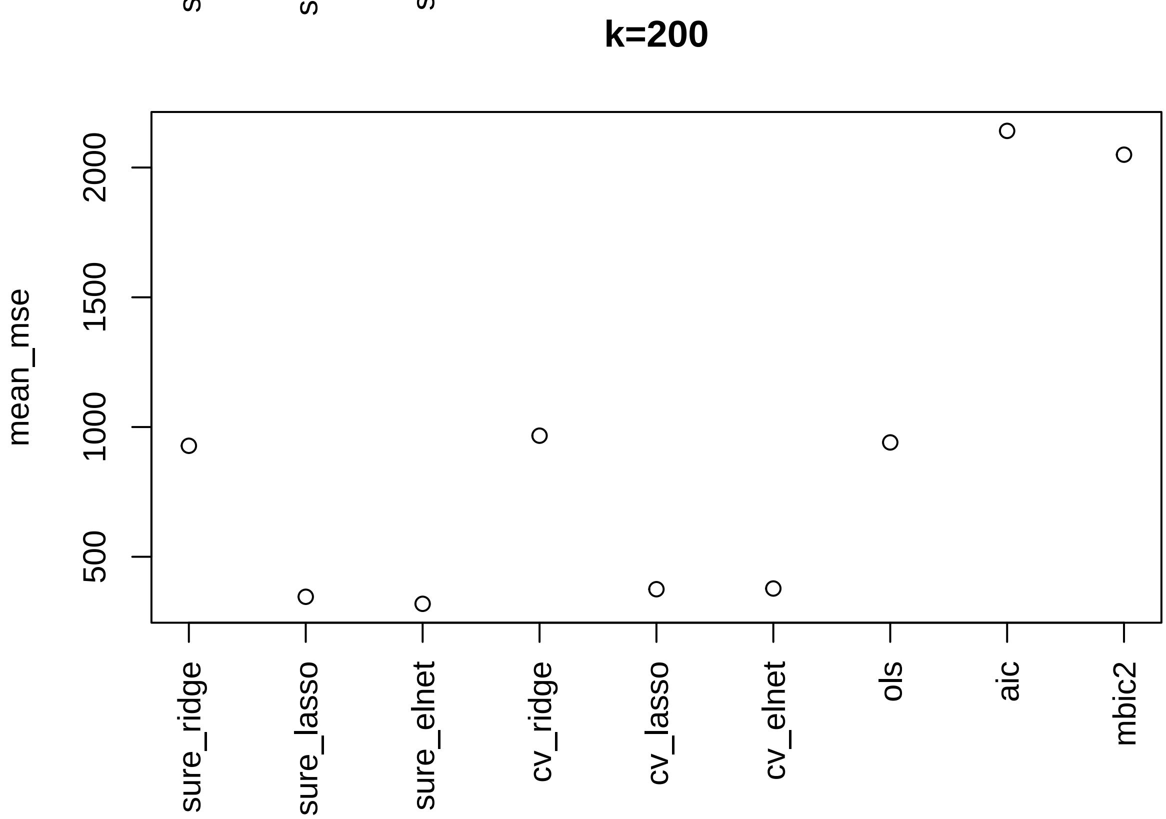


We can observe that beta error for k=20 is smaller for cv variants of methods than for sure based ones. AIC had the worst performance (omitting ols). For k=100 and k=200 cv is again better than sure for ridge, but for lasso and elastic net sure returned smaller errors. Ridge performed much worse than other methods for k=100. For k=200 aic and mbic2 performed similarly bad as ridge.

```
##     k mean_mse_sure_ridge mean_mse_sure_lasso mean_mse_sure_elnet
## 1  20            132.3283            76.90229            99.64248
## 2 100            447.3907           245.62463           249.16886
## 3 200            927.7078           345.87962           318.74017
##   mean_mse_crossval_ridge mean_mse_crossval_lasso mean_mse_crossval_elnet
## 1                98.99426                76.90229                83.70152
## 2               409.79120               253.02649               248.77832
## 3               966.65566               375.00815               377.67297
##   mean_mse_ols mean_mse_aic mean_mse_mbic2
## 1     951.1619     189.5051       63.31506
## 2     956.6549     314.7551      290.81048
## 3     940.6665    2141.2202     2049.55359
```



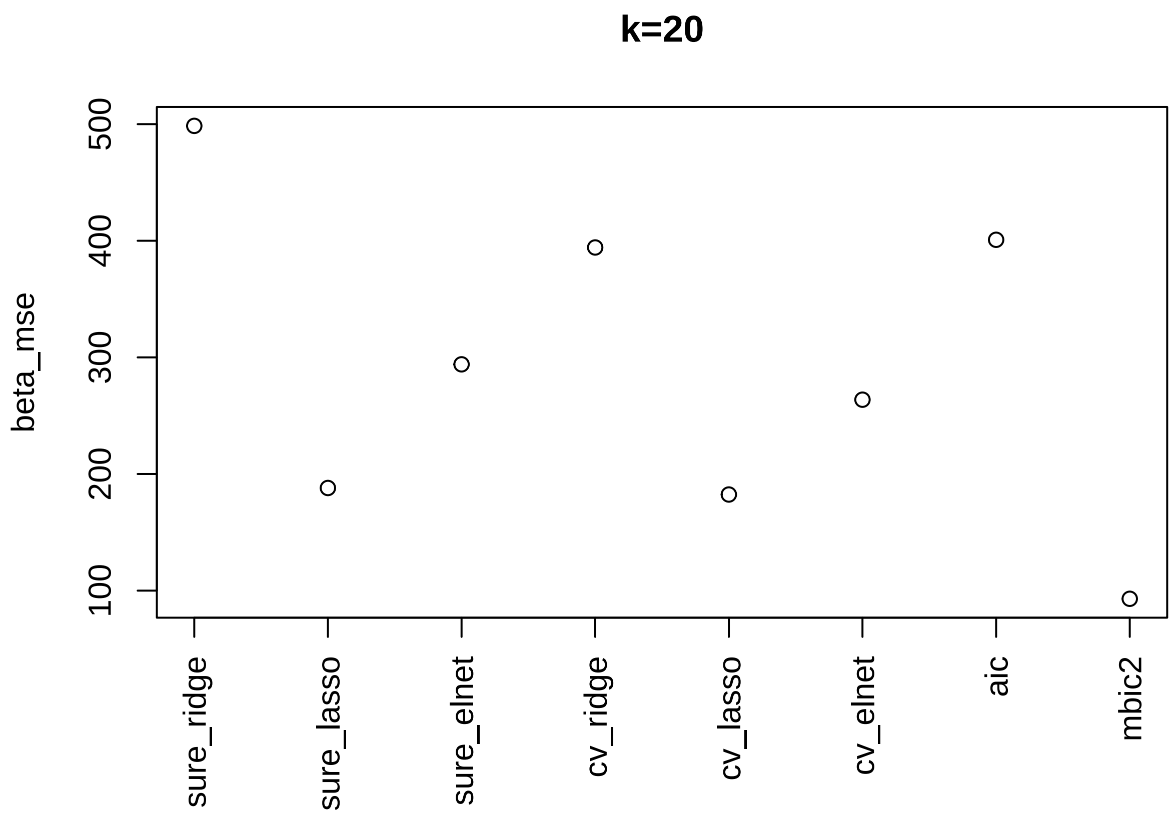


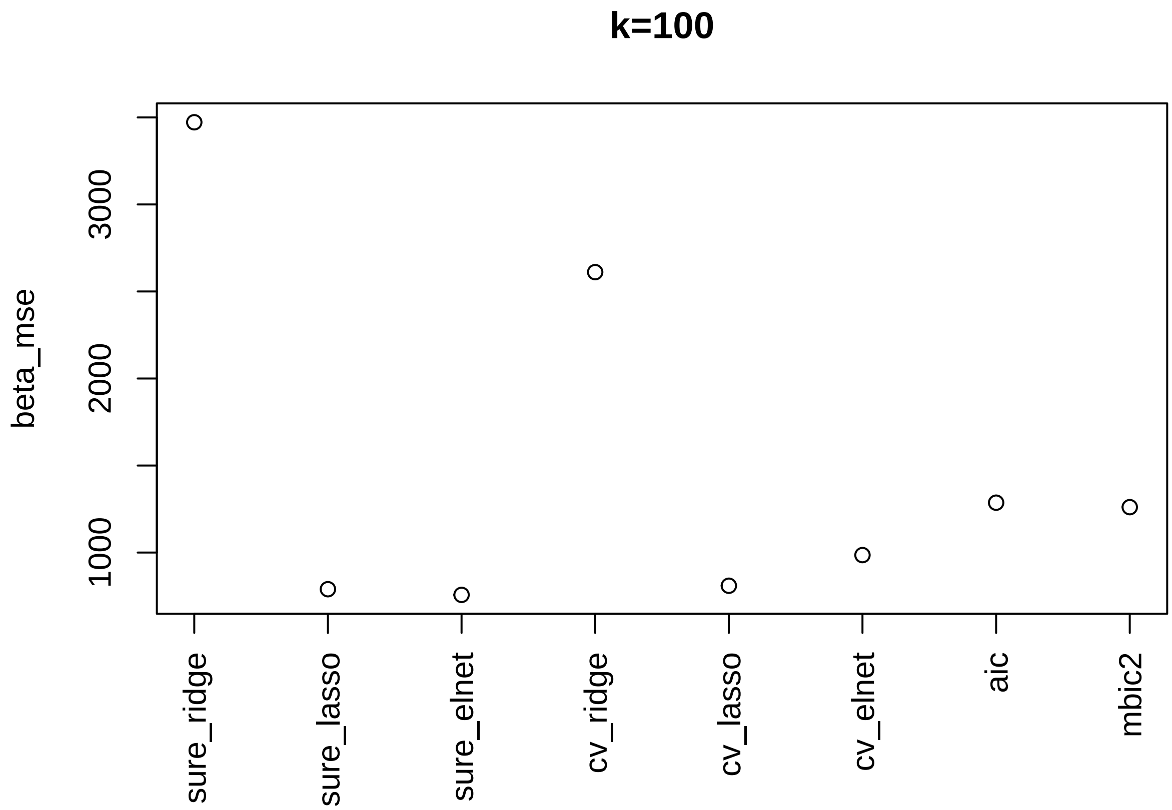


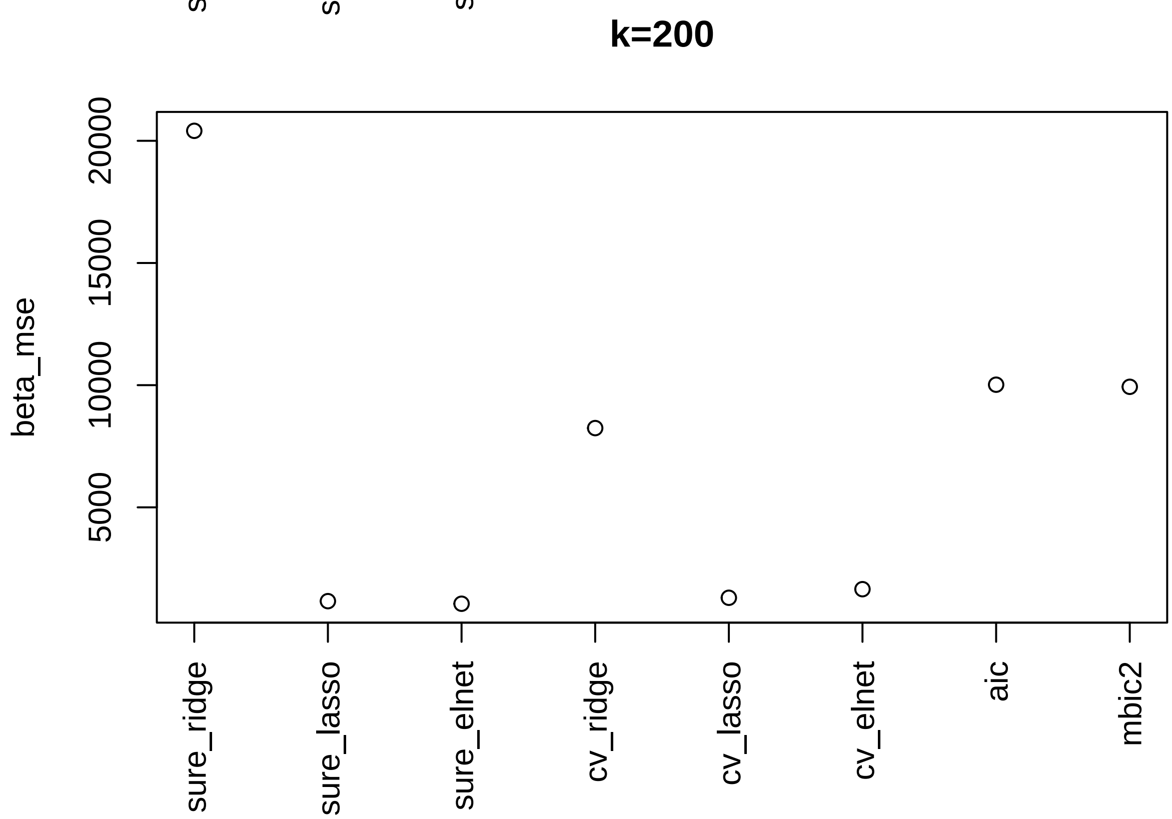
For prediction error we can see similar behaviors as in previous tasks, however sure tends to perform better for bigger k and worse for smaller. Also ridge tends to perform much worse than in previous tasks.

Task 4b

```
##     k beta_mse_sure_ridge beta_mse_sure_lasso beta_mse_sure_elnet
## 1  20             498.505            187.9974            294.0292
## 2 100            3472.213            789.2700            756.7665
## 3 200           20408.275           1160.9083           1055.0680
##   beta_mse_crossval_ridge beta_mse_crossval_lasso beta_mse_crossval_elnet
## 1                394.2478                182.4038                263.7126
## 2               2611.0872                809.2388                985.3704
## 3               8243.6581               1298.6807               1650.4757
##   beta_mse_ols beta_mse_aic beta_mse_mbic2
## 1     37656.24     400.8234        93.0004
## 2     39828.78    1286.5242      1260.7907
## 3     37808.33   10018.6552      9933.1365
```

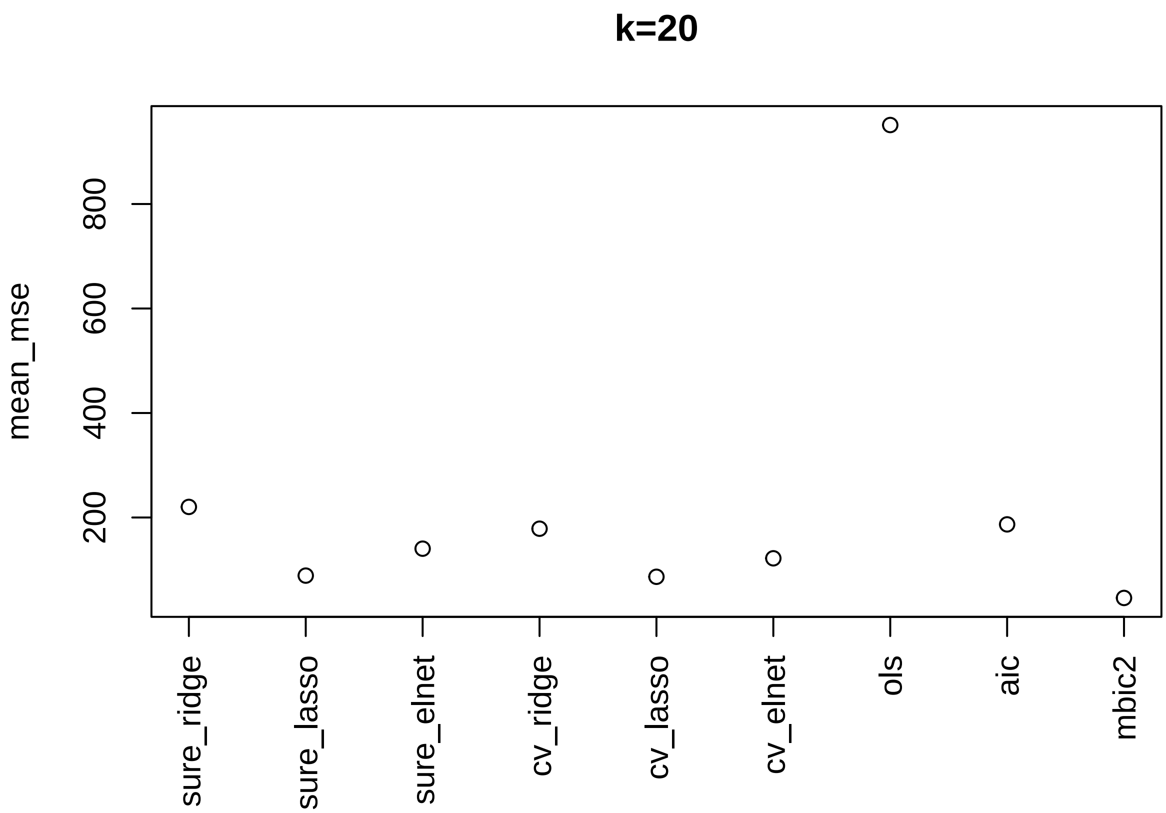


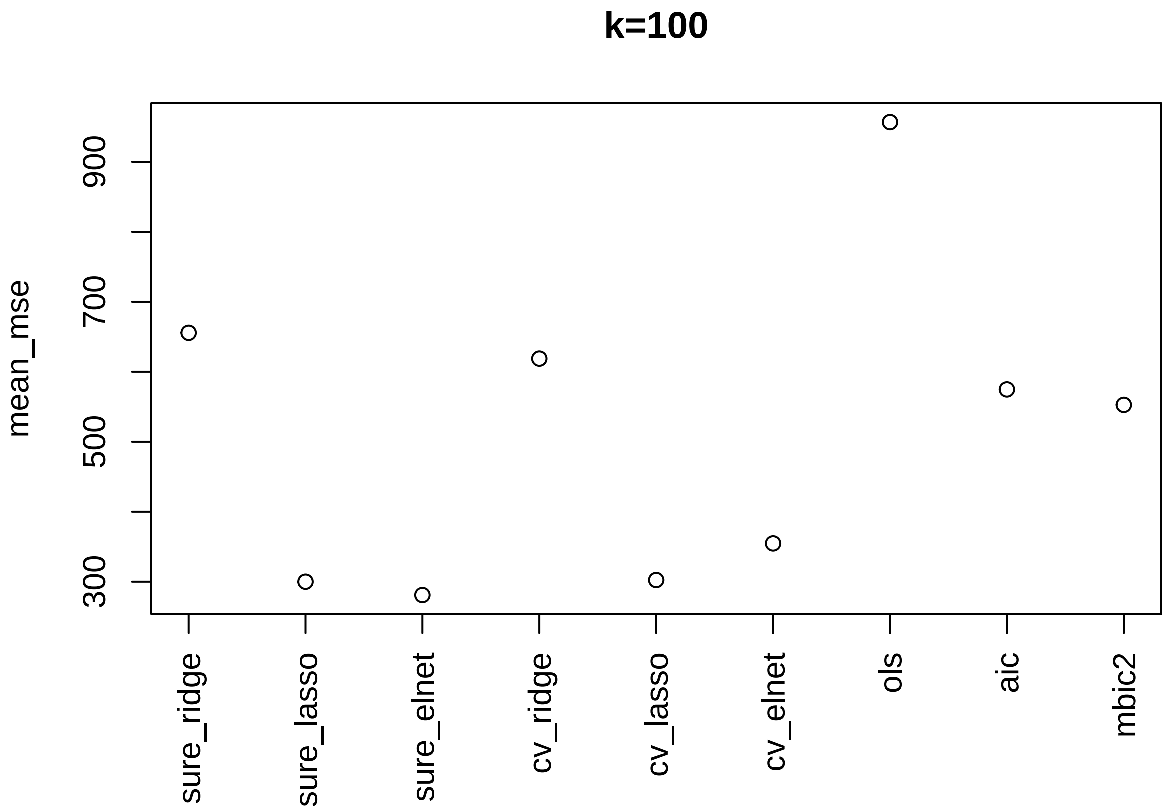


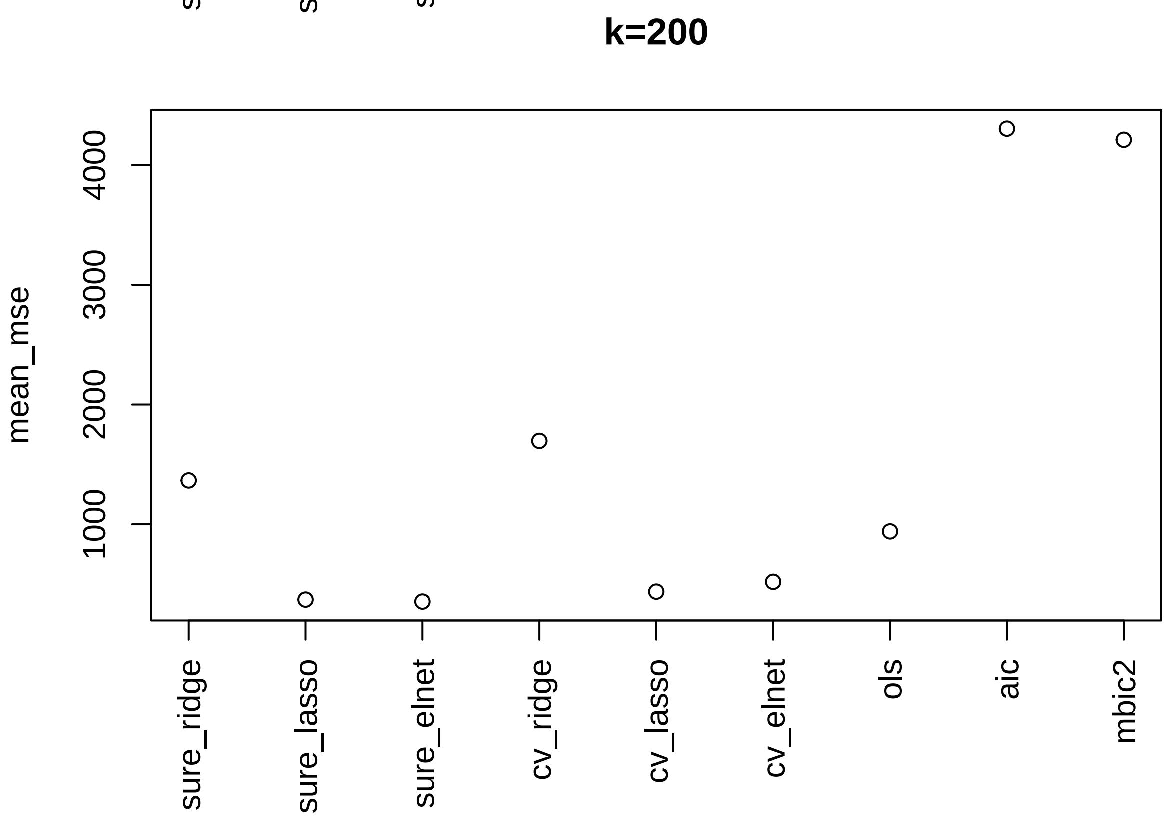


We can observe that beta error for k=20 is smaller for cv variants of methods than for sure based ones. For k=100 and k=200 cv is again better than sure for ridge, but for lasso and elastic net sure returned smaller errors than cv. Ridge performed much worse than other methods for k=100. For k=200 aic and mbic2 performed similarly bad as ridge.

```
##     k mean_mse_sure_ridge mean_mse_sure_lasso mean_mse_sure_elnet
## 1  20            220.2562            88.75759            140.3390
## 2 100            655.5940           299.98533            280.9368
## 3 200           1366.1738           371.05006            354.6457
##   mean_mse_crossval_ridge mean_mse_crossval_lasso mean_mse_crossval_elnet
## 1                178.5724                 86.5319                121.9050
## 2                618.7424                302.3945                354.6784
## 3               1696.4406                437.6994                519.8118
##   mean_mse_ols mean_mse_aic mean_mse_mbic2
## 1     951.1619     186.8461       46.01947
## 2     956.6549     574.7124      552.64829
## 3     940.6665    4303.4438     4210.76447
```

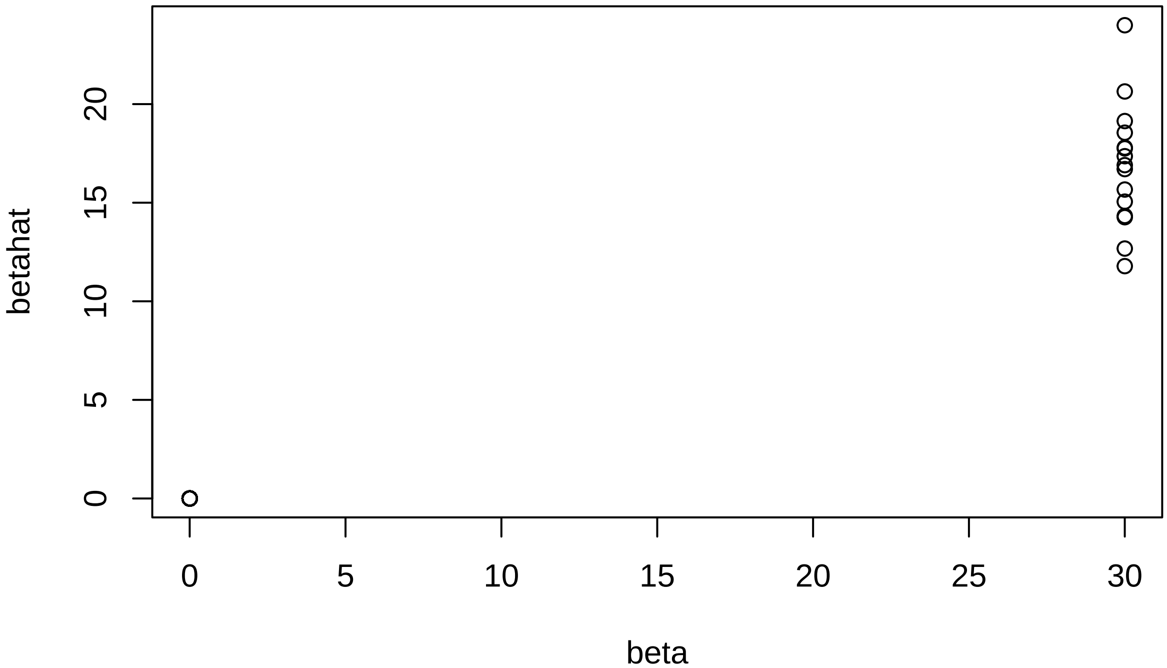




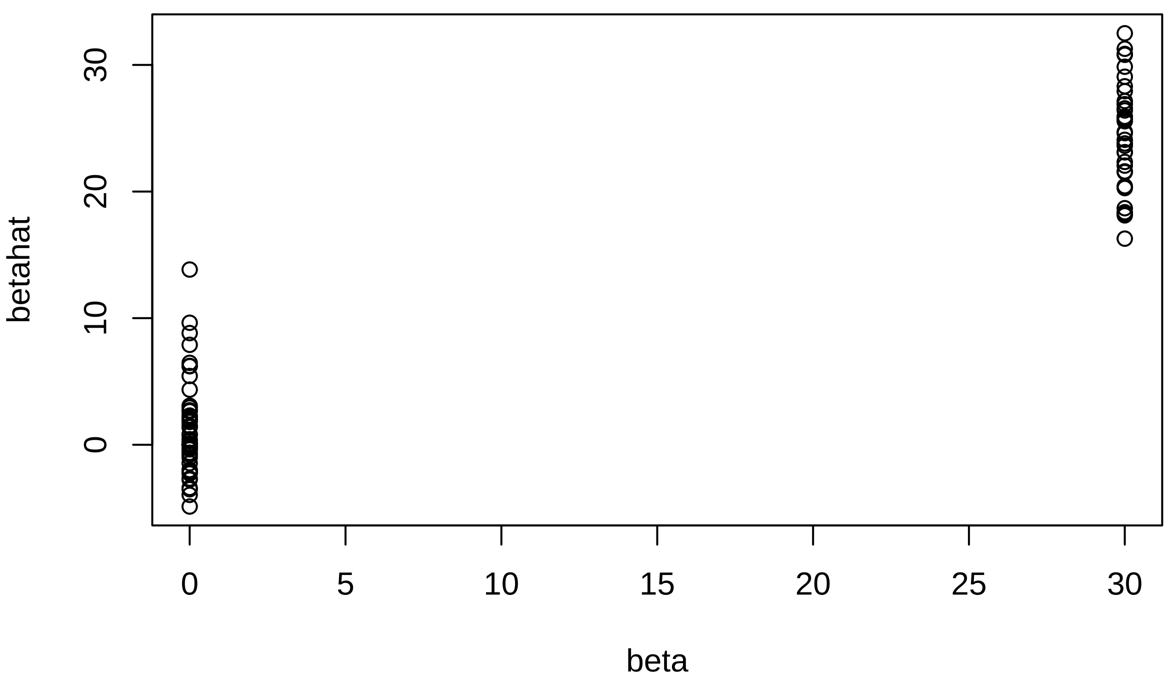


For prediction error we can see similar behaviors as in previous tasks, however sure tends to perform better for bigger k and worse for smaller. Also ridge tends to perform much worse than in tasks 2 and 3.

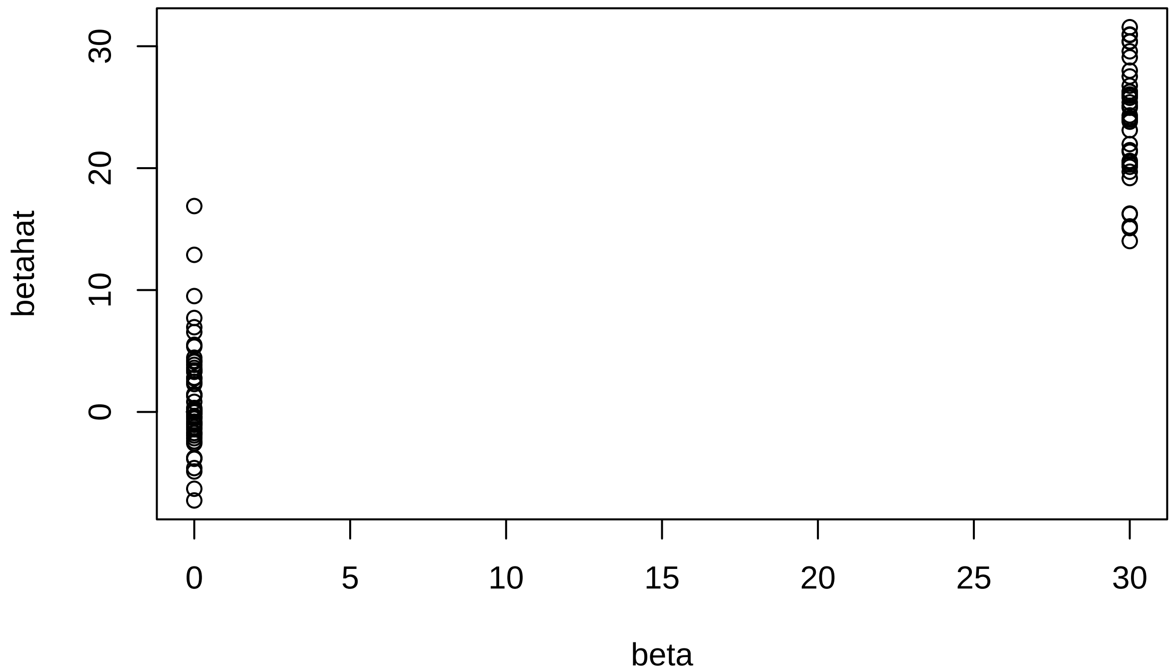
Task 5



At first try after I found k_{IR} I was not able to find λ such that LASSO can recover the sign of β so I increased the magnitude of the nonzero elements of β to 30. After the change the maximum k for which the LASSO irrepresentability condition is satisfied was equal to 15 and corresponding minimal value of lambda was 248.2006544 (1.2410033 in glmnet).



Similarly to irrepresentability I had to increase the magnitude of the nonzero elements of β to 30. After the change the maximum k for which the LASSO irrepresentability condition is satisfied was equal to 36 and corresponding minimal value of lambda was 11.375192 (0.056876 in glmnet).



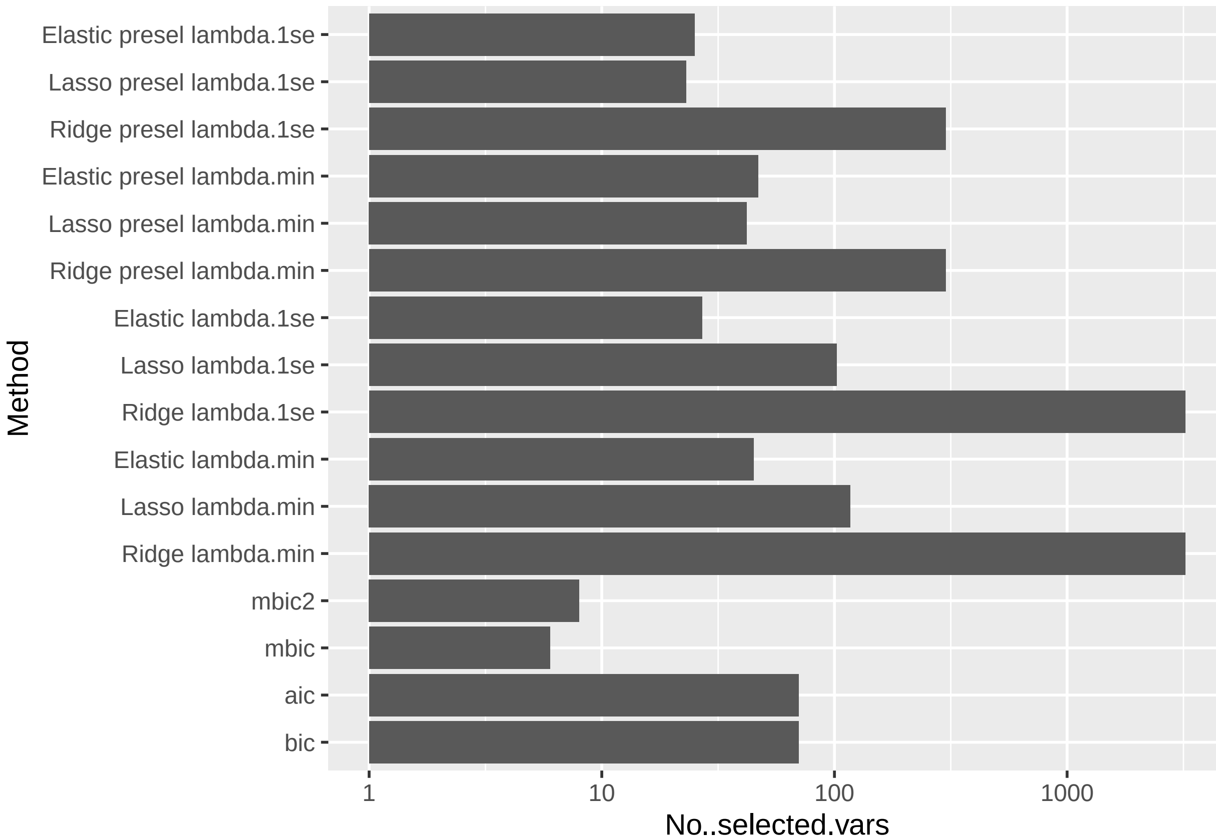
I increased k_{ID} by one and tried to find λ which allows for separating zero and nonzero elements of β but as expected I was not able to find such λ. The closest situation I found is in the plot above.

Task 6

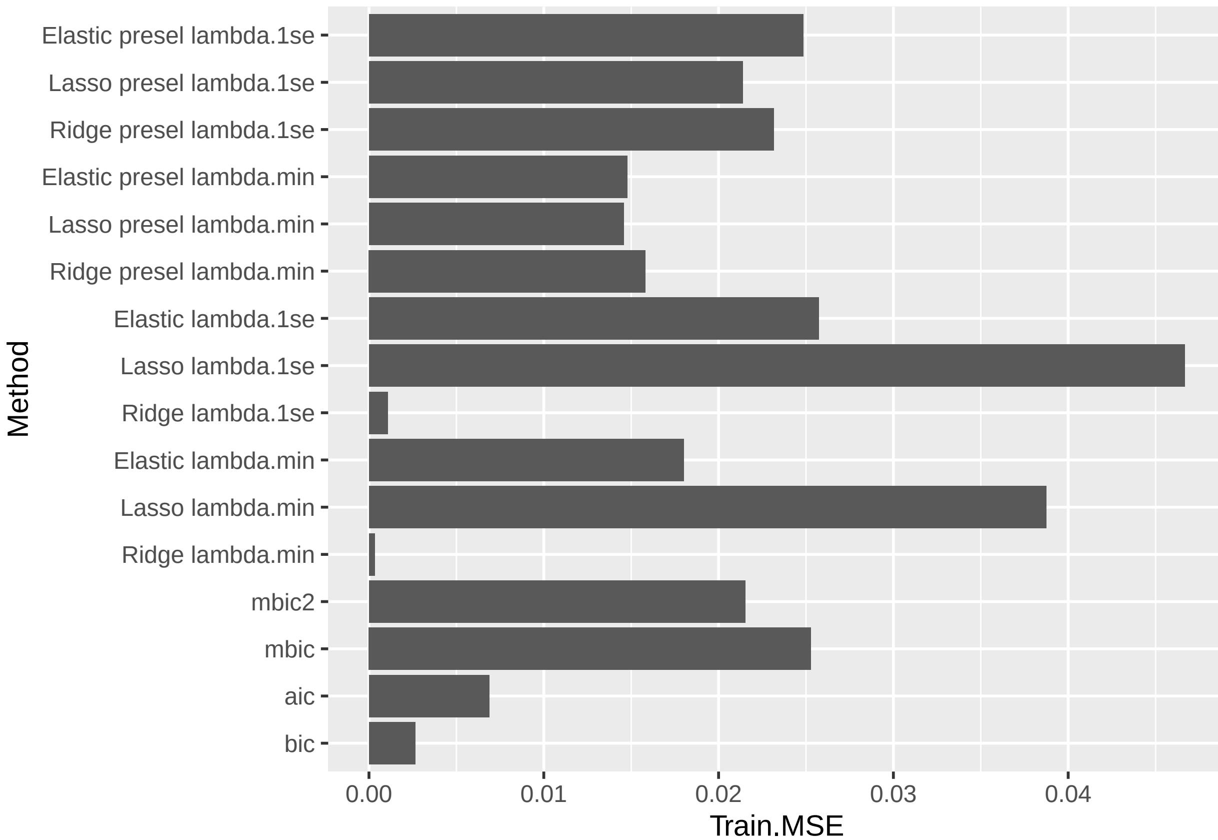
In this task I decided to explore not only models with min.lambda which minimizes mean cross-validated error, but also models with lambda.1se which is the value of lambda that gives the most regularized model such that the cross-validated error is within one standard error of the minimum.

```
##                        Method No. selected vars            Train MSE
## 1                         bic                70  0.00264070333508861
## 2                         aic                70  0.00689174579961226
## 3                        mbic                 6   0.0252871128110062
## 4                       mbic2                 8   0.0215324828848707
## 5            Ridge lambda.min              3220 0.000341591208090349
## 6            Lasso lambda.min               117   0.0387382200512681
## 7          Elastic lambda.min                45   0.0180177241702233
## 8            Ridge lambda.1se              3220   0.0010853003134977
## 9            Lasso lambda.1se               102   0.0466702744338215
## 10         Elastic lambda.1se                27   0.0257408907108202
## 11   Ridge presel lambda.min               301   0.0158179909500775
## 12   Lasso presel lambda.min                42   0.0145785858487278
## 13 Elastic presel lambda.min                47   0.0147656299591155
## 14   Ridge presel lambda.1se               301    0.023167004496337
## 15   Lasso presel lambda.1se                23   0.0213880184540559
## 16 Elastic presel lambda.1se                25   0.0248459202246319
##              Test MSE
## 1  0.0545643839296785
## 2  0.0670360578451789
## 3  0.0276573261967991
## 4   0.026189605266428
## 5   0.122418333911474
```

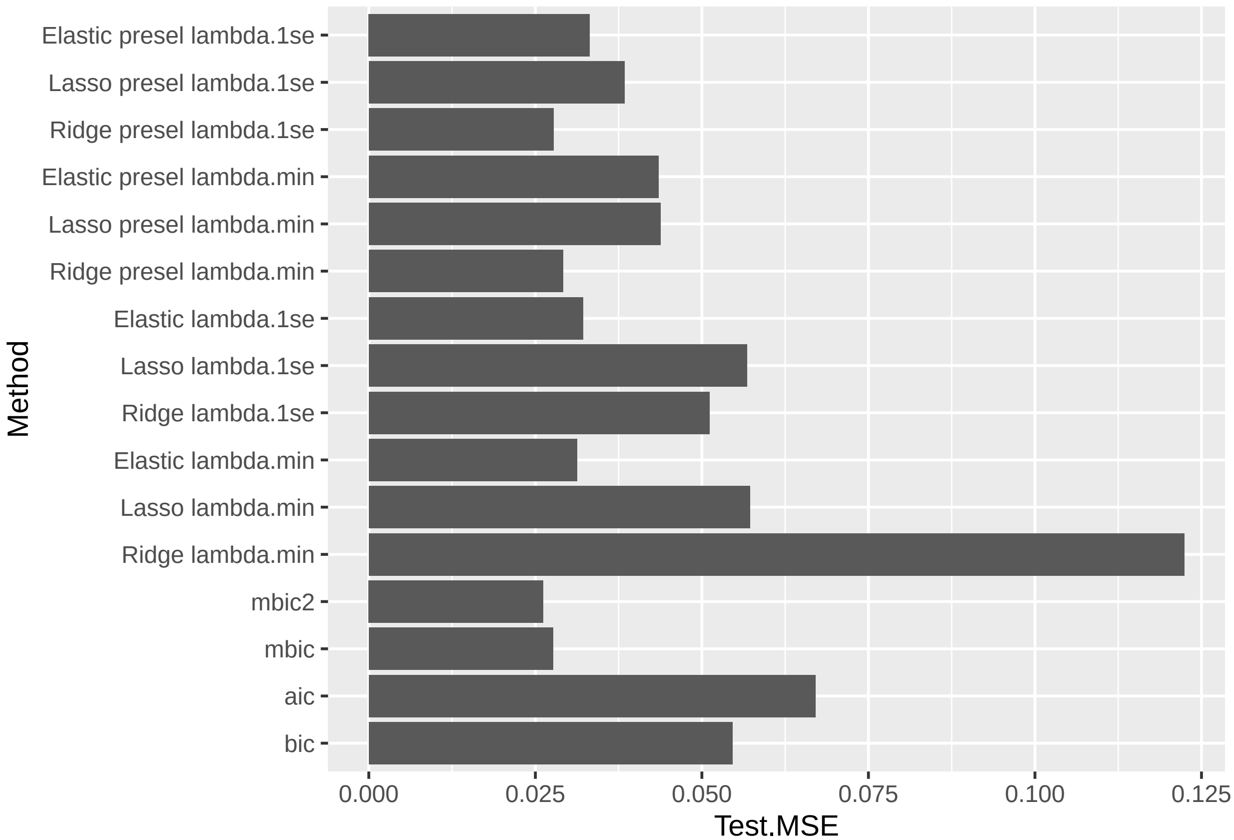
```
## 6  0.0572150195756781
## 7  0.0312773315843396
## 8   0.051165595107883
## 9  0.0567599471974304
## 10 0.0321810624896181
## 11 0.0291603744890595
## 12 0.0438088047541888
## 13 0.0434693242855471
## 14 0.0276974080252445
## 15 0.0383700967519027
## 16 0.0331666844897697
```



Ridge is the only considered method that does not perform variable selection. Elastic selects less variables then lasso when given all variables, but lasso selects a bit less after preselection. Models with lambda.1se for lasso and elastic selects a bit less variables than models with lambda.min (as lambda.1se > lambda.min). AIC and BIC selects more than elastic net and lasso after preselection. MBIC and MBIC2 selects the smallest number of variables.



Lambda.min ridge regression on all variables achieves the smallest error on training set. AIC and BIC are only a bit worse. Then models with preselected variables and lambda.min come having very similar performance. Also preselected lambda.1se models have similar but bigger errors (using lambda.min results in smaller training error than lambda.1se). Lasso has big training error when fitted on all variables.



Mbic and mbic2 have the smallest test error. Models that had the smallest training error (ridge without preselection, aic and bic) now appear to have much higher test error, so we may deduce that they overfitted. We can see that preselection improved test error for lasso and ridge, but decreased for the elastic. It also appears that lambda.1se gives smaller error that lambda.min, so a bit stronger regularization is beneficial for this dataset (glmnet actually uses lambda.1se as default).