Final project

By Ming han, Lei Kang, and Yanyu Long

In this project, we used best subset, forward stepwise, backward stepwise and lasso to find the best yet simple model. All the methods perform variable selection, therefore can help to identify the most important genes in predicting Mapk1. For the first three method, we used Cp, BIC and cross-validation to find the best models; for lasso, we used cross-validation to determine the best lambda and then the best model. Then, all the methods, together with random forest (which does not perform variable selection but we included it to compare with other methods), are compared using cross-validation. The result shows that the lasso and random forest produce the lowest test MSE. Since lasso can select a small number of variables, we used this method to produce our final model, which is xxx .

At last, we checked whether the selected model has collinearity in order to see whether the model can be further simplified. I~~t is found that Rik and Pik3r3 are highly correlated, so either one of these variables can be dropped. However, further examination using cross-validation show that the performance of the (\*) model is better than the models with either Rik and Pik3r3 being dropped.~~ We concluded that (\*) is the simple yet best model. we also checked the linear assumptions of the model.

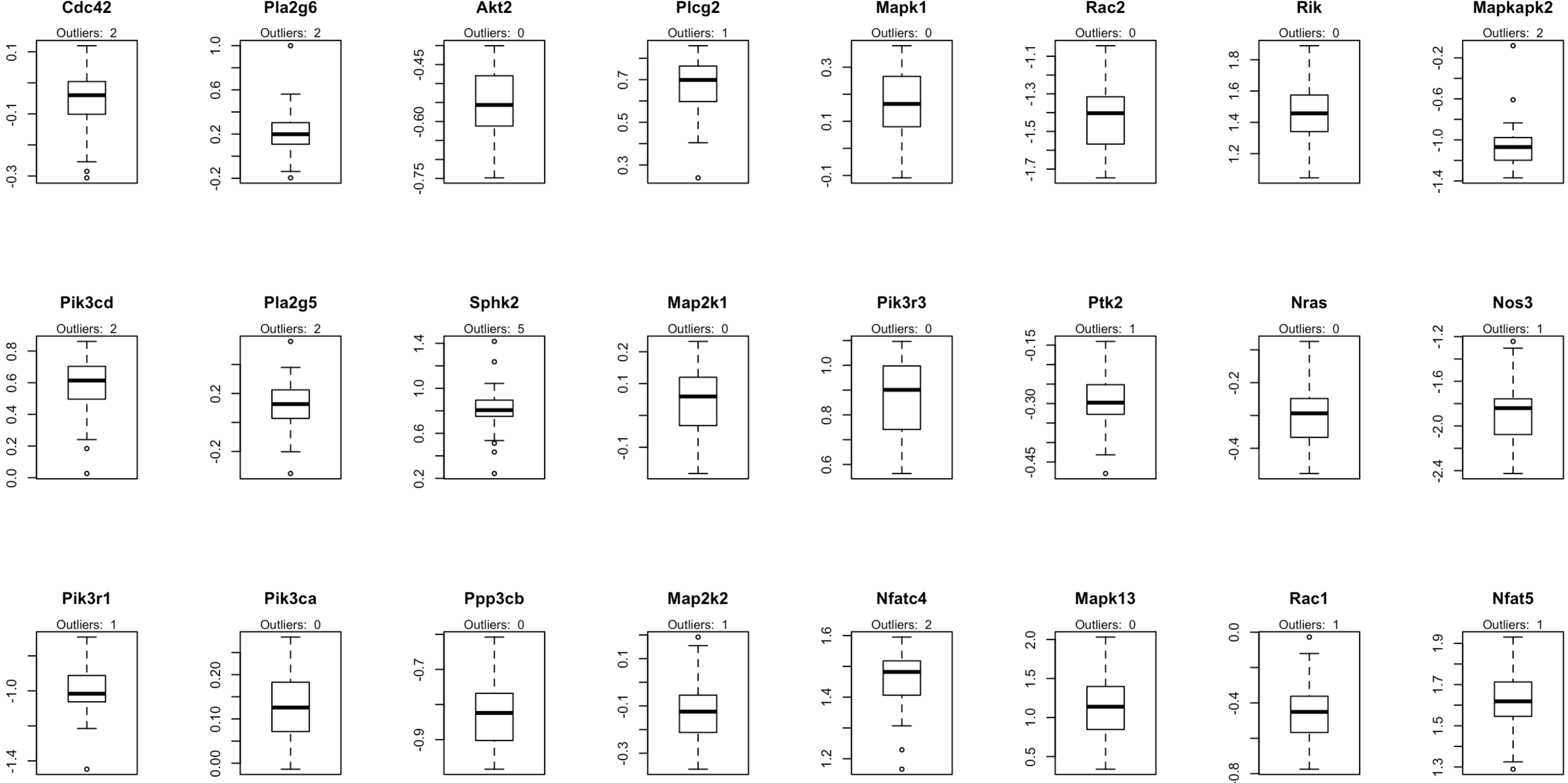
1. Summary of the data

The variables in this dataset:

"Cdc42", "Pla2g6", "Akt2", "Plcg2", "Mapk1", "Rac2", "Rik", "Mapkapk2", "Pik3cd", "Pla2g5", "Sphk2", "Map2k1", "Pik3r3", "Ptk2", "Nras", "Nos3", "Pik3r1", "Pik3ca", "Ppp3cb", "Map2k2", "Nfatc4", "Mapk13", "Rac1", "Nfat5"

For the sake of simplicity, the variables are assigned number #1-#24.

Check the distribution of all the data:



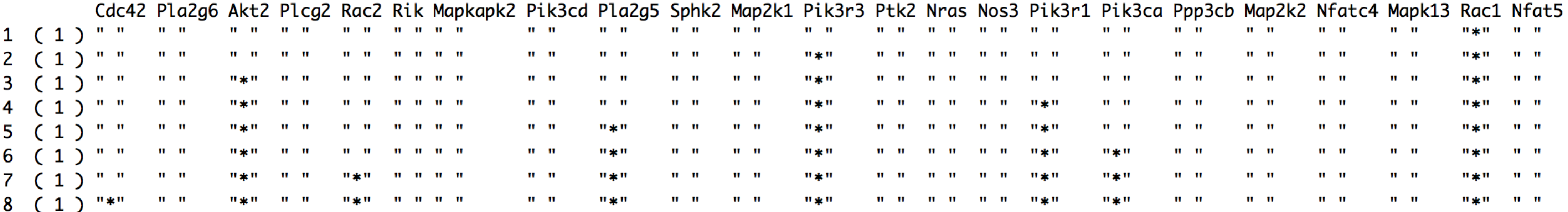
*Figure 1. boxplots of all the variables*

There are some outliers in some variables (1.5 IQR). In the later analysis we did not exclude them, because the sample size is small (n = 40); if all the outliers are excluded, the sample pool will be reduced to about 25.

1. Best subset, forward and backward models

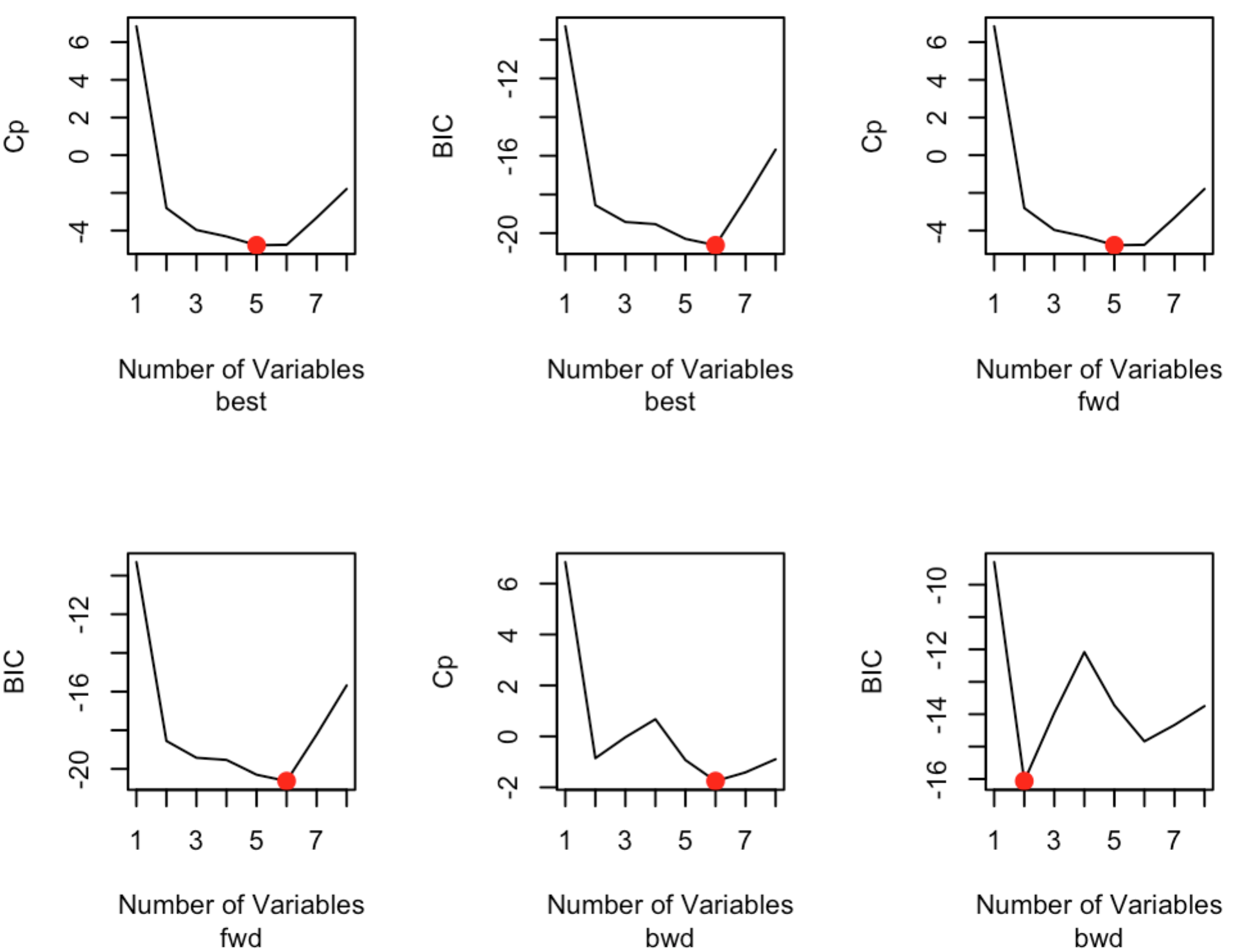
We first split the data into training data and test data. We used the training data to obtain the best model for each method, and used the test data to get the test MSE. Since we plan to do cross-validation to obtain the test MSE, the splitting procedures will be repeated nfold (e.g. 5 or 10) times and the training set will be different each time. Below we reported results from one of the training set (when nfold = 10 and set.seed(1)) just to show examples. The results are different every time when we have different training set.

Each three model gave 8 models with 1-8 variables. An example from best subset approach is given below (each model gave different 8 set of models). “\*” indicate the variable is included in such a model.



Then, we used Cp, BIC and CV to select the best model.

* Evaluation of the three models using Cp and BIC

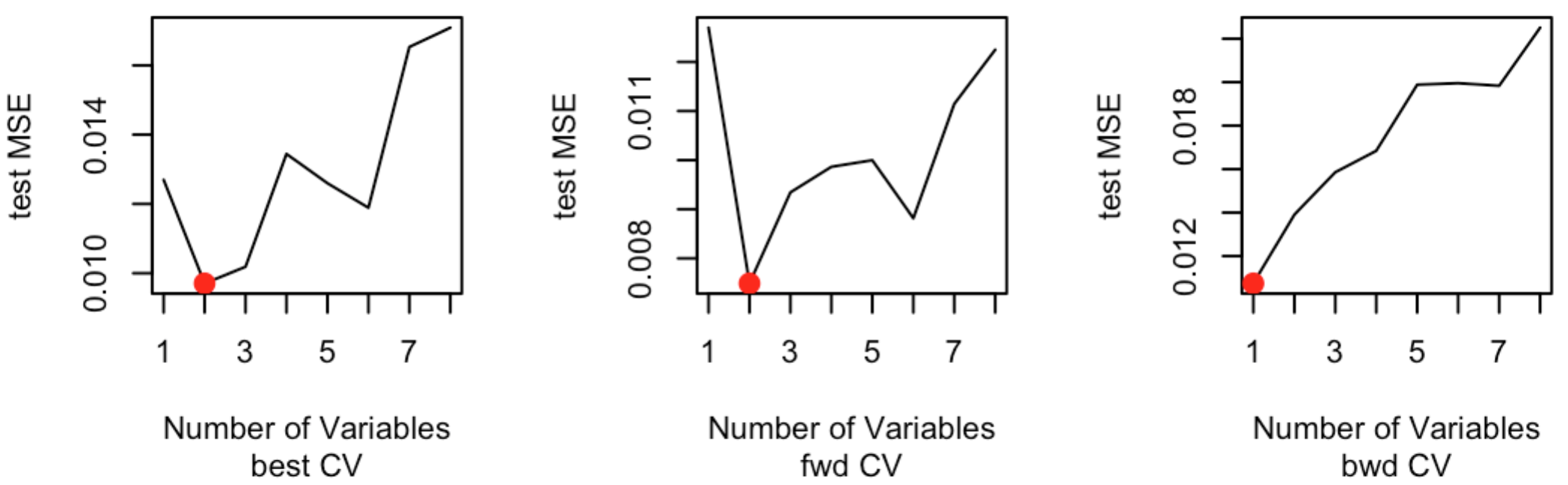


*Figure 2. Cp and BIC for models selected by the three methods*

The best models are selected when Cp/BIC is the lowest. The x-axis is the number of variables, which indicate the 8 models for each method. The red dots are the selected best model using corresponding model-selection and evaluation method. As shown from the graph, with this particular training set, Cp and BIC gave different best models for each method (but can be the same with some training sets).

* Evaluation of the three models using 10-fold cross-validation[[1]](#footnote-1)

This CV was conducted to select the best model within each method. The best models selected by CV are as below:



*Figure 3. cross-validation test MSE for models selected by the three methods*

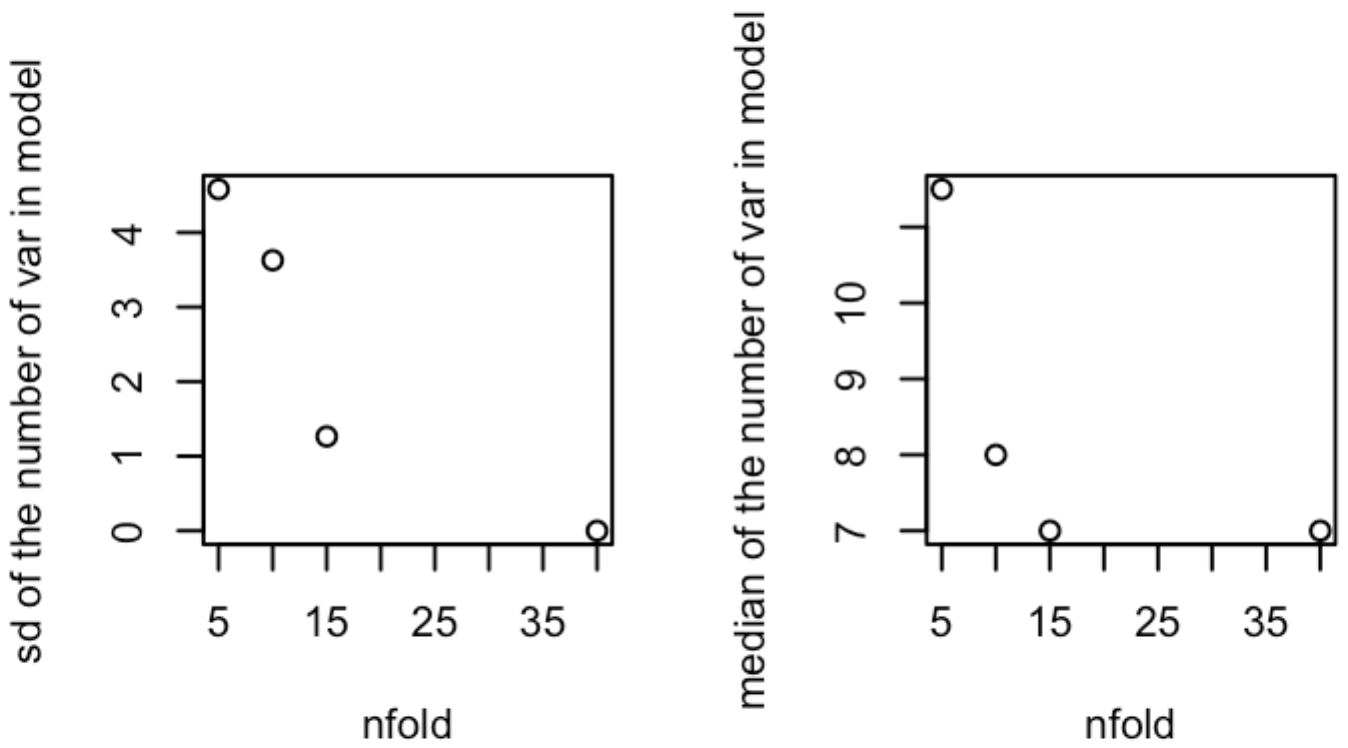
Again, for each method, different best models were given using CV, and they are also different from those selected using BIC and Cp (but it seems that CV tend to give smaller model).

1. Lasso method:

Within each training set, we used 10 fold cross-validation to obtain the best lambda that gives the lowest test MSE, and used the best lambda to get the best model for this training set (variables with non-zero coefficients).

For this particular training set, the best lambda is 0.02462796 (set.seed(1)); when applying the best lambda to the whole training set, it produced a best model, which contains 6 variables.

We noticed that when set.seed are different, the models produced are different and contain different variables. We did a brief check on the influence of the number of folds on the produced best model. We used the number of selected variable (*n*) as a simple indicator of whether the final models are the same. We obtained the standard deviation and median of *n* by setting set.seed different. The results are present below:



*Figure 4. sd and median of the number of variables in the best models using different set.seed(n)*

For this training set (the generalization, however, is a little limited), as nfold increases, variation decreases; also the median becomes more stable. Therefore when nfold is bigger, the best models given by this method should be more stable. In the model comparison in the next section, we set nfold =10 (instead of 5) for both CV in model selection and obtaining the test MSE to compare methods. Since there may still be some variation, we used different seeds to get values for each model and averaged them within a model. The one that yields the lowest (and among the lowest) test MSE should provide the best model.

1. Compare methods:

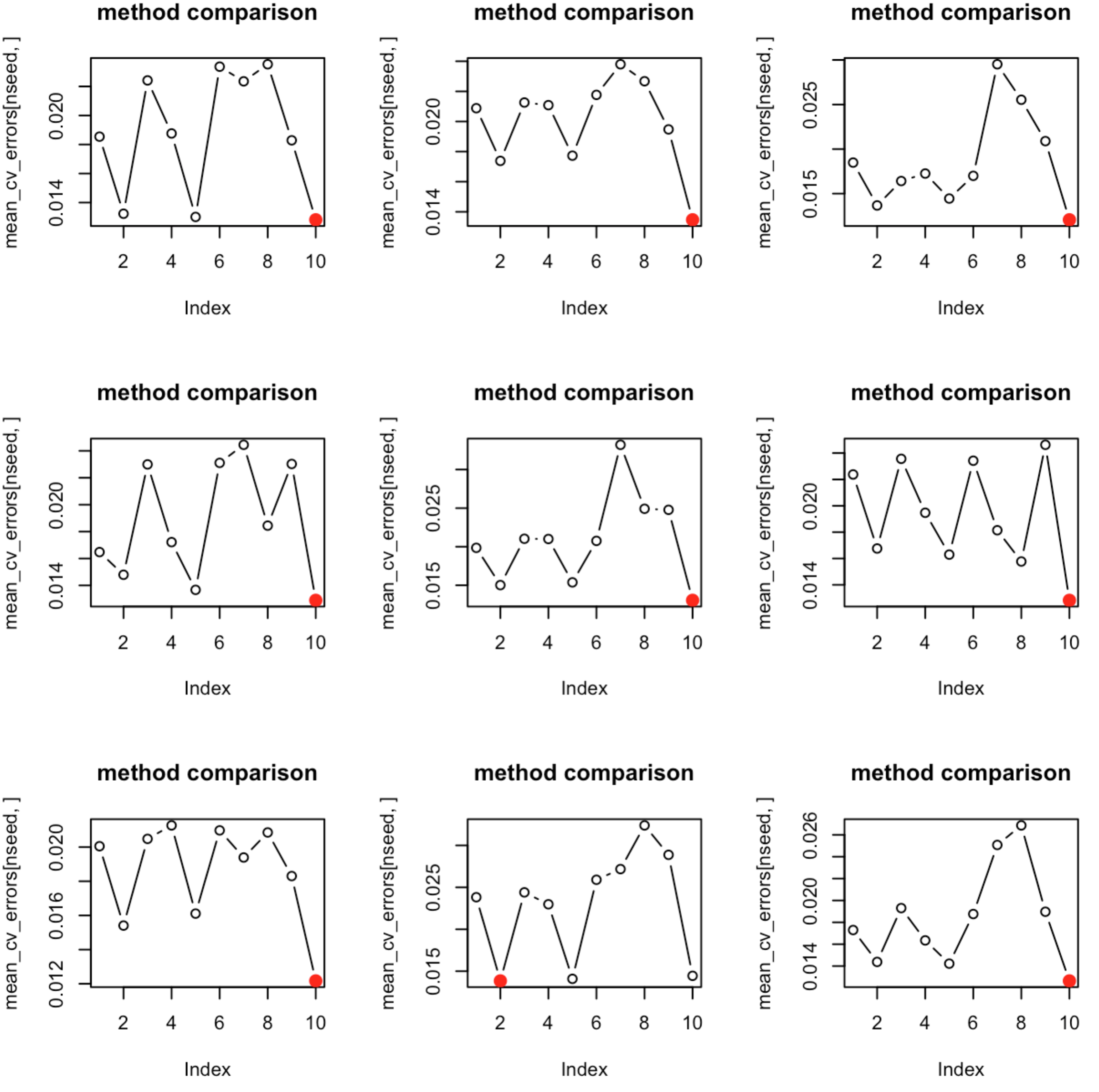
Set.seed(1:9)：

1-3: Best subset, forward, backward + Cp

4-6: Best subset, forward, backward + BIC

7-9: Best subset, forward, backward + CV

10: Lasso + CV



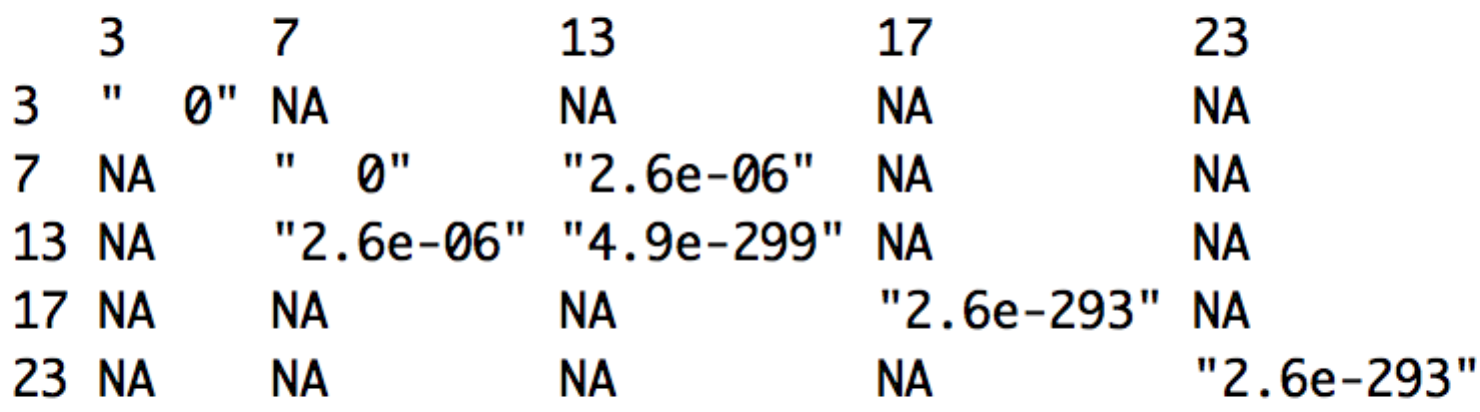
Most of time, Lasso + CV choose the best model among all the method. Forward + Cp/BIC gave low test MSE too.

Lasso compare with random forest:

1. Check the model:

1) Collinearity

We then checked the correlation between each variable in the selected model #4, since high correlation between variables may cause collinearity, and a simpler can be obtained if some correlated variables are dropped. The p-values of Pearson correlation of each pair of variables in #4 are given below (NA indicate p > 0.05)



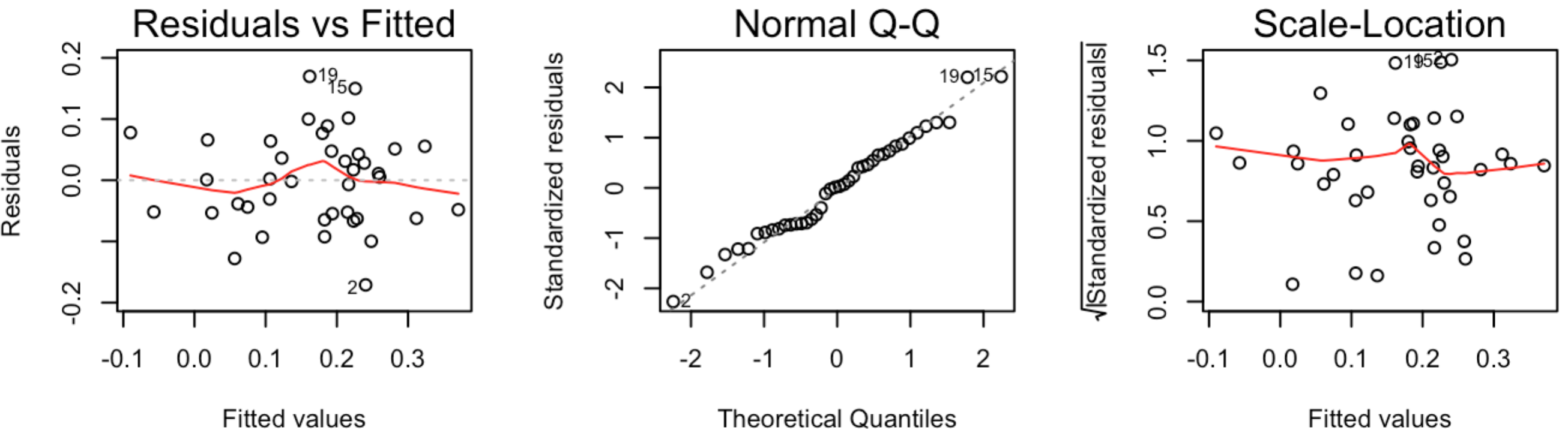
It can be seen that Rik (#7) and Pik3r3 (#13) is highly correlated. In the case of the collinearity, we can choose/drop either variable in the model. We then have two more models:

(7). Mapk1 ~ Akt2 + Pik3r3 + Pik3r1 + Rac1

(8). Mapk1 ~ Akt2 + Rik + Pik3r1 + Rac1

We performed another 10-fold cross-validation to evaluate whether (7) and (8) will have better performance (lower test MSE). The results show that the model with all the five variables (model 4) always has better performance (we leave out the graph here).

Besides collinearity, we also checked some assumptions of linear models, such as whether the relationship between the responses and predictors is linear and whether there are outliers that may distort the results (some outliers are shown in the boxplots). By plotting the model, we can see that 1) there is no apparent pattern in residuals vs. fitted values; 2) the QQ plot of standardized residuals suggests a normal distribution; 3) there is no obvious outliers (no standardized residual exceeds 3). Therefore the model satisfied the linear assumptions.



*Figure 6. Check the assumptions of linear models*

*Figure 6. correlation between Mapk1 and other variables:*

1. We did not use validation approach set because the sample size is small, the training and test set would be 20 if the validation set approach were used, which would yield results (the number of variables) with large variation (tested by using different set.seed(n)). The number of folds is set to be 10 instead of 5, because when k=5, the results also have larger variation, while the variation is small when k = 10 (always model with 1-3 variables are selected) [↑](#footnote-ref-1)