P8160 Project 1 - Designing a simulation study to compare variable Selection methods

Yuqi Miao (ym2771), Jiayi Shen (js5354), Jack Yan (xy2395), Jungang Zou (jz3183)

1 Objectives

Design a simulation study to investigate and illustrate how well each of the two methods (Step-wise forward method and LASSO) are in identifying weak and strong predictors. Show how missing "weak" predictors impacts the parameter estimations. Generate simulated data with a combination of strong, weak-but-correlated and weak-and-independent predictors.

2 Statistical methods to be studied

2.1 Step-wise forward method

Starting with the empty model, and iteratively adds the variables that best improves the model fit. That is often done by sequentially adding predictors with the largest reduction in AIC. For linear models,

$$AIC = n\ln(\sum_{i=1}^{n}(y_i - \widehat{y_i})^2/n) + 2p$$

where $\widehat{y_i}$ is the fitted values from a model, and p is the dimension of the model (i.e., number of predictors plus 1).

2.2 Automated LASSO regression LASSO

LASSO is another popular method for variable selection. It estimates the model parameters by optimizing a penalized loss function:

$$\underset{\beta}{min}\frac{1}{2n}\sum_{i=1}^{n}(y_i - x_i\beta)^2 + \lambda\sum_{k=1}^{p}|\beta_k|$$

where λ is a tunning parameter. We use Cross-validation (CV) to select λ for LASSO.

3 Scenarios to be investigated

In the simulation study, we fix the total number of predictors (p = 200), among which 100 are null predictors. The correlation coefficient between strong signals and weak but correlated signals is 0.5. Here we change the ratio of signal types r and sample size n.

Ratio of signal types

The strength of signals are defined by the following criteria:

Definition of strong predictors:

$$S_1 = \left\{ j : |\beta_j| > c\sqrt{\log(p)/n}, some\ c > 0, 1 \leq j \leq p \right\}$$

Definition of weak but correlated predictors:

$$S_2 = \left\{ j : 0 < |\beta_j| \leq c\sqrt{\log(p)/n}, some\ c > 0, corr(X_i, X_{j'}) \neq 0, some\ j' \in S_1, 1 \leq j \leq p \right\}$$

Definition of weak and independent predictors:

$$S_3 = \left\{ j : 0 < |\beta_j| \leq c\sqrt{\log(p)/n}, some\ c > 0, corr(X_i, X_{j'}) = 0, all\ j' \in S_1, 1 \leq j \leq p \right\}$$

Definition of null predictors:

$$S_4 = \{ j : |\beta_j| = 0, 1 \leq j \leq p \}$$

Here we consider 2 ratios of predictor types (r: strength: weak_but_correlated: weak_and_independent) 1) r = 5:1:4 2) r = 5:3:2

Sample size

The following 3 sample sizes are used in the simulation.

1) n = 200
2) n = 1000
3) n = 5000

We use n = 200 to test the robustness of the two methods facing high-dimensional data. We are also interested to see how the performance of the methods would improve with increased sample size.

4 Methods for generating data

4.1 Data generation

The data matrix `X` is generated by R function `MASS::mvrnorm`. A pre-defined covariance matrix is passed to `mvrnorm` function to specify the correlation between predictors. To ensure the positive definite attribute of covariance matrix, we restrict one strong predictor to be correlated with one weak predictor.

4.2 True model Parameter

Linear model parameters are randomly generated from a uniform distribution, subject to the definition of signal strength and ratio of predictor types. All the coefficients are positive values.

4.1 Generating true outcome Y

True distribution of outcome variable is defined as

$$Y \sim N(\boldsymbol{X}^T\boldsymbol{\beta}, \sigma^2)$$

Where $\boldsymbol{X}$ is the data matrix, $\boldsymbol{\beta}$ is the parameters vector and σ^2 is the constant variance in normal distribution. The variance is fixed at 9.

5 performance measures

5.1 Predictor identification performance

In order to compare the identification performance for these 2 methods, we regard the identification as a classification problem, where the signal predictors are defined as positive and null predictors are defined as negative. 3 indicators have been established:

$$recall = sensitivity = \left(\frac{True\ positive}{True\ positive + False\ negative}\right)$$

$$specificity = \left(\frac{True\ negative}{True\ negative + False\ positive}\right)$$

$$accurcy = \frac{True\ selection}{total\ number\ of\ predictors}$$

$$F_1 = \left(\frac{2}{recall^{-1} + precision^{-1}}\right)$$

Where precision is defined as above, and recall is defined as

$$precision = \left(\frac{True\ positive}{True\ positive + False\ positive}\right)$$

5.2 Parameter Estimation performance

To compare the paramater estimation performance, 3 indicators were calculated to evaluate the estimation: mean bias is calculated as the mean difference between true parameter and estimated parameters for a set of parameters, variance is defined as the variance of the estimated parameter among simulation. mean squared error (MSE) is also used to assess estimation performance.

- bias

$$\frac{1}{p}\sum_{j=1}^{p}(\hat{\boldsymbol{\beta}}_j - \boldsymbol{\beta}_j)$$

- variance

$$\frac{1}{p}\sum_{j=1}^{p}(\hat{\boldsymbol{\beta}}_j - \overline{\boldsymbol{\beta}_j})^2$$

- MSE

$$\frac{1}{p}\sum_{j=1}^{p}(\hat{\boldsymbol{\beta}}_j - \boldsymbol{\beta}_j)^2$$

6 Simulation results

6.1 Identification performance

F1 score (see Fig 1.) is used as an indicator for an overall performance assessment for identification signals. There is a clear trend for both methods that when the ratio of predictors to sample size is decreasing, the overall performance enhances, and there is no significant difference between this two methods when sample size is large enough comparing to predictor numbers, while in high dimensional situation (where n = p), stepwise has a poor performance.

In terms of variable selection size (see Fig 2.) , lasso tends to choose more predictors while stepwise tend to choose less overall, in the settings of this report, where there are 100 true signals and 100 noises, lasso select more predictors than true occasion in all scenarios, while stepwise tends to choose less. When comparing in terms of sample size, lasso performs stably in both high dimensional(where n = p) and normal scenario, but stepwise tends lose its select ability when the ratio between number of parameters and sample size is large.

Sensitivity (see Fig 3.) is an indicator for how many predictors can be captured by the methods. For lasso, strong predictors can be perfectly captures in all scenario with stable performance, while for weak predictors, difference occurs when correlated-to-independent ratios change, for less correlated scenario, lasso tends to perform better to capture weak and independent predictors compare to weak but correlated predictors, while in more correlated case, there is no clear difference between these two types of predictors, and performance for capture weak but correlated predictors is relatively stable. For stepwise methods, except for the same stability for capture weak but correlated predictors in more correlated scenario, there is no clear difference in different settings. When comparing the sensitivity of these 2 methods, lasso performs better overall but with a comparatively higher variance, and stepwise has a relatively poor performance in aspect of capturing weak predictors in all settings, but the performance stabilise at 0.1 to 0.25.

Correspondingly, the specificity (see Fig 4.) for stepwise is relatively higher since there is an overal trend for over-screening, but the specificity performance for lasso is also acceptable especially when the predictor-to-sample-size ratio decrease.

Finally, comparing the accuracy of these two methods (see Fig 5.) , lasso performs better in every category of predictors in 6 settings. Above all, in terms of variable identification performance, lasso tends to over-select predictors and performs stable in high dimensional scenarios, while stepwise tends to under-select predictors with a lower overall performance and less accuracy. As compare to the correlation bettween predictors, there is also a clear trend that a lower correlation ratio of predictors tends to decrease the overall variance of identification performance, this may due to the high correlation between predictors may reduce the overall variance of the sample space and leads a more stable selection results.

6.2 Parameter Estimation

When n = 200, the parameter estimate of stepwise selection is highly biased and unstable, while LASSO method has both low bias and variance for the esimation of all types of predictors. When n = 1000 and 5000, LASSO underestimates the coefficient of true signals, especially strong signals, while stepwise selection method is approximately unbiased in estimating strong signals. Stepwise selection also overestimates the effect of weak but correlated signals. Both methods overestimates the effect of null predictors.

6.3. comparision of missing vs no missing

In this part, we will discuss how missing “weak” signals impact the coefficients in linear models. Due to the definition of “missing variables”, we consider 2 situations. The first part is the original “no missing” models, which we have discussed above. The second part is the “missing” models, which were constructed by deleting the weak_but_correlated data and weak_independent data. After model constructions, we‘d like to analyze the performance of “no missing” models and “missing” models.

The procedure to construct models in 2 situations is as follows:

- Use calculated covariance matrix to generate “no missing” data X_{true}, and generate random noise ϵ.
- Use pre-calculated β_{true}, “no missing” data X_{true} and random noise ϵ to calculate response variable $Y = X_{true}\beta_{true} + \epsilon$
- Apply lasso and stepwise regression on “no missing” data X_{true}, to get the estimated “no missing” model $\hat{Y} = X_{true}\hat{\beta_{no}}$ and estimated coefficient $\hat{\beta_{no}}$.
- Delete the weak_but_correlated and weak_independent variables in “no missing” data X_{true}, to get “missing” data $X_{missing}$.
- Apply lasso and stepwise regression on “missing” data $X_{missing}$, to get the estimated “missing” model $\hat{Y} = X_{missing}\hat{\beta_{missing}}$ and estimated coefficient $\hat{\beta_{missing}}$.

1) bias

As mentioned above, bias is an important criterion to assess model performance. As in Fig , we can see that with the increasing number of samples, the biases for both models are becoming small, because of the average effect on outliers. Another important criterion variance also shows negative relationship when sample size becomes large. So for both “missing” and “no missing” models, the large number of sample size has positive effect to decrease both bias and variance.

For lasso models and stepwise models, we find different result. For stepwise model, we can find the average of bias has very little difference between “missing” and “no missing” data. However, things become distinct for lasso models. With the “missing” data, we find lasso models perform better than “no missing” data, regardless of sample size. This significant difference may result from the different mechanism of model selection. For stepwise model, a variable will be decided to include in the model or not. On the other hand, the lasso model uses shrinkage method and considers the co-effect among all the variables. If a variable is not important, lasso model will gradually shrink its coefficient to 0 by considering the relationships for other variables. As a result, in model selection, lasso model is easy to be influenced by weak signals. On the contrary, stepwise model is robust for weak signals.

2) RMSE

After we draw the conclusion of bias, we need to analyse the rmse for both models. As in Fig , we can see that with the increasing number of samples, the rmse for both models are becoming small, due to the decrease for both bias and variance.

For lasso models and stepwise models, we find the result same as bias. For stepwise model, the average of rmse has very little difference between “missing” and “no missing” data. However, the lasso model has smaller rmse and large rmse variance with “missing” data. This result shows the disappearance of weak signals will improve the performance of lasso model, and has little significance for stepwise.

However, as is known to us, lasso model is a regularized model that can decrease the predictive error on test dataset. This property indicates lasso models have better generalization ability over all the sample space. Due to the bias nad variance tradeoff, lasso models increase its bias to decrease its variance. So, on training dataset, the rmse of lasso will be larger than rmse of stepwise model. However, on the test dataset, the rmse of lasso will be smaller than rmse of stepwise model. In conclusion, stepwise model is likely to be overfitting on the training dataset.

Finally, we can draw the conclusion, that stepwise model is robust to weak signals, and lasso model shows a significant improvement with “missing” data.

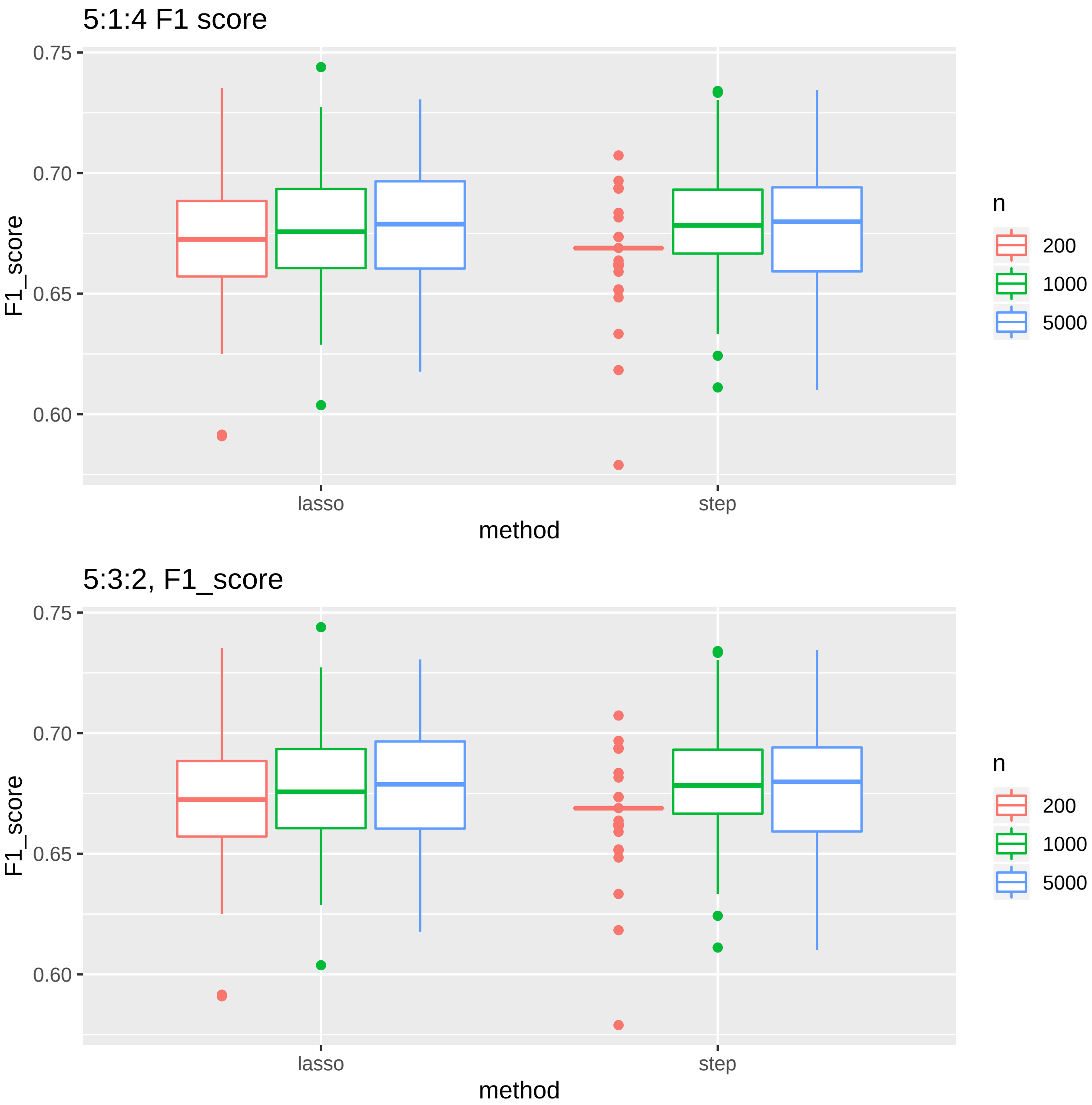


Figure 1: F1 score for 2 variable selection methods in 6 settings

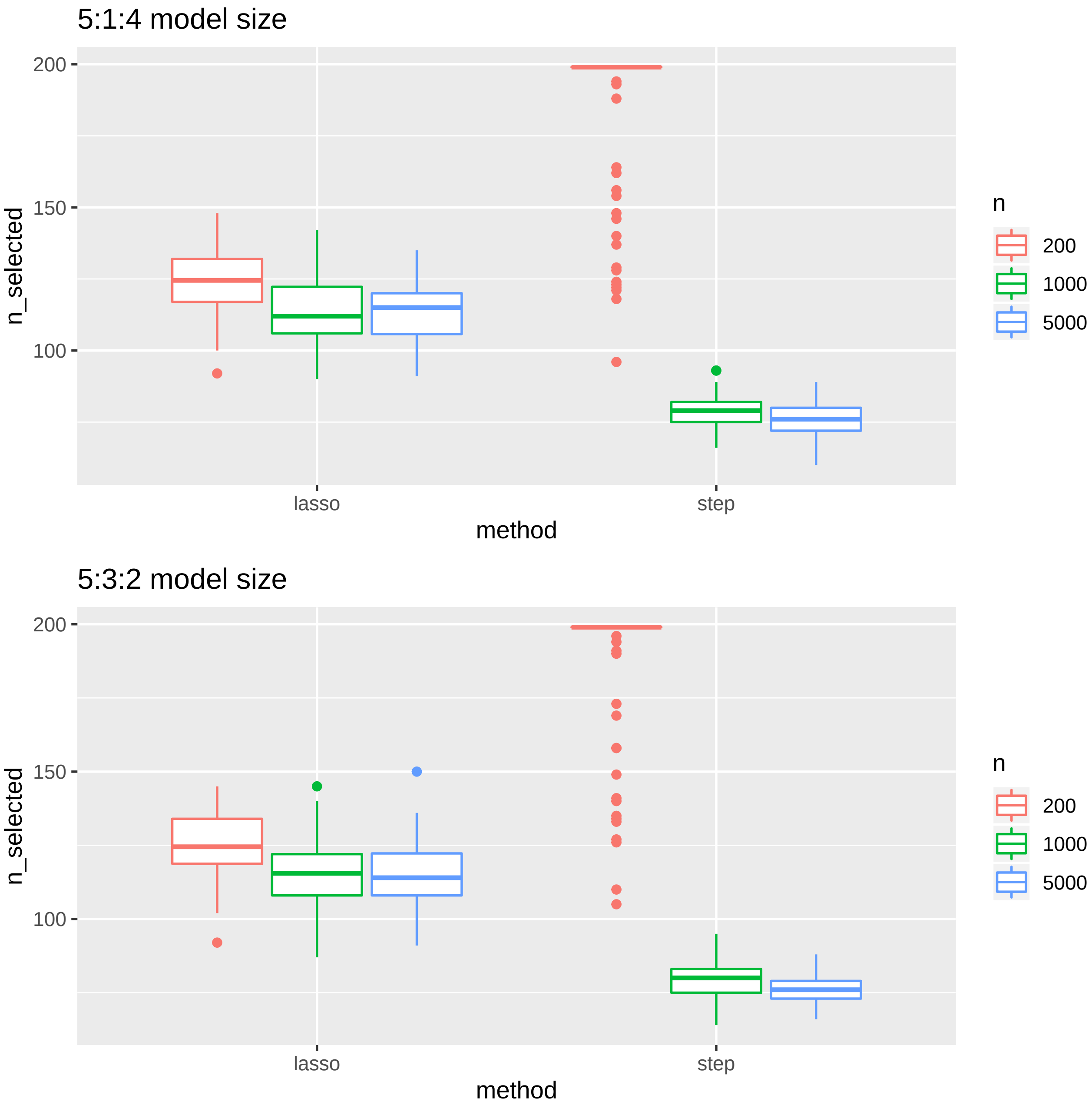


Figure 2: Model size for 2 variable selection methods in 6 settings

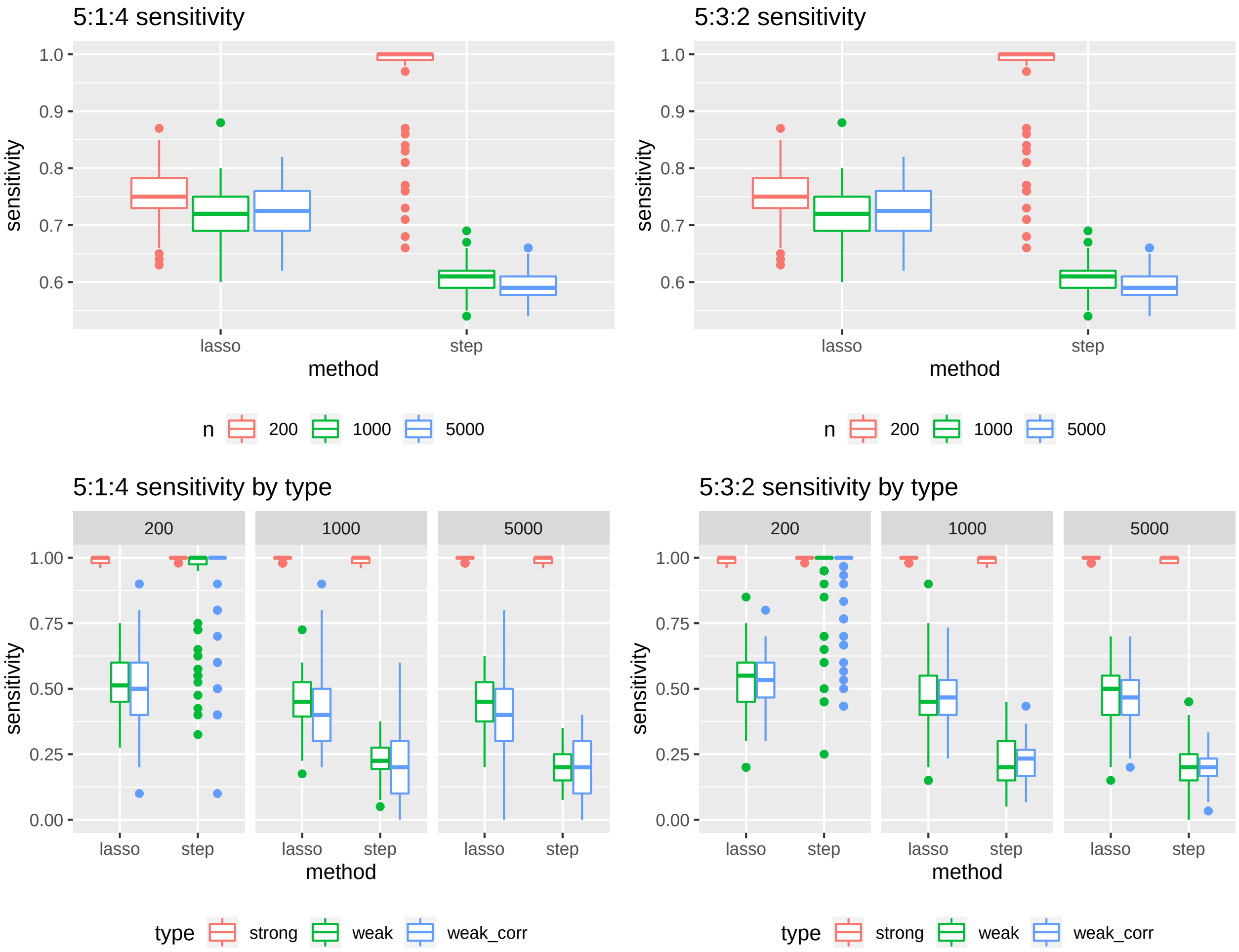


Figure 3: Sensitivity for 2 variable selection methods in 6 settings

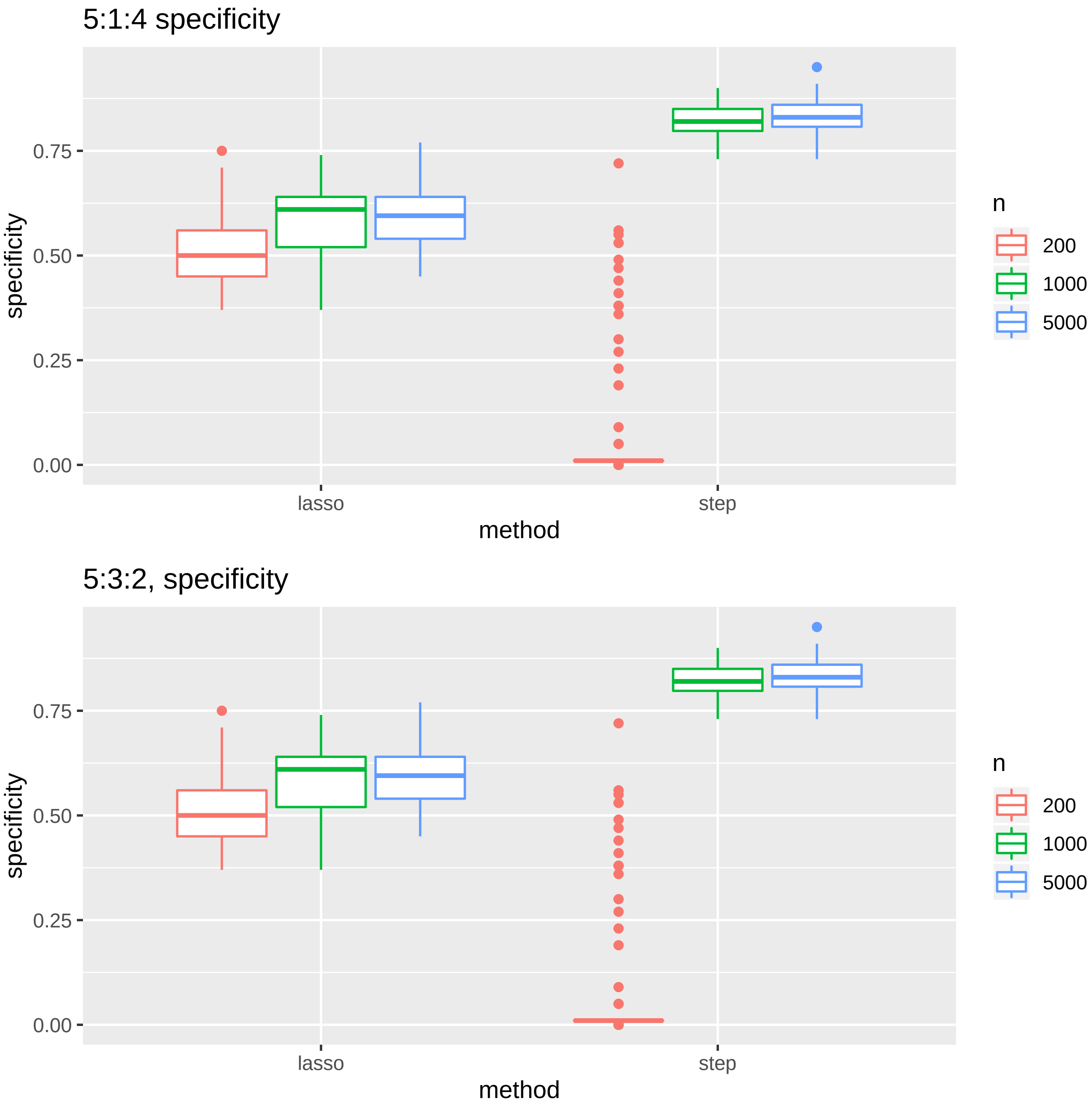


Figure 4: Specificity for 2 variable selection methods in 6 settings

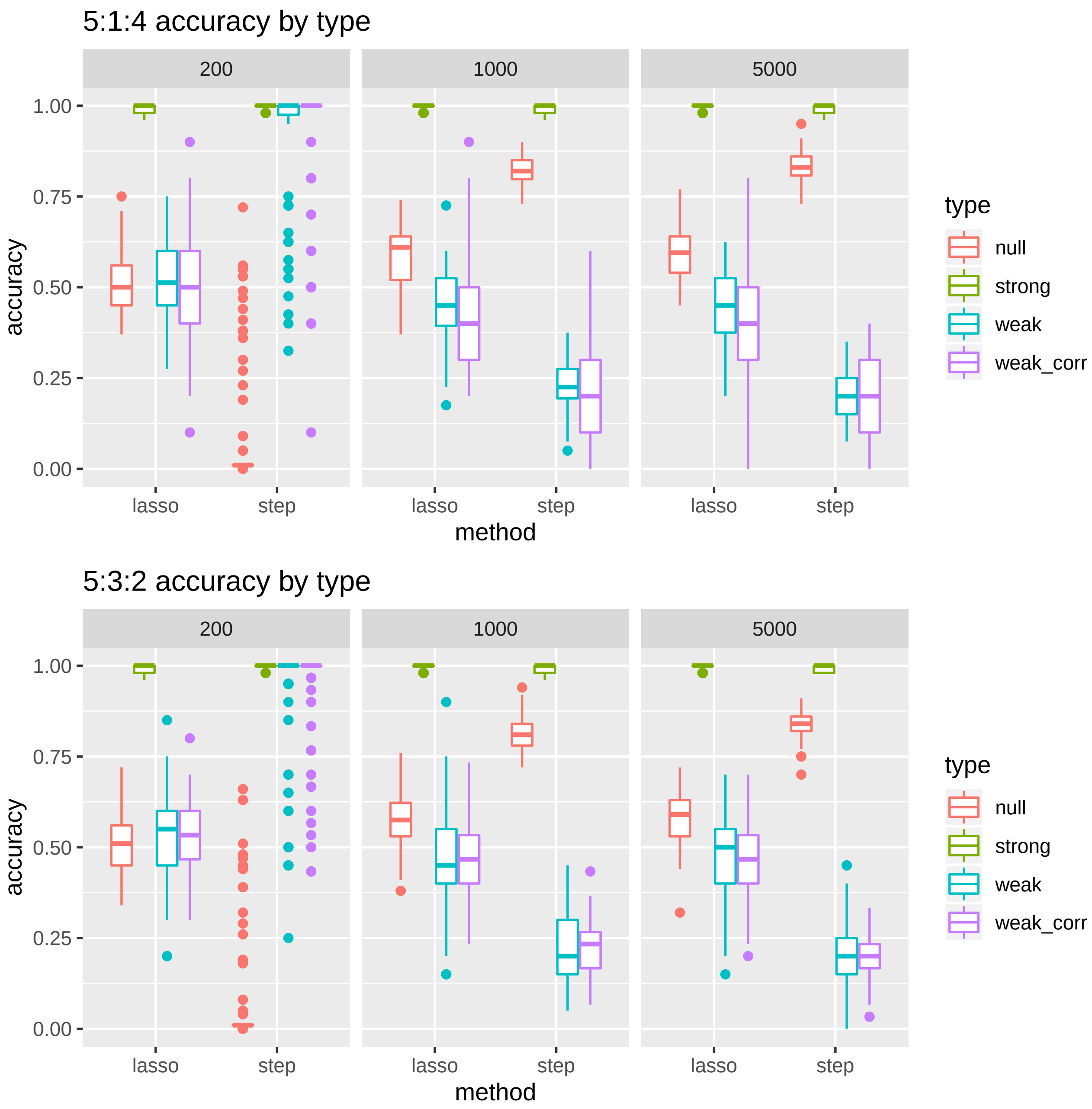


Figure 5: Specificity for 2 variable selection methods in 6 settings

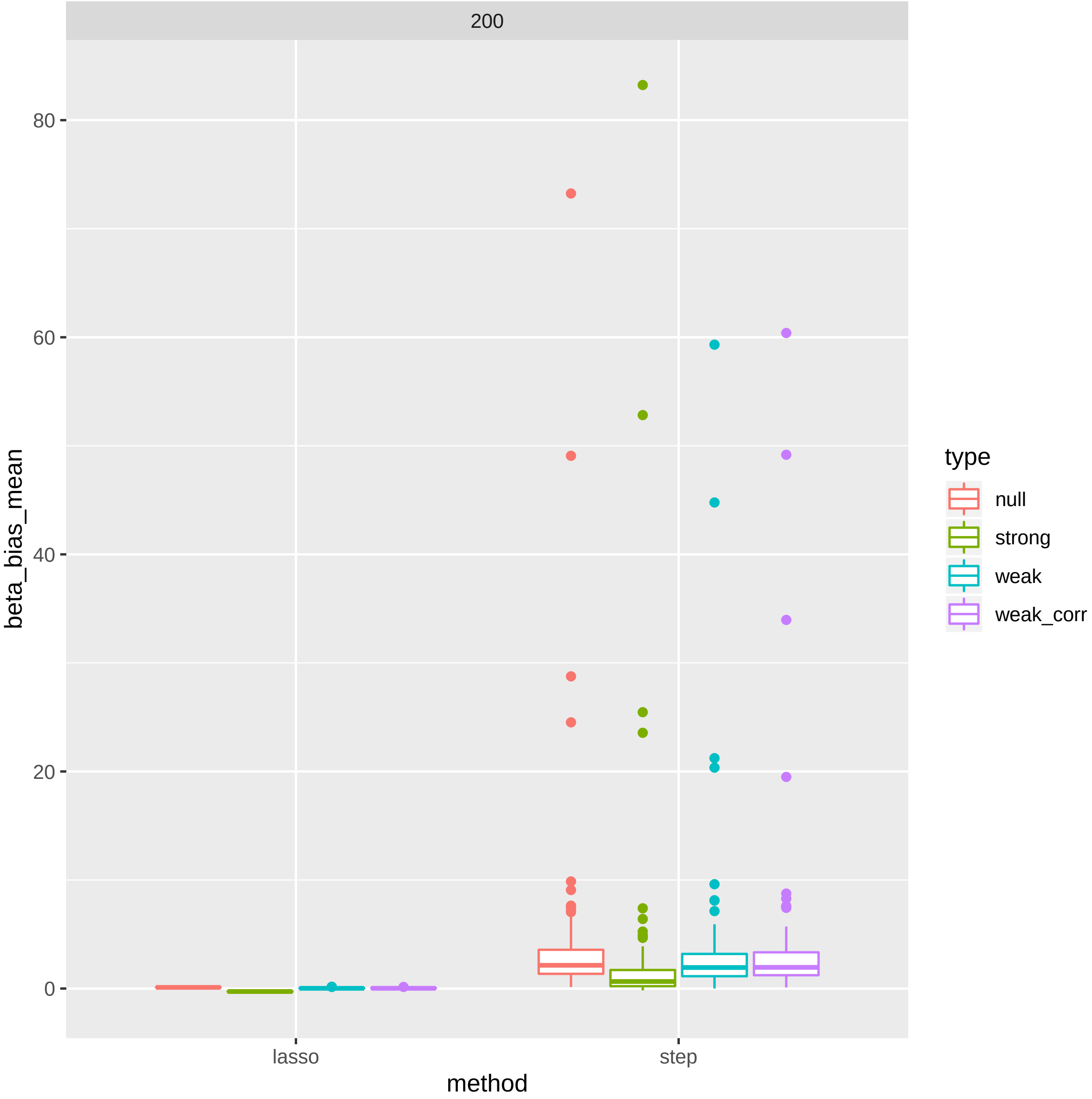


Figure 6: Mean beta bias n = 200, 40 weak-but-correlated predictors

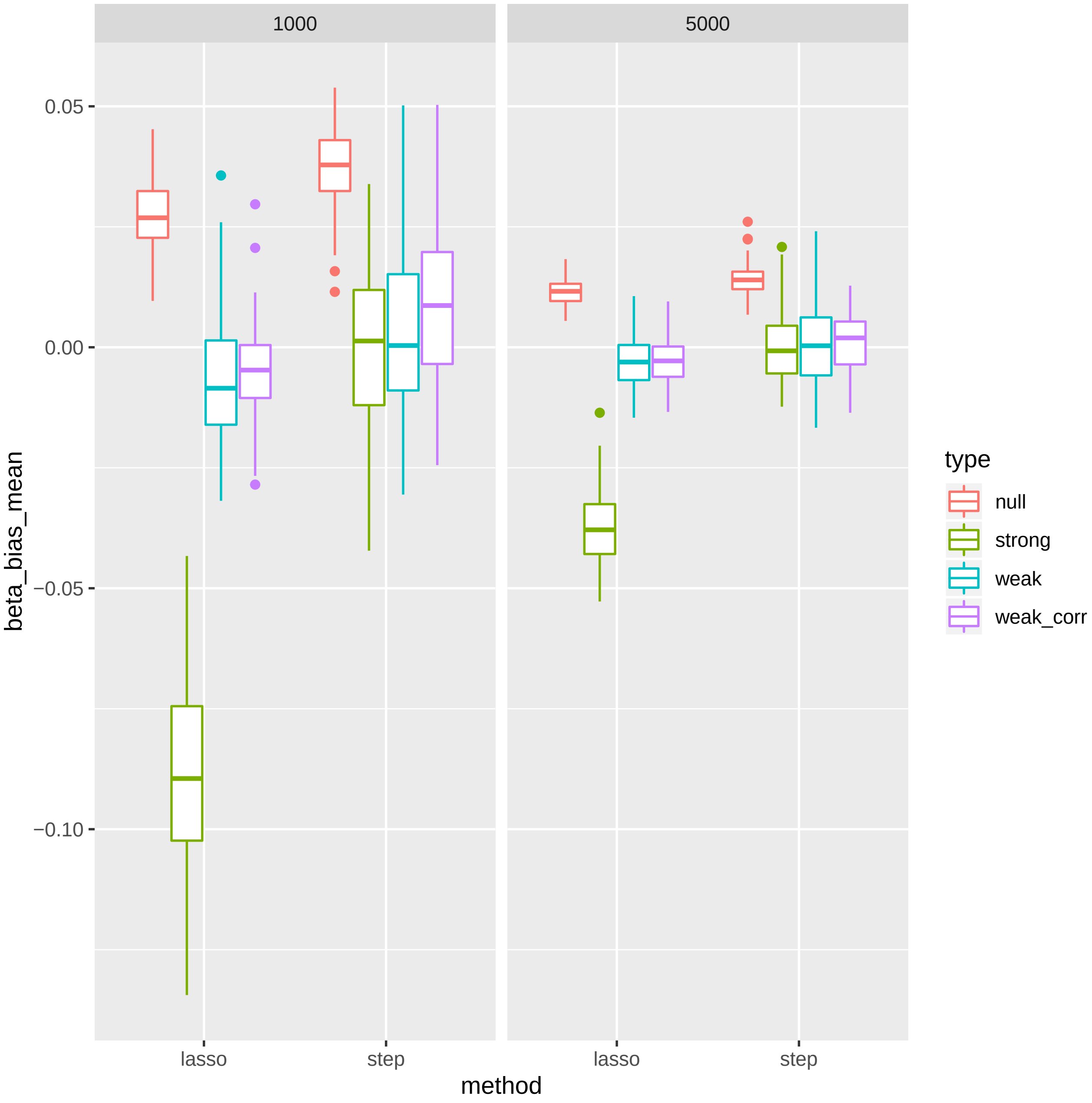


Figure 7: Mean beta bias n = 200, 40 weak-but-correlated predictors

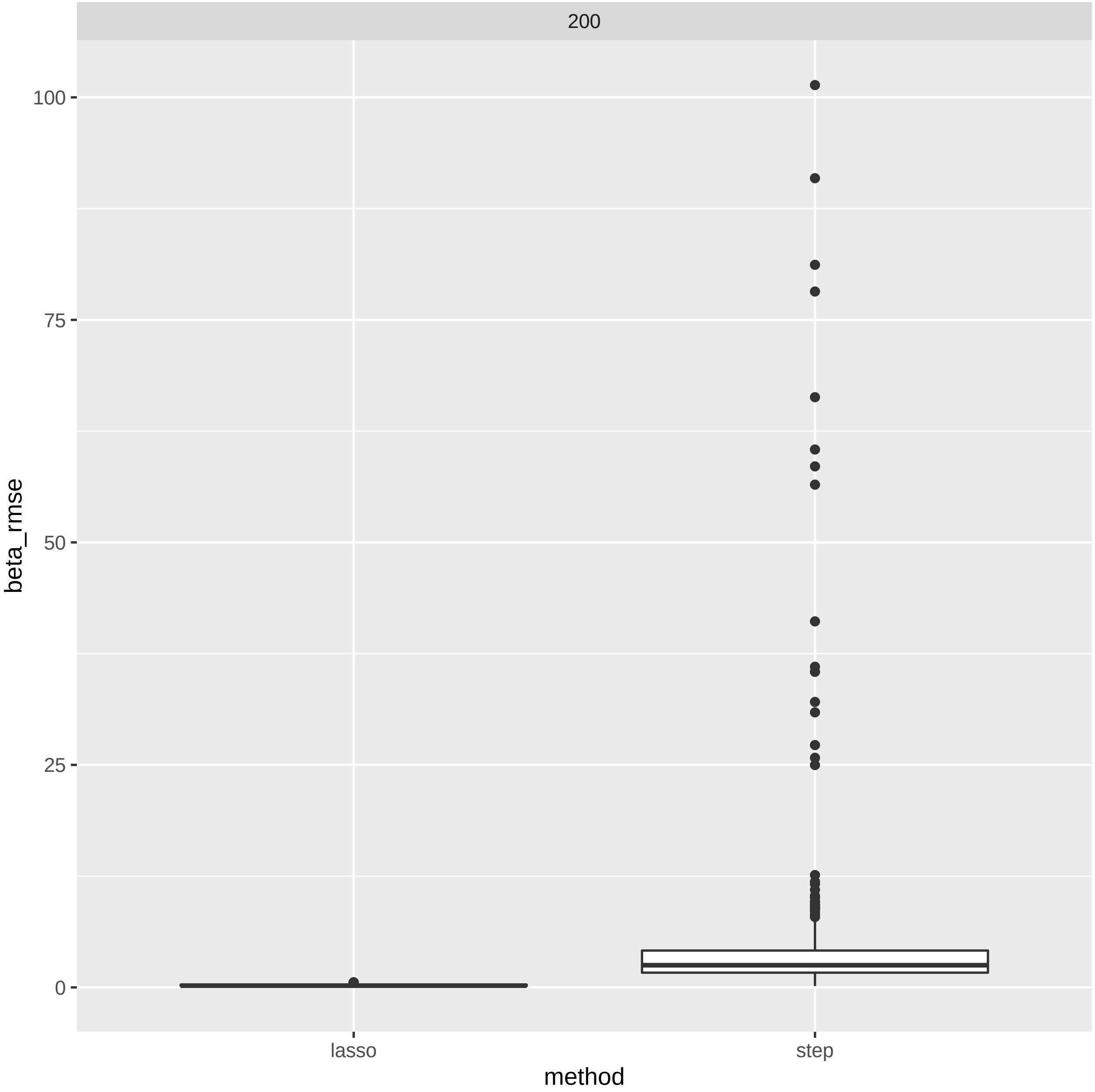


Figure 8: Mean beta bias n = 1000, 2000

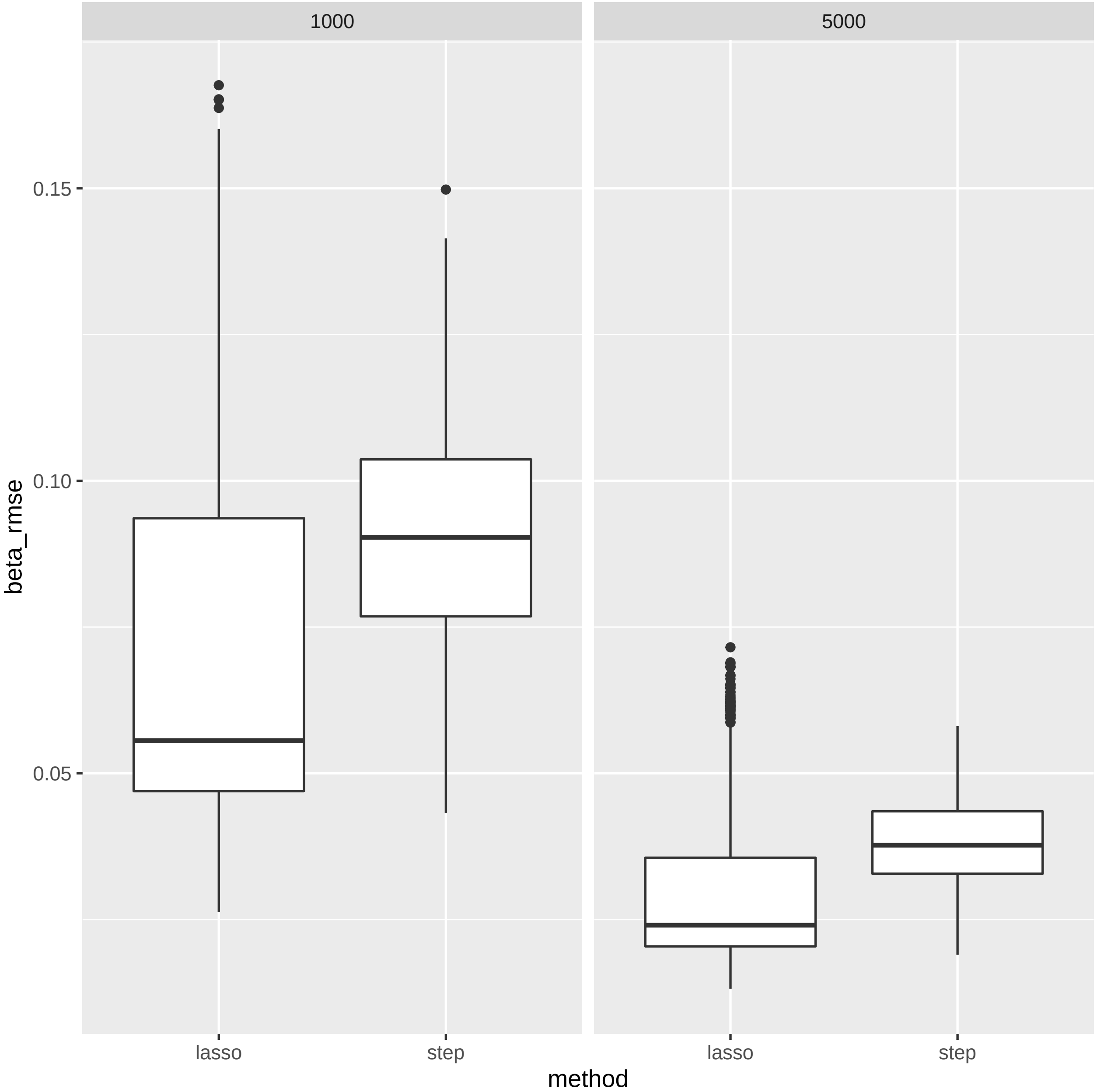
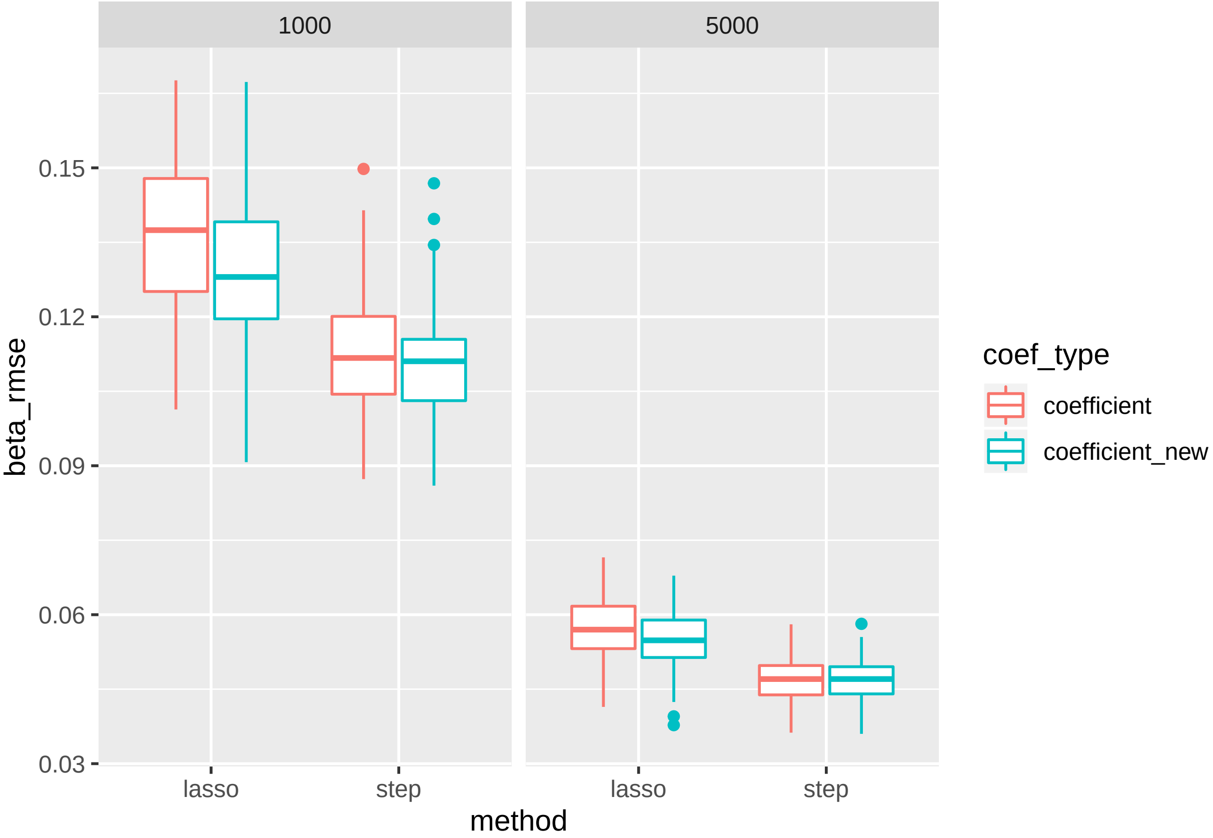
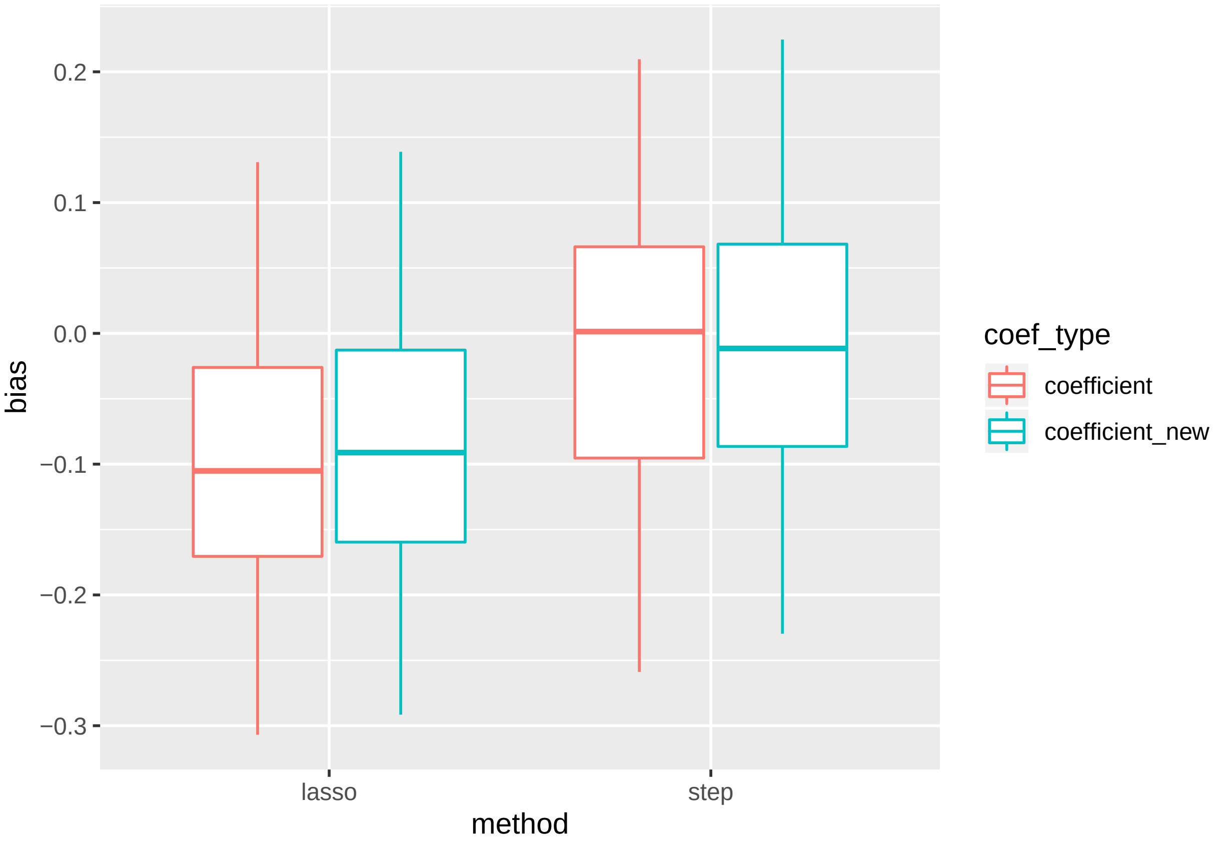


Figure 9: Beta RMSE n = 1000, 2000

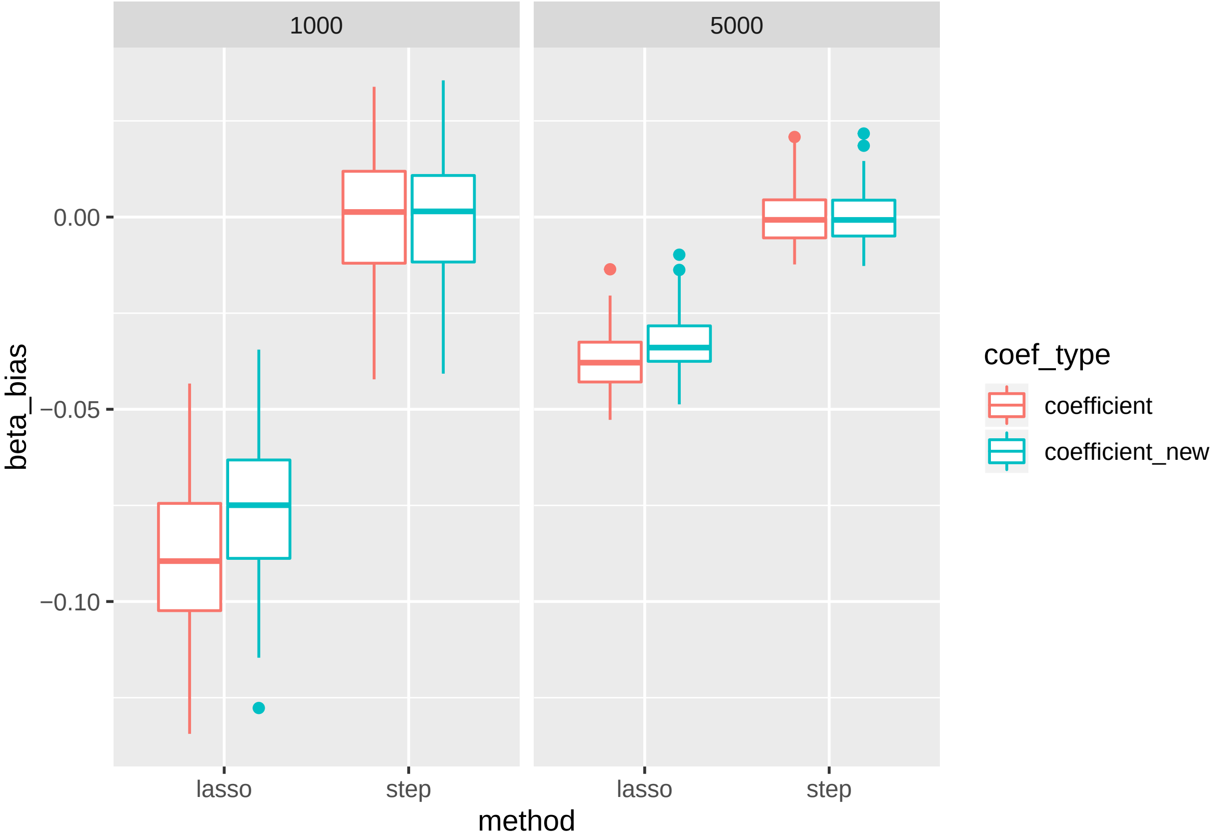
```r
temp_df %>%
  summarize(beta_rmse = sqrt(mean((coef-true_beta)^2)) ) %>%
  ggplot(aes(x = method, y = beta_rmse, color = coef_type)) +
    geom_boxplot() +
    facet_grid(~n)
```



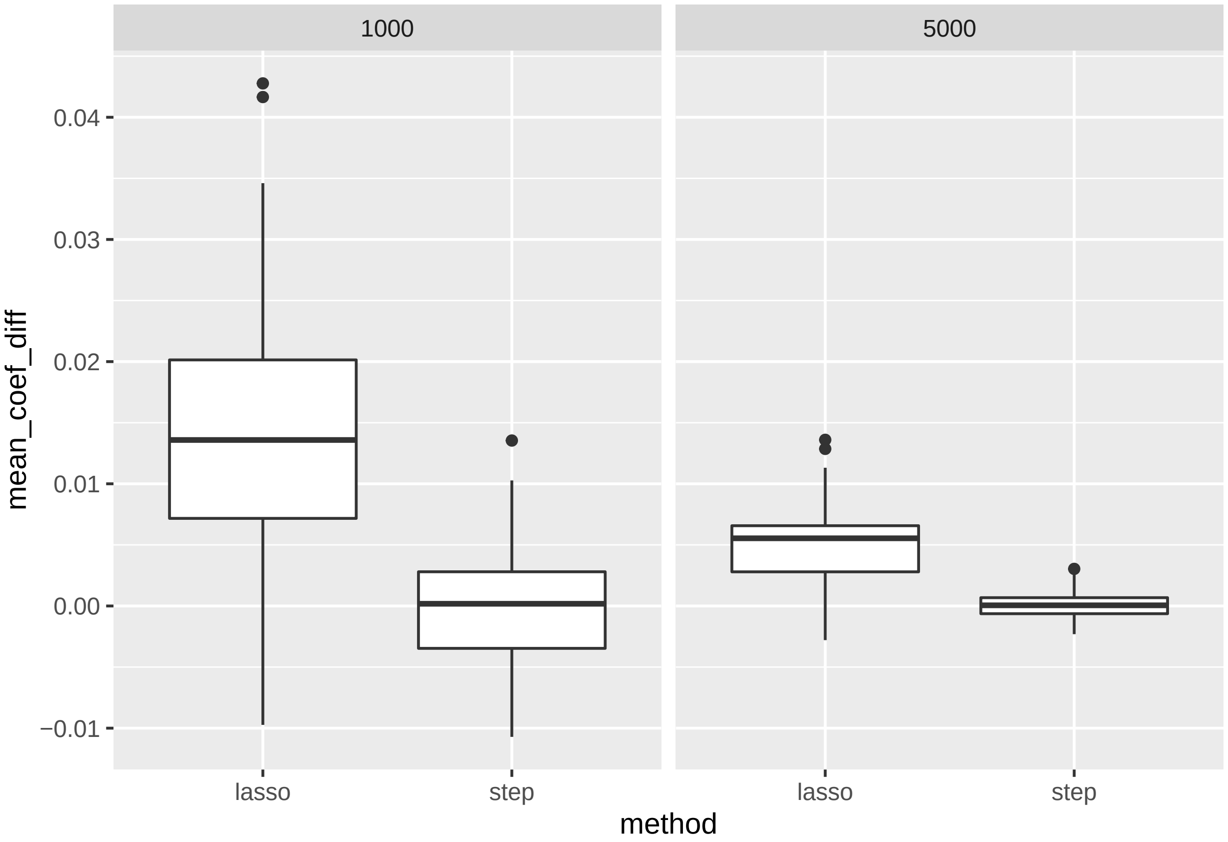
```r
temp_df %>%
  filter(true_beta == unique(temp_df$true_beta)[3]) %>%
  ggplot(aes(x = method, y = bias, color = coef_type)) +
    geom_boxplot()
```



```
beta_missing_weak_combined %>%
  filter(n != "200") %>%
  filter(type == "strong") %>%
  group_by(idx_sim, method, n, coef_type) %>%
  summarize(beta_bias = mean(bias) ) %>%
  ggplot(aes(x = method, y = beta_bias, color = coef_type)) +
    geom_boxplot() +
    facet_grid(~n)
```



```
bind_cols(beta_missing_weak, beta_missing_weak_new) %>%
  mutate(coef_diff = coefficient_new - coefficient) %>%
  filter(n != "200") %>%
  filter(type == "strong") %>%
  group_by(idx_sim, method, n) %>%
  summarize(mean_coef_diff = mean(coef_diff) ) %>%
  ggplot(aes(x = method, y = mean_coef_diff)) +
    geom_boxplot() +
    facet_grid(~n)
```


1000
5000
mean_coef_diff
0.04
0.03
0.02
0.01
0.00
−0.01
lasso
step
lasso
step
method