Machine Learning for Economists

Lab 1: Regularization

University of Cologne

Summer 2024

Intro

In this exercise, we want to investigate the effect of regularization on prediction. Our setting is very simple and in line with the linear model in the lecture notes. In particular, we estimate the model

$$y_i = \beta_1 x_{1i} + \beta_2 x_{2i} + \dots \beta_p x_{pi} + \epsilon_i$$

and we set

$$\beta_1 = 2, \beta_2 = 3, \beta_j = 0 \text{ for } j > 2$$

We will work on a simulation exercise, i.e. we

1. simulate data according to the model above many times
2. estimate the model parameters using standard OLS, LASSO or Ridge
3. compare bias and variance for an out-of-sample prediction

Simulation code

Make sure you understand the simulation code

Simulation code

```r
library(glmnet)

nsim = 1000  # Number of simulations
nobs = 100  # Number of observations in the simulated dataset
p = 10      # Number of regressors
beta = c(2,3, rep(0,p-2))  # Coefficient vector
                           # (2 for beta1, 3 for beta2,
                           # 0s for remaining betas)

x0 = c(rep(2, p))  # out-of-sample observation

# Initialize vectors so that we can store results
f0_ols   = c()
f0_lasso = c()
f0_ridge = c()
```

Simulation code

```
for (s in 1:nsim) {

  X = MASS::mvrnorm(nobs, #Randomly drawn X variables
                    mu = rep(0, p),
                    Sigma = diag(p)
                    )

  y = X %*%beta + rnorm(nobs)  # Simulate y with normal noise

  # Fit OLS model
  fitOLS = lm(y~X)
  f0_ols[s] = fitOLS$coefficients%*%c(1, x0)  # Prediction at x0

  # Fit LASSO regression
  fitLASSO = cv.glmnet(X, y, alpha = 1)
  f0_lasso[s] = t(coef(fitLASSO, s="lambda.1se"))%*%c(1, x0) %>% as.num

  # Fit Ridge regression
  fitRidge = cv.glmnet(X, y, alpha = 0)
  f0_ridge[s] = t(coef(fitRidge, s="lambda.1se"))%*%c(1, x0) %>% as.num

}
```

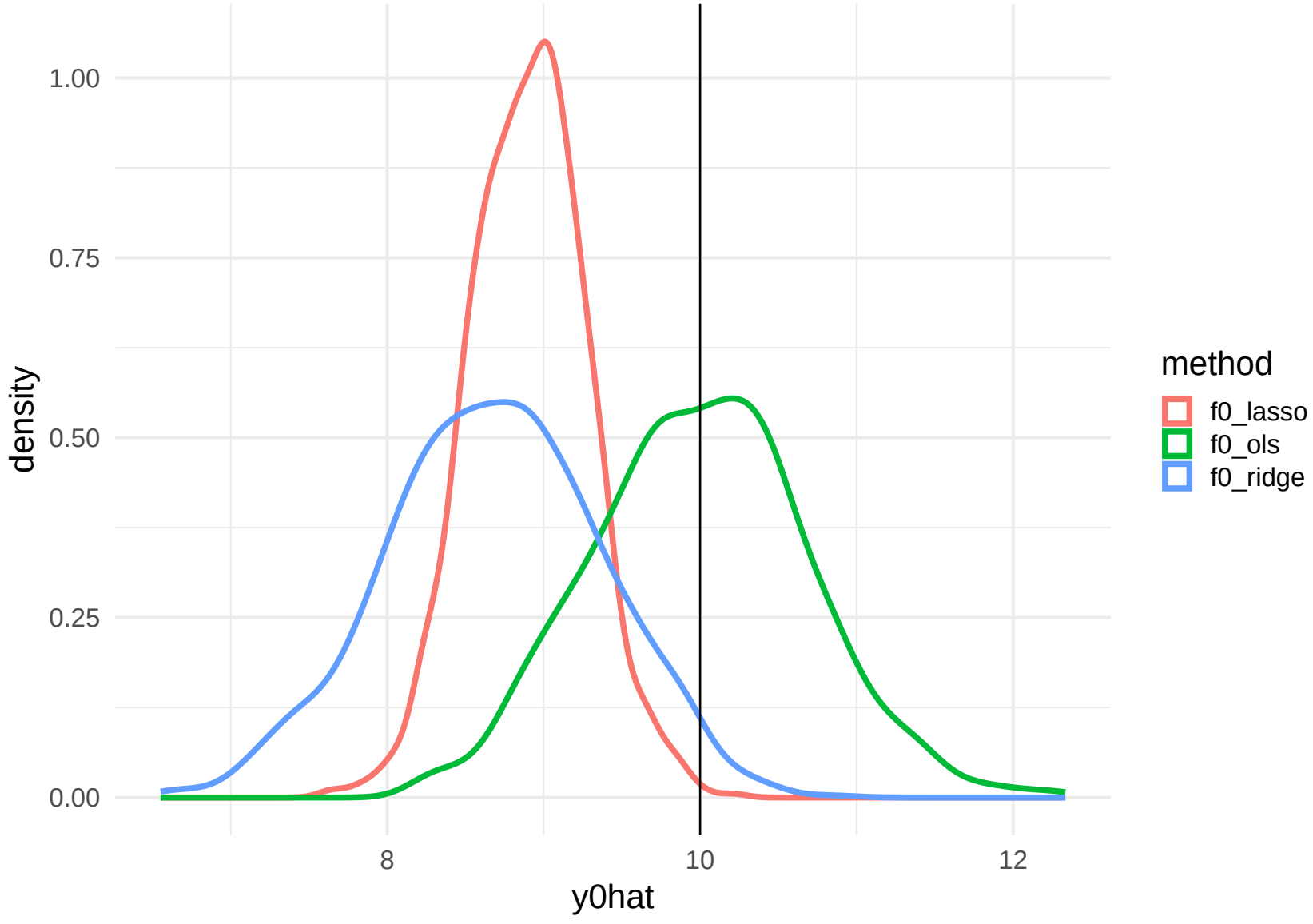
1. Out-of-sample prediction at $x_0 = (2, 2, \ldots, 2)$

$$E[y_0|x_0 = (2, 2, \ldots, 2)] = 2 \cdot 2 + 3 \cdot 2 + 0 \cdot 2 + \ldots = 10$$

2. Histograms of $\hat{f}(x_0)$

Run the code as is. Plot the distribution of predictions $\hat{f}(x_0)$ for the different models. What do you observe?

2. Histograms of $\hat{f}(x_0)$



2. Histograms of $\hat{f}(x_0)$

```
y0 = beta%*%x0  # Expected value at x0

tibble(f0_ols, f0_lasso, f0_ridge) %>%
  gather(key = 'method', value = 'y0hat') %>%
  ggplot(aes(x = y0hat, color = method, group = method)) +
  geom_density(size = 1.5) +
  geom_vline(xintercept = y0) +
  theme_minimal(base_size = 18)
```

3. Bias, variance and MSE

Compute bias, variance and mean squared error for the three different models at x_0, our out-of-sample observation.

3. Bias, variance and MSE

```
## # A tibble: 3 x 4
##   method       bias variance   mse
##   <chr>       <dbl>    <dbl> <dbl>
## 1 f0_lasso -1.10        0.137 1.34
## 2 f0_ols   -0.00234     0.478 0.478
## 3 f0_ridge -1.32        0.473 2.22
```

3. Bias, variance and MSE

```
tibble(f0_ols, f0_lasso, f0_ridge) %>%
  gather(key = 'method', value = 'y0hat') %>%
  group_by(method) %>%
  summarise(bias = mean(y0hat - 10),
            variance = var(y0hat),
            mse = mean((y0hat-10)^2)
  )
```

4. Number of irrelevant regressors

How do bias and variance depend on the number of irrelevant regressors?

4. Number of irrelevant regressors

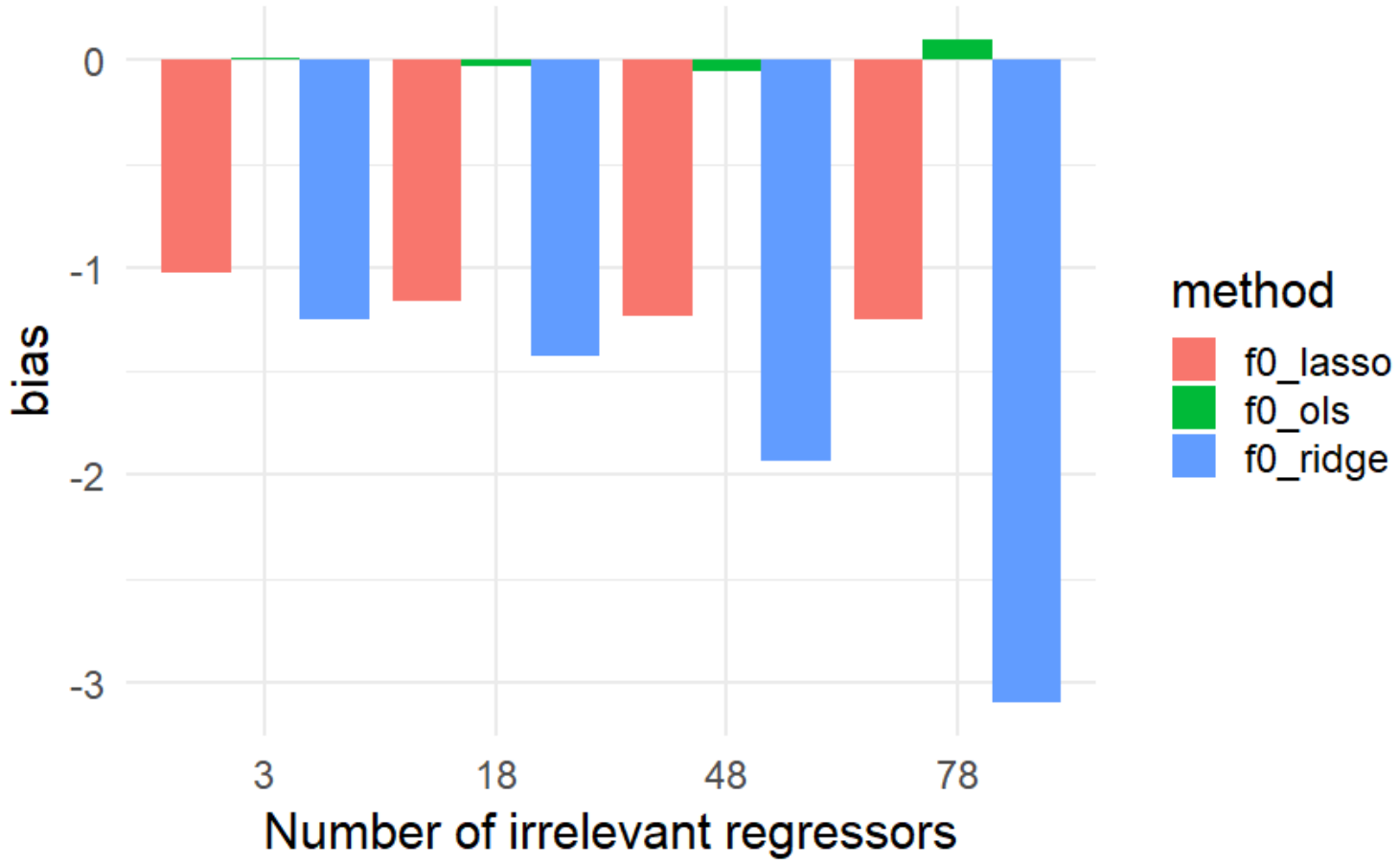
Wrap code into function for convenience

```
simResults = function(nsims, nobss, ps){

nsim = nsims  # Number of simulations
nobs = nobss  # Number of observations in the simulated dataset
p = ps       # Number of regressors

....

return(tibble(f0_ols, f0_lasso, f0_ridge))
}
```

4. Number of irrelevant regressors

```
nIR = c(3, 18, 48, 78)

dfAll = tibble()
for (i in 1:length(nIR)) {

  temp = simResults(nsims = 1000, nobss = 100, ps = nIR[i]+2)

  dfAll = bind_rows(dfAll,
                    temp %>% mutate(numIR = nIR[i]))
}
```

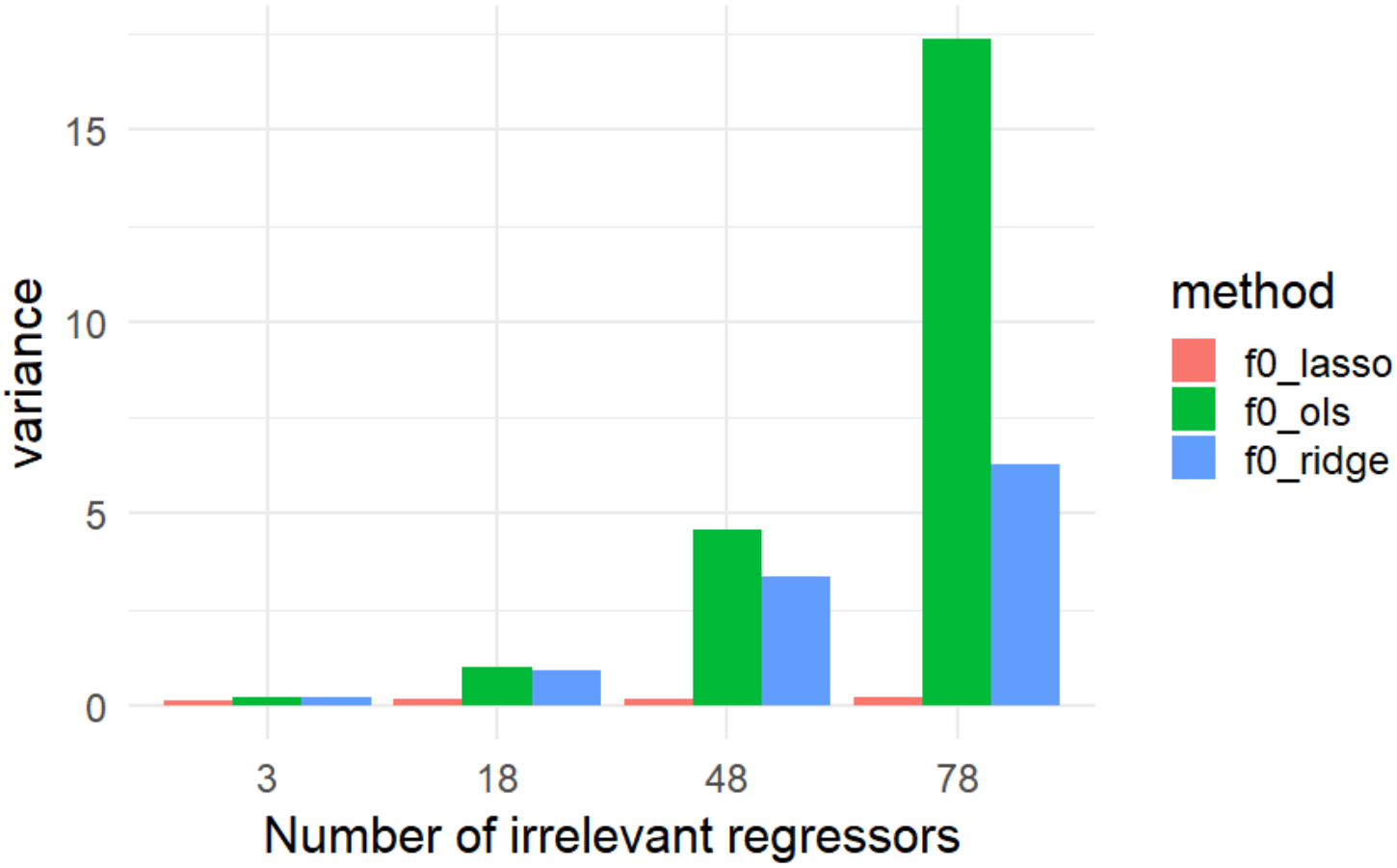
4. Number of irrelevant regressors



4. Number of irrelevant regressors

```
dfAll %>%
  gather(key = 'method', value = 'y0hat', -numIR) %>%
  group_by(method, numIR) %>%
  summarise(bias = mean(y0hat - 10),
            variance = var(y0hat),
            mse = mean((y0hat-10)^2)
  ) %>%
  ggplot(aes(x = factor(numIR), y = bias, fill = method)) +
  geom_col(position = 'dodge') +
  labs(x = 'Number of irrelevant regressors') +
  theme_minimal(base_size = 18)
```

4. Number of irrelevant regressors



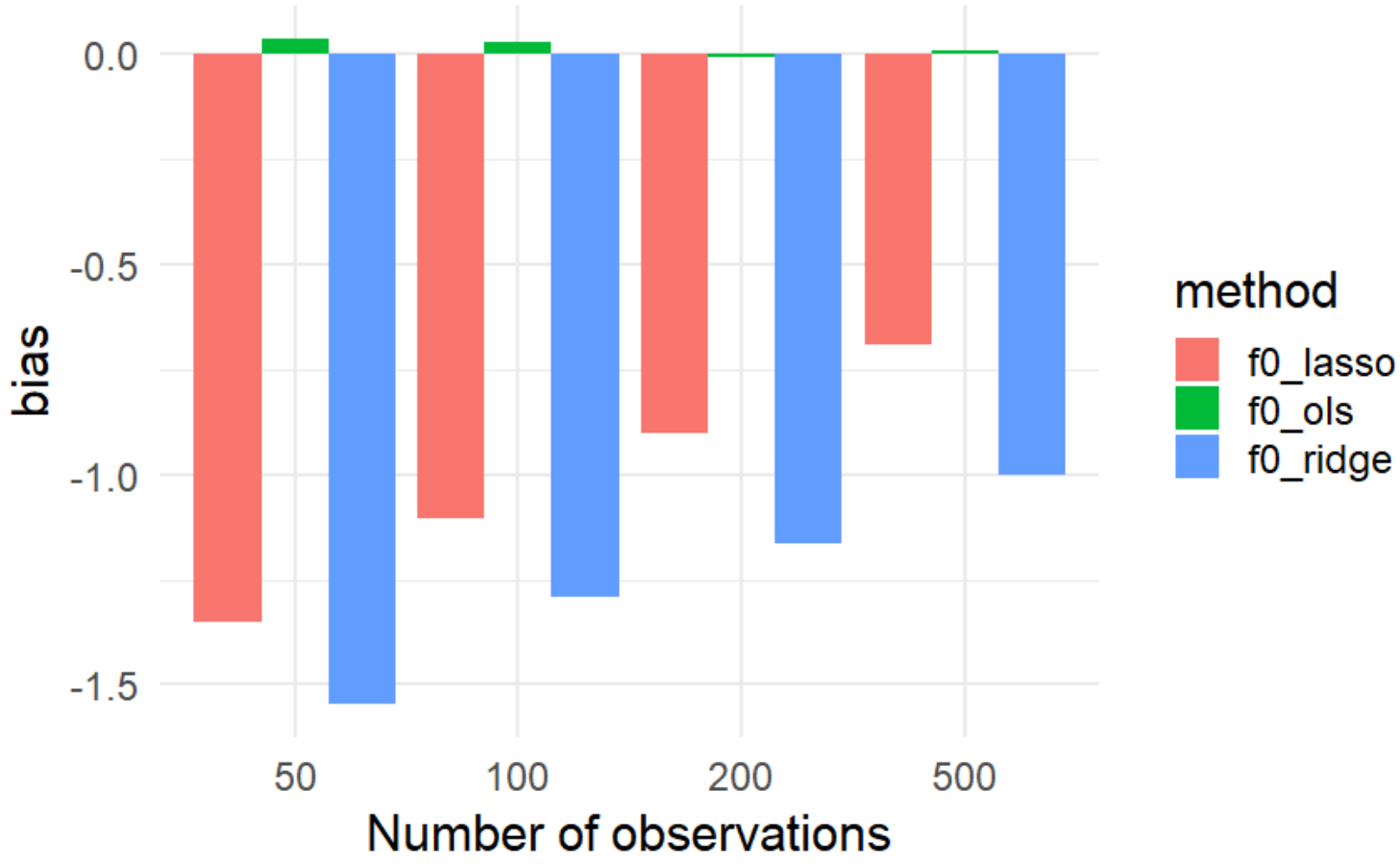
5. Number of observations

How do bias and variance depend on the number of observations?

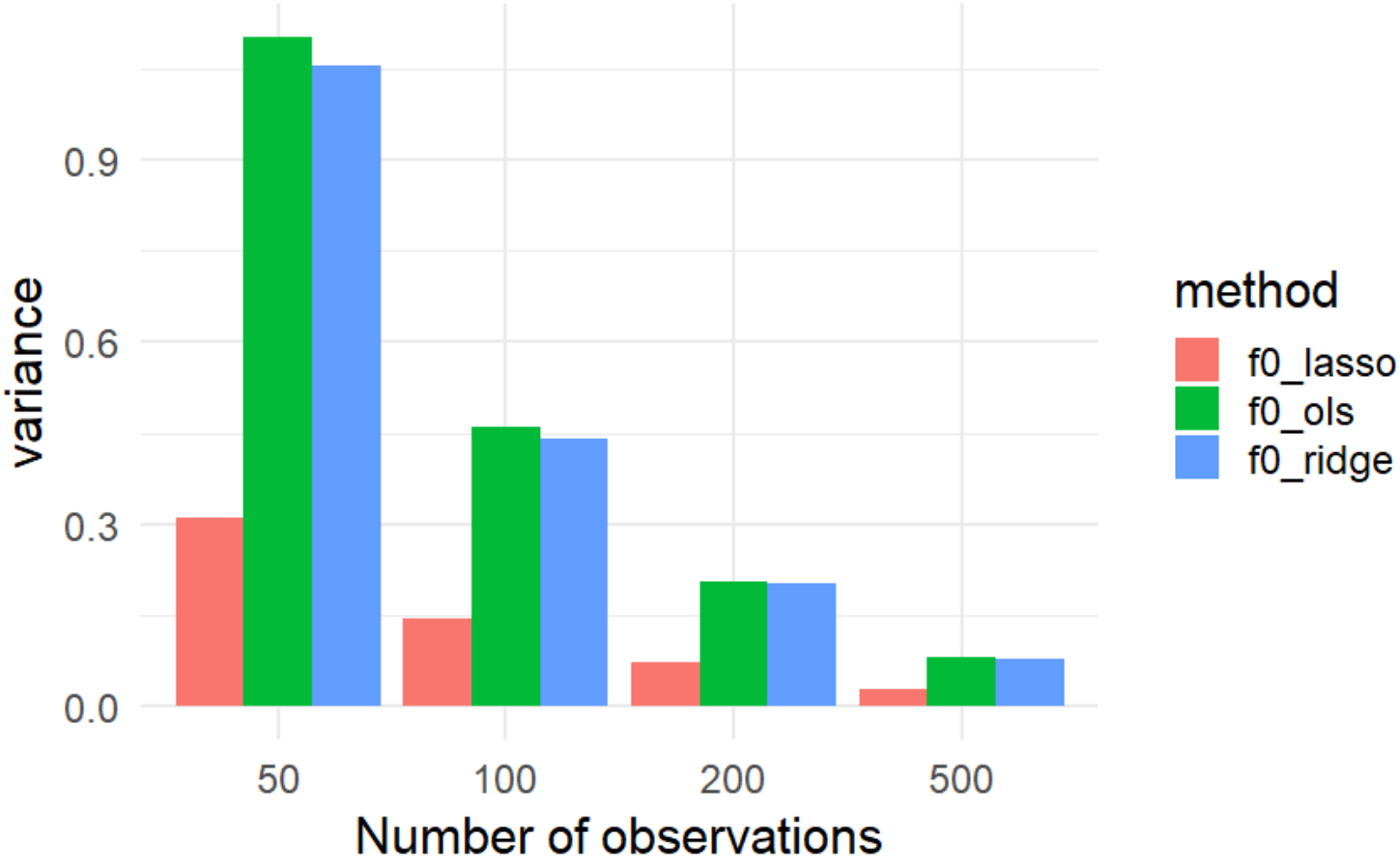
5. Number of observations

```
nObservations = c(50, 100, 200, 500)

dfAll = tibble()
for (i in 1:length(nObservations)) {

  temp = simResults(nsims = 1000, nobss = nObservations[i], ps = 10)

  dfAll = bind_rows(dfAll,
                    temp %>% mutate(numObservations = nObservations[i]))
}
```

5. Number of observations



5. Number of observations



6. Correlated regressors

How do results change when correlation between regressors is instead given by $\rho > 0$?

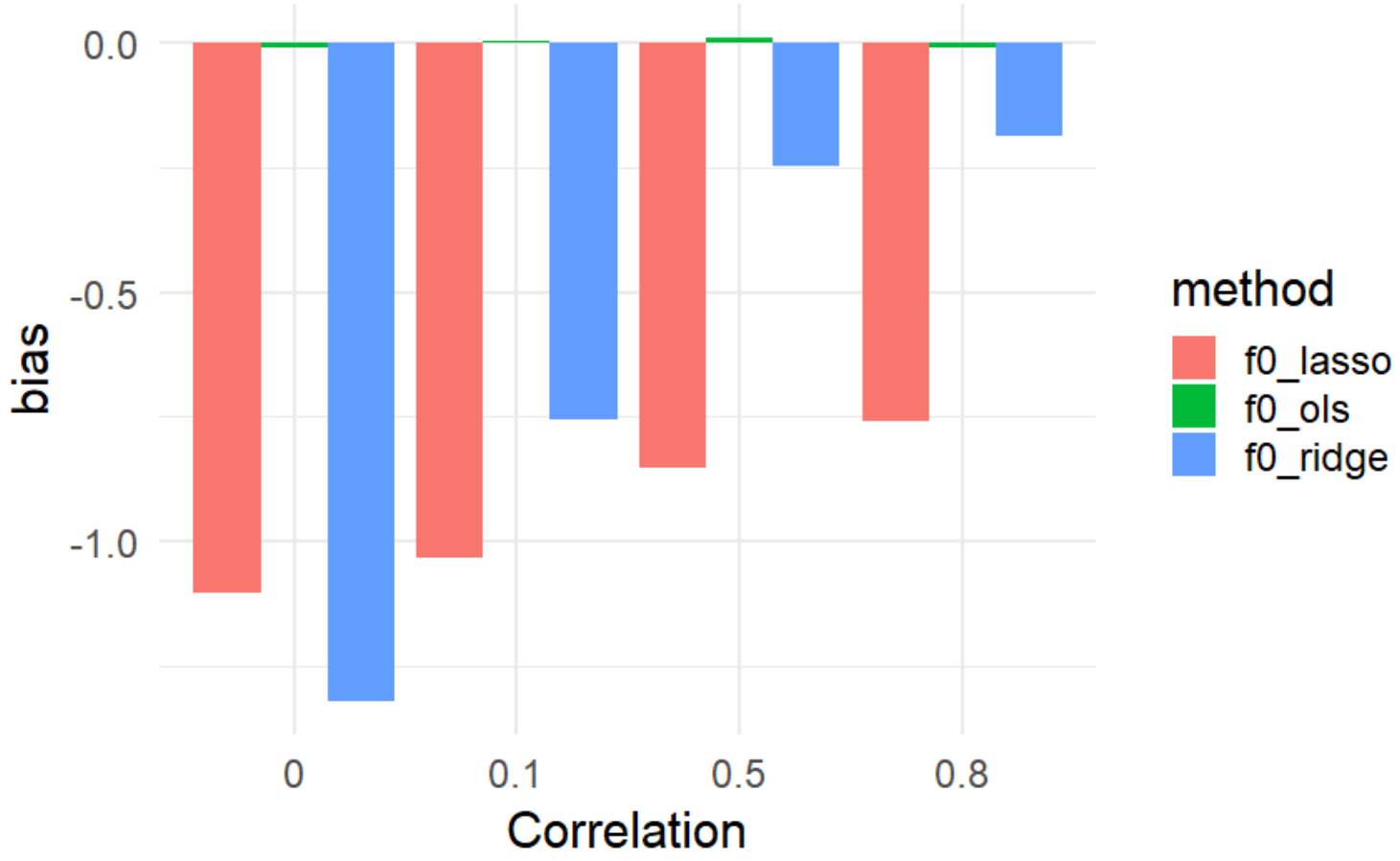
6. Correlated regressors

```
simResultsCorr = function(nsims, nobss, ps, rhos){

nsim = nsims  # Number of simulations
nobs = nobss   # Number of observations in the simulated dataset
p = ps      # Number of regressors with coefficient 0
beta = c(2,3, rep(0,p-2))  # Coefficient vector
                           # (2 for beta1, 3 for beta2,
                           # 0s for remaining betas)

rho    # Assumed regressor correlation
sigma = matrix(rho, nrow = p, ncol = p)
diag(sigma) = 1

...

 X = MASS::mvrnorm(nobs, #Randomly drawn X variables
                           mu = rep(0, p),
                           Sigma = sigma
                           )
...
}
```

6. Correlated regressors

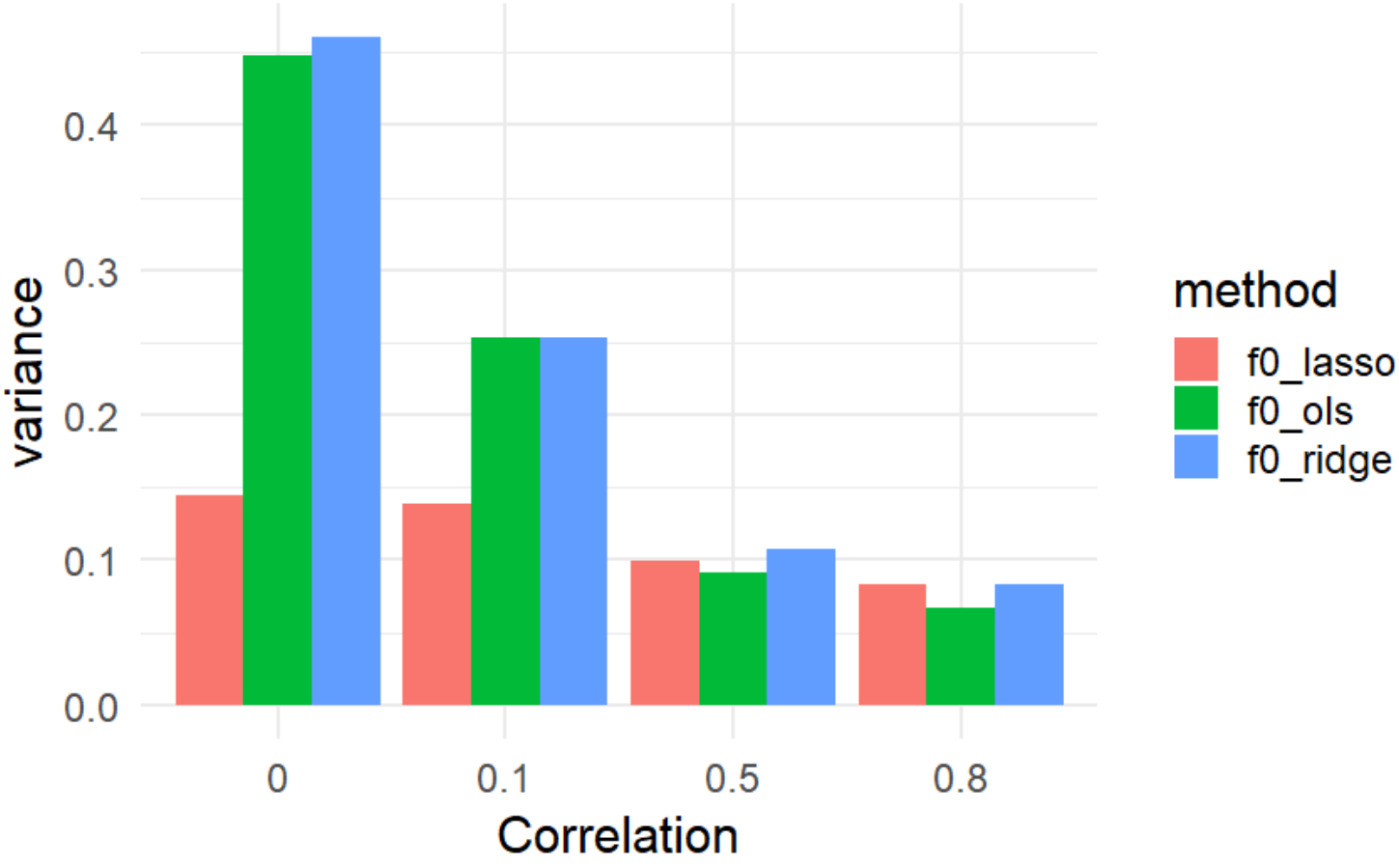
```
corrRho = c(0, .1, .5, .8)

dfAll = tibble()
for (i in 1:length(corrRho)) {

  temp = simResultsCorr(nsims = 1000, nobss = 100, ps = 10,
                        rhos = corrRho[i])

  dfAll = bind_rows(dfAll,
                    temp %>% mutate(Correlation = corrRho[i]))
}
```

6. Correlated regressors



- Variable selection (shrinkage) does not matter as much if relevant regressors are correlated with irrelevant regressors

6. Correlated regressors



Questions?