Coding Part

Bohao Tang

April 1, 2018

iv

Two distribution is

$$Y = 1 + 2X + \epsilon$$

and

$$Y = 1 + 2X^2 + \epsilon$$

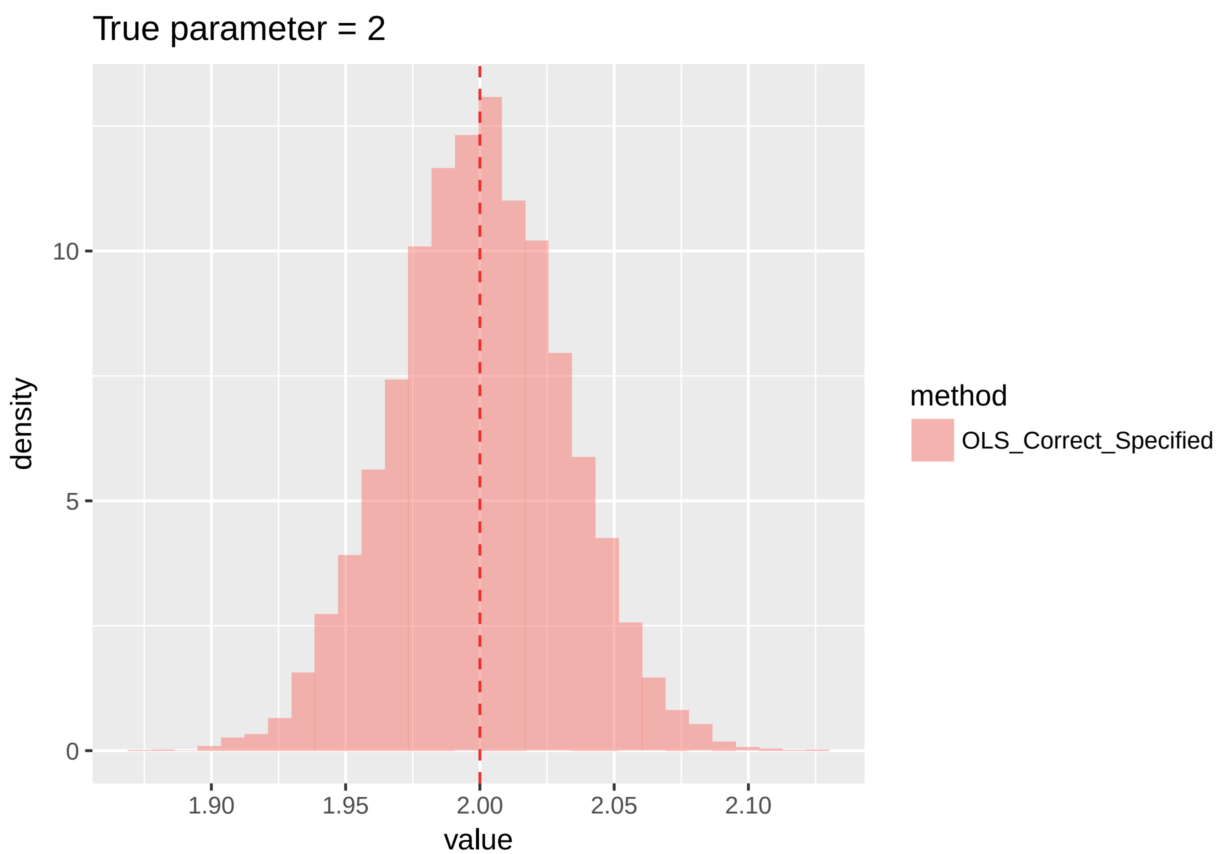
Where X, ϵ mutual independent and $\epsilon, X \sim N(0, 1)$.

Then the regression $E(Y|X) = \beta_0 + \beta_1 X$ correctly specifies first distribution and misspecifies the second.

```
model1 = function(X, eps){
    1 + 2*X + eps
}
model2 = function(X, eps){
    1 + 2 * X^2 + eps
}
```

For the first situation, we run estimation 10000 times, each have 1000 samples and only focus on β_1. The red dashed line is the true β_1:

```
library(ggplot2)

beta1 = c()

n = 1000
for(i in 1:10000){
    X = rnorm(n, 0, 1)
    eps = rnorm(n, 0, 1)

    Y = model1(X, eps)

    estimator = cov(X, Y) / var(X)

    beta1 = c(beta1, estimator)
}

data = data.frame(value = beta1)
data$method = "OLS_Correct_Specified"

ggplot(data, aes(value, fill = method)) +
    geom_histogram(alpha = 0.5, aes(y = ..density..), position = 'identity') +
    geom_vline(xintercept = 2, linetype="dashed", color="firebrick2") +
    ggtitle("True parameter = 2")
```



Then the bias and variance for $\hat{\beta}_1$ is:

```
bias = mean(beta1) - 2
bias
```

```
## [1] -9.523031e-05
```

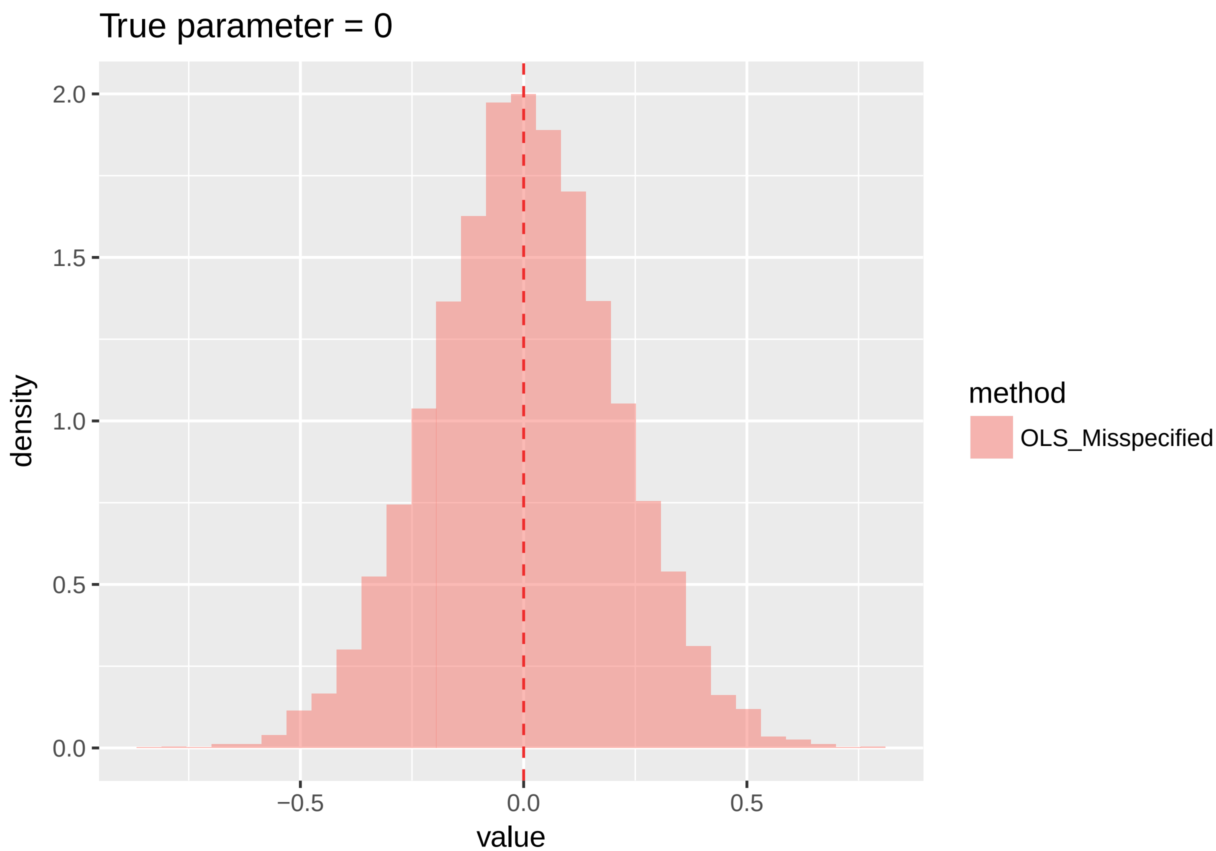
```
variance = var(beta1)
variance
```

```
## [1] 0.0009997933
```

In the second situation, now true β_1 doesn't exit, we need to compare the result to $\beta_1^* = \frac{Cov(X,Y)}{Var(X)} = 0$. We do the same thing as above:

```
beta1mis = c()

n = 1000
for(i in 1:10000){
    X = rnorm(n, 0, 1)
    eps = rnorm(n, 0, 1)

    Y = model2(X, eps)

    estimator = cov(X, Y) / var(X)

    beta1mis = c(beta1mis, estimator)
}
```

```r
datamis = data.frame(value = beta1mis)
datamis$method = "OLS_Misspecified"

ggplot(datamis, aes(value, fill = method)) +
    geom_histogram(alpha = 0.5, aes(y = ..density..), position = 'identity') +
    geom_vline(xintercept = 0, linetype="dashed", color="firebrick2") +
    ggtitle("True parameter = 0")
```



Then the bias and variance for $\hat{\beta}_1^*$ is:

```r
bias = mean(beta1mis) - 0
bias
```

```
## [1] 0.00111693
```

```r
variance = var(beta1mis)
variance
```

```
## [1] 0.04135902
```

v

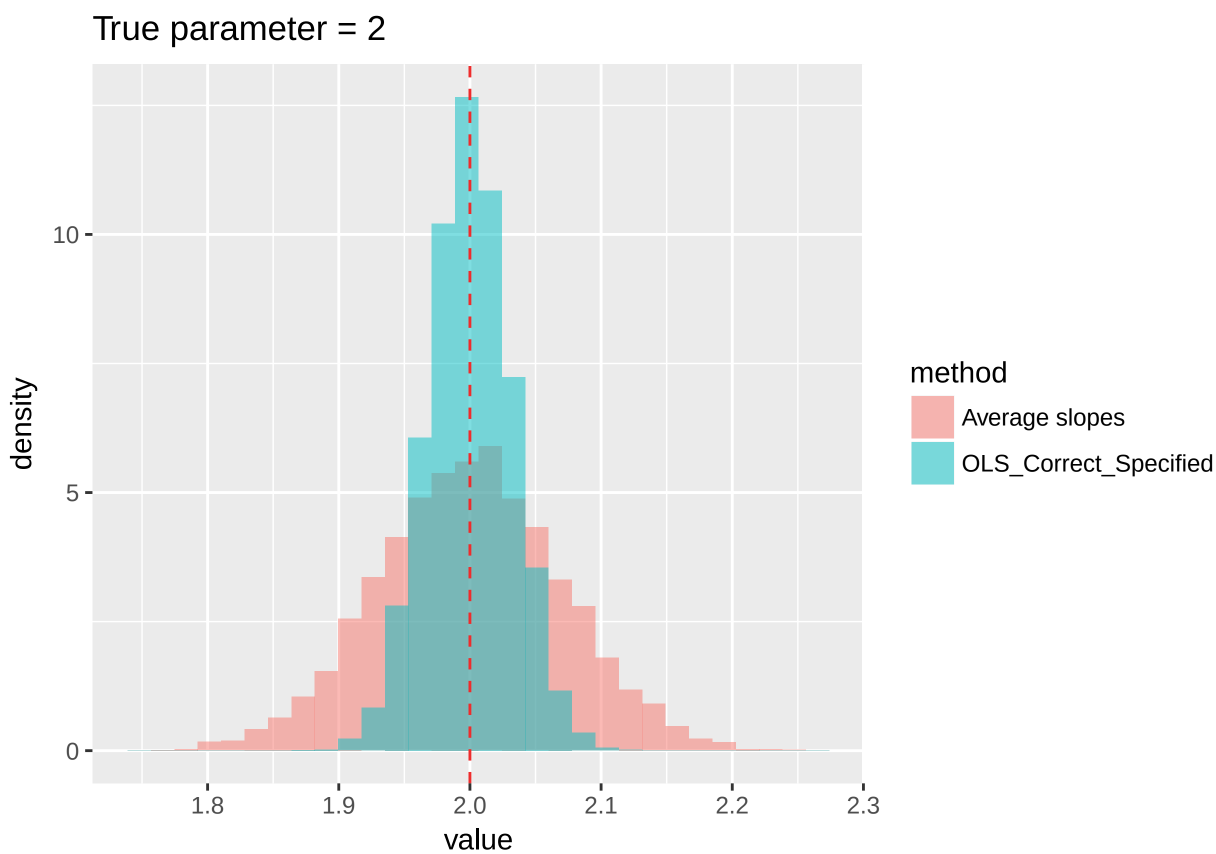
We do the model:

$$Y = 1 + 2X + \epsilon$$

where X, ϵ mutual independent and $\epsilon, X \sim N(0, 1)$.

And the second estimator mentioned in homework solution, which is just a median of slopes. We use data from `iv` for OLS estimator and for second estimator:

```
slope = c()

n = 1000
index = 1:1000
oddi = index[index%%2 == 1]
eveni = index[index%%2 == 0]
for(i in 1:10000){
    X = rnorm(n, 0, 1)
    eps = rnorm(n, 0, 1)

    Y = model1(X, eps)

    estimator =
        median( (Y[eveni] - Y[oddi]) / (X[eveni] - X[oddi]) )

    slope = c(slope, estimator)
}

slopes = data.frame(value = slope)
slopes$method = "Average slopes"

estimators = rbind(data, slopes)

ggplot(estimators, aes(value, fill = method)) +
    geom_histogram(alpha = 0.5, aes(y = ..density..), position = 'identity') +
    geom_vline(xintercept = 2, linetype="dashed", color="firebrick2") +
    ggtitle("True parameter = 2")
```



Therefore we can see OLS is a better estimator for β_1.