In class questions

1. Option x

We can set option x=TRUE or not

```r
library(faraway)
set.seed(123)
n = 10
beta = 1
eps = rnorm(n)
X = rnorm(n)
Y = 1 + X*beta + eps
lmod <- lm(Y ~ X, x=TRUE)
lmod_f <- lm(Y ~ X)
```

The output of two linear models are exactly the same.

```r
summary(lmod)$coefficients
```

```
##             Estimate Std. Error  t value    Pr(>|t|)
## (Intercept) 0.963907  0.2669399 3.610951 0.006872187
## X           1.530714  0.2651745 5.772479 0.000418096
```

```r
summary(lmod_f)$coefficients
```

```
##             Estimate Std. Error  t value    Pr(>|t|)
## (Intercept) 0.963907  0.2669399 3.610951 0.006872187
## X           1.530714  0.2651745 5.772479 0.000418096
```

However, one can return design matrix x, but another does not.

```r
lmod_f$x #no return of design matrix x
```

```
## named list()
```

```r
lmod$x #returns design matrix x
```

```
##    (Intercept)          X
## 1            1  1.2240818
## 2            1  0.3598138
## 3            1  0.4007715
## 4            1  0.1106827
## 5            1 -0.5558411
## 6            1  1.7869131
## 7            1  0.4978505
## 8            1 -1.9666172
## 9            1  0.7013559
## 10           1 -0.4727914
## attr(,"assign")
## [1] 0 1
```

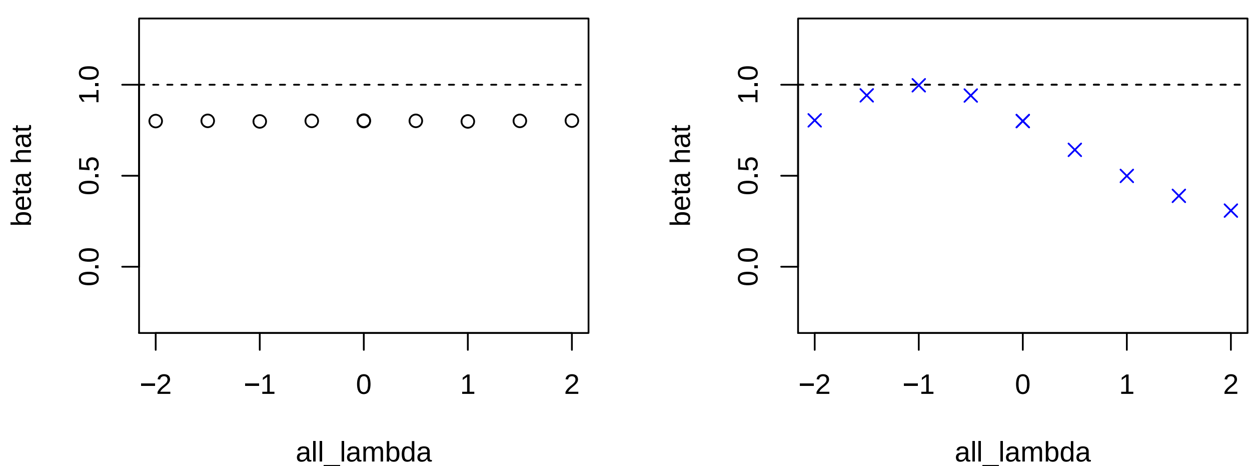
2. Fitting model with added errors

```r
set.seed(123)
N_rep = 300
beta = 1
all_lambda = seq(0, 2, 0.5)
num_lambda = length(all_lambda)
beta_OLS_estimates = matrix(NA, N_rep, num_lambda)  #Direct regression with X
beta_PLUS_estimates = matrix(NA, N_rep, num_lambda) #Use X+lambda*E, generate E
beta_MINUS_estimates = matrix(NA, N_rep, num_lambda) #Use X-lambda*E, generate E
beta_oracle_PLUS_estimates = matrix(NA, N_rep, num_lambda) #Use X+lambda*tau, true tau
beta_oracle_MINUS_estimates = matrix(NA, N_rep, num_lambda) #Use X-lambda*tau, true tau

for(k_ind in 1:num_lambda){
  lambda_val = all_lambda[k_ind]

  for(ind_rep in 1:N_rep){
```

```r
n = 200
sigma_v = 0.5
Z = rnorm(n)
eps = rnorm(n, sd = sigma_v)
tau = rnorm(n, sd = sigma_v)
X = Z + tau
Y = Z*beta+eps #consider a model without intercept

#0. Direct calculatoin
lmod <- lm(Y ~ X - 1)
beta_OLS_estimates[ind_rep, k_ind] <- lmod$coefficients[1]

#1. Plus generate E
E_generate <- rnorm( n, mean = 0, sd = sigma_v )
X1 = X + lambda_val * E_generate
lmod1 <- lm(Y ~ X1 - 1)
beta_PLUS_estimates[ind_rep, k_ind] <- lmod1$coefficients

#2. Plus generate E
X2 = X - lambda_val * E_generate
lmod2 <- lm(Y ~ X2 - 1)
beta_MINUS_estimates[ind_rep, k_ind] <- lmod2$coefficients

#3. Plus true tau
X3 = X + lambda_val * tau
lmod3 <- lm(Y ~ X3 - 1)
beta_oracle_PLUS_estimates[ind_rep, k_ind] <- lmod3$coefficients

#4. Minus true tau
X4 = X - lambda_val * tau
lmod4 <- lm(Y ~ X4 - 1)
beta_oracle_MINUS_estimates[ind_rep, k_ind] <- lmod4$coefficients
```

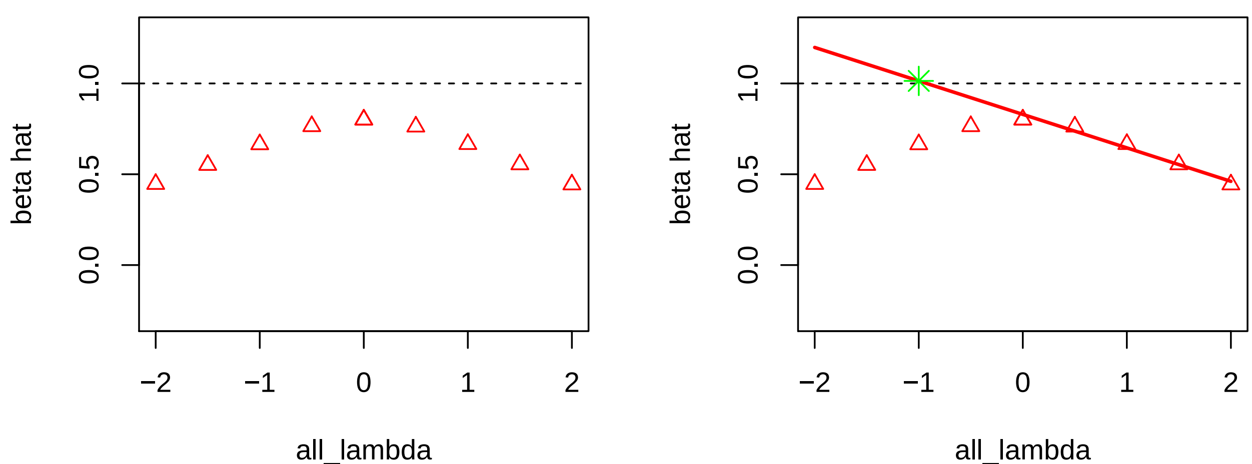
```
  }
}
```

```
par(mfrow=c(1,2))
plot(all_lambda, colMeans(beta_OLS_estimates), ylim = c(-0.3,1.3),
     xlim=c(-2,2), ylab="beta hat")
points(sort(-all_lambda), colMeans(beta_OLS_estimates))
abline(a=beta, b = 0,  lty = "dashed") #true beta value

plot(all_lambda, colMeans(beta_OLS_estimates), type="n",
     ylim = c(-0.3,1.3), xlim=c(-2,2), ylab="beta hat")
abline(a=beta, b = 0,  lty = "dashed") #true beta value

points(all_lambda, colMeans(beta_oracle_PLUS_estimates), col="blue", pch=4)
points((-all_lambda), colMeans(beta_oracle_MINUS_estimates), col="blue", pch=4)
abline(a=beta, b = 0,  lty = "dashed") #true beta value
```



- Dashed line: true $\beta = 1$.
- Left panel, estimated $\hat{\beta} = (X^\top X)^{-1} X^\top Y$ using observed X. (Bias)
- Right panel, estimated β using $X + \lambda * \tau$, where τ is true errors.
  - Note true $\beta = 1$ is achieved by correcting τ error at $\lambda = -1$.
- However, in practice, we cannot know τ. We can only create an error E by ourself.

```r
par(mfrow=c(1,2))
plot(all_lambda, colMeans(beta_OLS_estimates), type="n",
     ylim = c(-0.3,1.3), xlim=c(-2,2), ylab="beta hat")
abline(a=beta, b = 0,  lty = "dashed") #true beta value

points(all_lambda, colMeans(beta_PLUS_estimates), col="red", pch=2)
points((-all_lambda), colMeans(beta_MINUS_estimates), col="red", pch=2)

plot(all_lambda, colMeans(beta_OLS_estimates), type="n",
     ylim = c(-0.3,1.3), xlim=c(-2,2), ylab="beta hat")
abline(a=beta, b = 0,  lty = "dashed") #true beta value

points(all_lambda, colMeans(beta_PLUS_estimates), col="red", pch=2)
points((-all_lambda), colMeans(beta_MINUS_estimates), col="red", pch=2)
lo <- lm(colMeans(beta_PLUS_estimates) ~ all_lambda )
all_lambda_pos_neg <- seq(-2,2,0.5)
predict_covariates <- data.frame(all_lambda = all_lambda_pos_neg)
predict_values <- predict(lo,predict_covariates)
lines(all_lambda_pos_neg, predict_values, col='red', lwd=2)
points( -1,  predict_values[3], col='green', pch=8, cex = 1.6 )
```



- The above plots are obtained by adding $\lambda * E$ where we allow λ to positive and negatve.
- We can see that positive and negative λ yield the symmetric estimates of β. (Symmetric

around 0.)

- But if we fitted a smooth line and get an estimate value at $\lambda = -1$ (green point). We can see that a value close to true β is estimated.