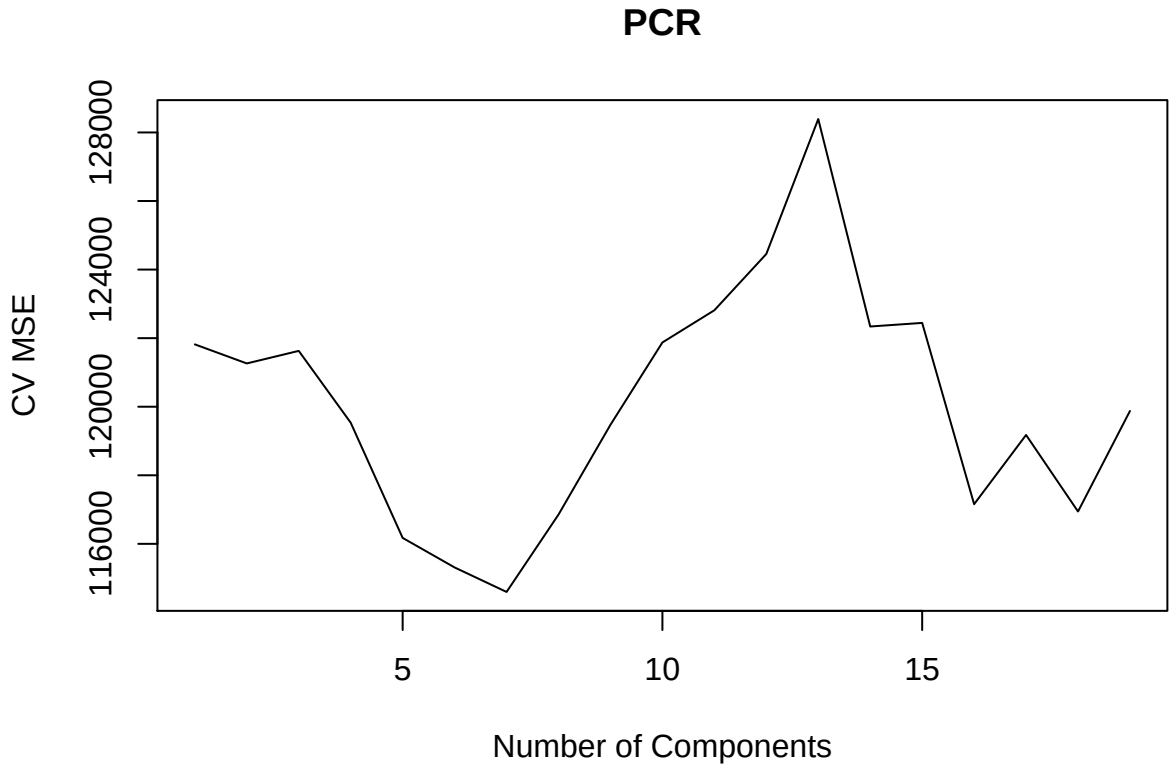
R Notebook

```
library(ISLR)
library(glmnet)
library(caret)
library(tidyverse)
library(pls)
```

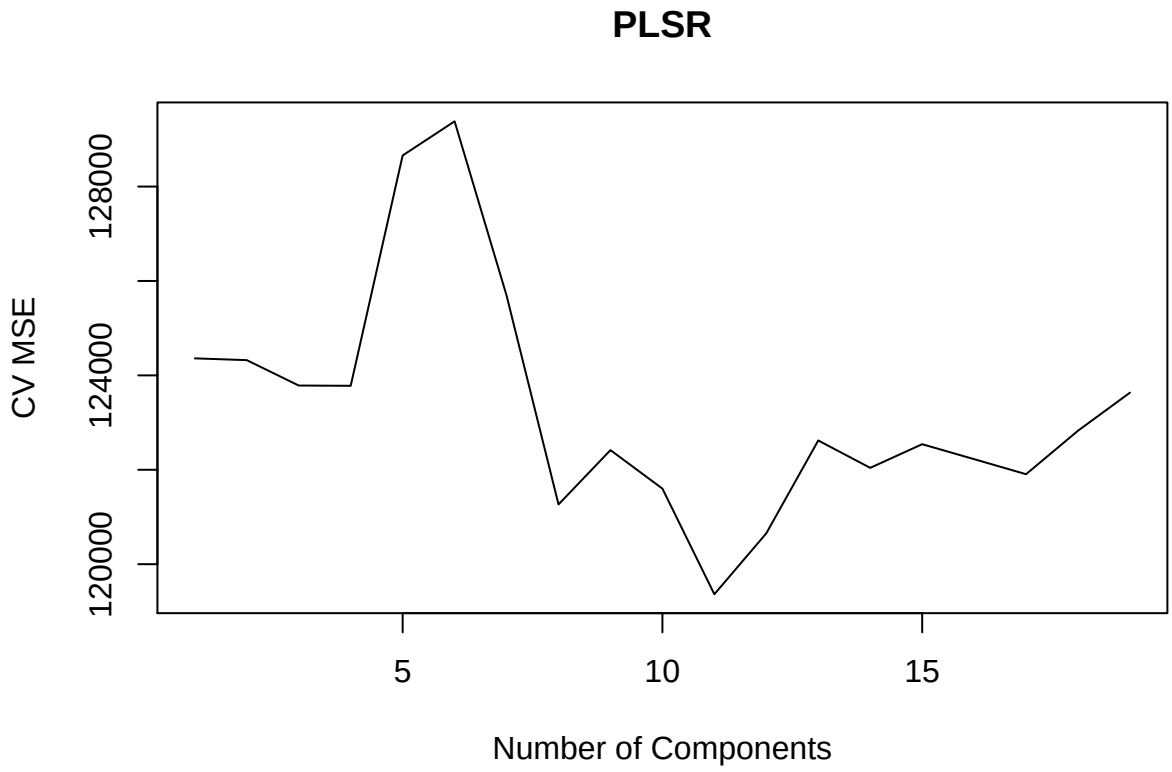
```
Hitters <- na.omit(Hitters)
y <- Hitters$Salary
x <- Hitters %>% select(-Salary) %>% data.matrix()
```

Cross-validation for pcr() and plsr()

```
n <- nrow(Hitters)
set.seed(100)
pcr_fit <- pcr(Salary ~ ., data = Hitters, scale = TRUE,
validation = "CV", segments=10)
plot(pcr_fit$validation$PRESS[1, ] / n, type="l", main="PCR",
xlab="Number of Components", ylab="CV MSE")
```




```
set.seed(200)
plsr_fit <- plsr(Salary ~ ., data = Hitters, scale = TRUE,
validation = "CV", segments=10)
plot(plsr_fit$validation$PRESS[1, ] / n , type="l", main="PLSR",
xlab="Number of Components", ylab="CV MSE")
```




```
which.min(pcr_fit$validation$PRESS[1, ] / n)
```

```
## 7 comps
##       7
```

```
which.min(plsr_fit$validation$PRESS[1, ] / n)
```

```
## 11 comps
##       11
```

I will use 7 and 11 components for PCR and PLSR(the number of components that gives lowest CV MSE)

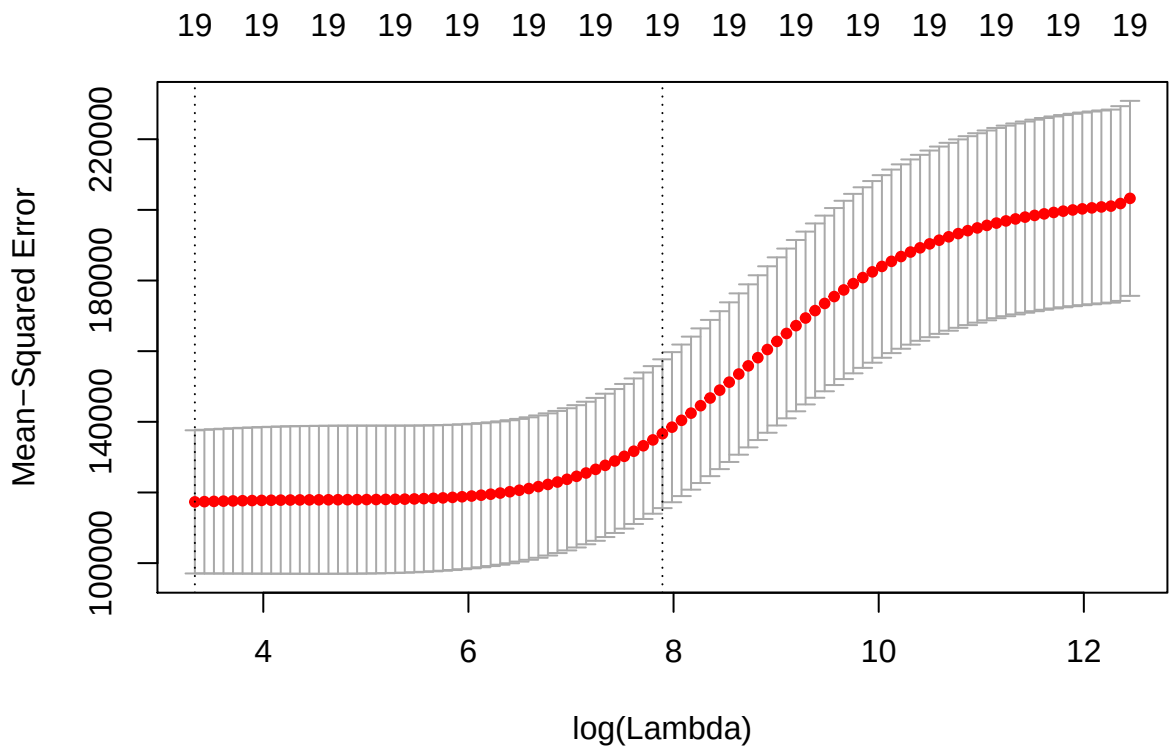
Cross-validation for ridge regression and lasso

Ridge: a is 0 Lasso: a is 1

```
set.seed(300)
# code for ridge regression CV
ridge.mod <- cv.glmnet(x , y, alpha = 0, nfolds = 10)
ridge.mod$lambda.min
```

```
## [1] 28.01718
```

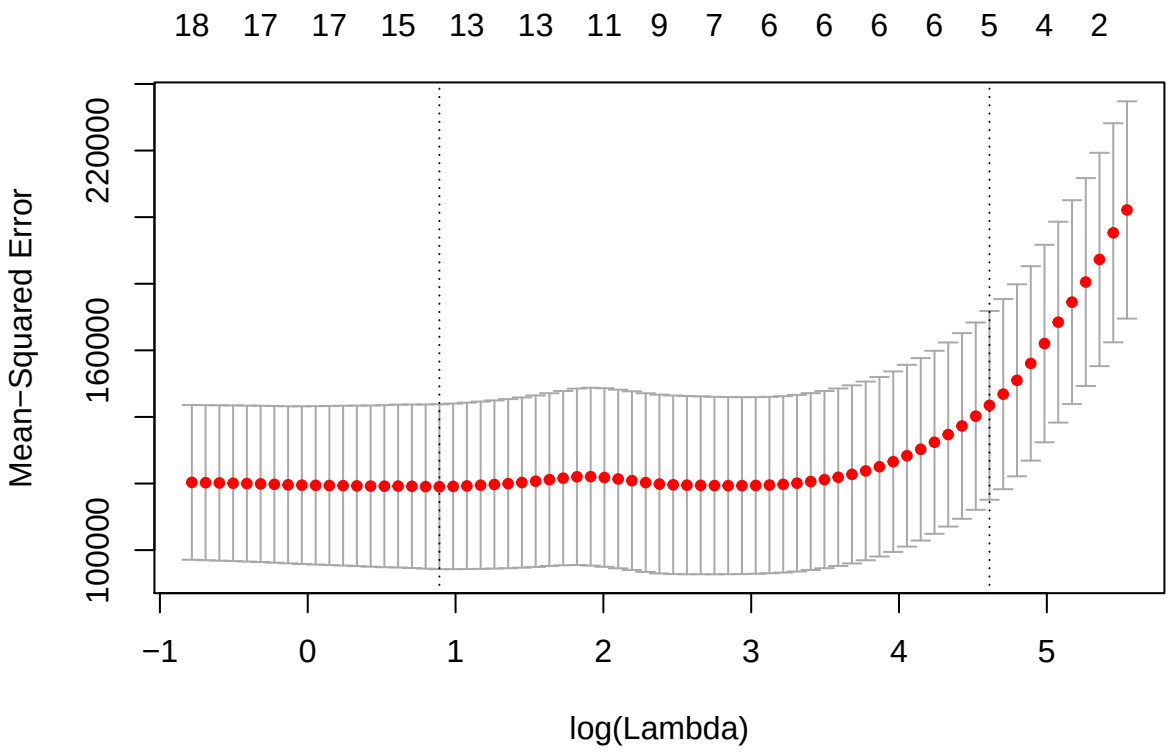
```
plot.cv.glmnet(ridge.mod)
```




```
set.seed(400)
# code for lasso CV
lasso.mod <- cv.glmnet(x , y, alpha = 1, nfolds = 10)
lasso.mod$lambda.min
```

```
## [1] 2.436791
```

```
plot.cv.glmnet(lasso.mod)
```



Nested Cross Validation

```
set.seed(9)
lm.cv <- numeric()
ridge.cv <- numeric()
lasso.cv <- numeric()
plsr.cv <- numeric()
pcr.cv <- numeric()
folds <- createFolds(Hitters$Salary, 10)
```

```
set.seed(9)
for (i in 1:10) {
  train <- Hitters[-folds[[i]],]
  train.x <- train %>% select(-Salary) %>% data.matrix()
  test <- Hitters[folds[[i]],]
  test.x <- test %>% select(-Salary) %>% data.matrix()
  ## basic OLS
  lm.mod <- lm(Salary~., data = train)
  lm.pred <- predict(lm.mod, newdata = test)
  lm.cv[i] <- mean((lm.pred - test$Salary)^2)
  ## ridge
  ridge.mod <- cv.glmnet(train.x, train$Salary, alpha = 0, nfolds = 10)
  ridge.pred <- predict(ridge.mod, s = "lambda.mi", newx = test.x)
  ridge.cv[i] <- mean((ridge.pred - test$Salary)^2)
  ## lasso
  lasso.mod <- cv.glmnet(train.x, train$Salary, alpha = 1, nfolds = 10)
  lasso.pred <- predict(lasso.mod, s = "lambda.mi", newx = test.x)
  lasso.cv[i] <- mean((lasso.pred - test$Salary)^2)
```

```
  ## pcr
  pcr.fit <- pcr(Salary ~ ., data = train, scale = TRUE, validation = "CV", segments=10)
  pcr.ncomp <- which.min(pcr_fit$validation$PRESS[1, ] / n)
  pcr.pred <- predict(pcr.fit, ncomp = pcr.ncomp ,newdata = test)
  pcr.cv[i] <- mean((pcr.pred - test$Salary)^2)
  ## plsr
  plsr.fit <- plsr(Salary ~ ., data = train, scale = TRUE, validation = "CV", segments=10)
  plsr.ncomp <- which.min(plsr_fit$validation$PRESS[1, ] / n)
  plsr.pred <- predict(plsr.fit, ncomp = plsr.ncomp ,newdata = test)
  plsr.cv[i] <- mean((plsr.pred - test$Salary)^2)
}
```

```
print(c(min(lm.cv), min(ridge.cv), min(lasso.cv), min(pcr.cv), min(plsr.cv)))
```

```
## [1] 75574.35 67916.45 70006.41 72139.52 71379.68
```

```
print(c(mean(lm.cv), mean(ridge.cv), mean(lasso.cv), mean(pcr.cv), mean(plsr.cv)))
```

```
## [1] 115946.0 118982.1 112901.3 119301.0 113907.3
```