# STATS 305A_HW3

Muhammad Ahmed Chaudhry

11/19/2021

First we load the packages as needed:

## Problem 3.3

First we read in the data:

```r
lprostate_data <- read.delim("../Data/lprostate.dat", sep = "\t")

# removing extraneous row.names column
lprostate_data <- lprostate_data[,-1]
```

a) Next we implement the `predictKRR` function. First, we write a helper function which calculates the Gaussian (RBF) kernel between two vectors.

```r
rbf_kernel <- function(x = NULL, z = NULL, bandwidth){
  # parameters:
  # vectors x and z
  # bandwidth tau of the Gaussian kernel, a real number > 0

  diff <- x - z
  sum_squares <- sum(diff^2)
  kernel <- exp((-1 * sum_squares) / (2*(bandwidth^2)))
  return(kernel)
}
```

Next we implement the predictKRR function:

```r
predictKRR <- function(X = NULL, Z = NULL, alpha = NULL,
                       tau = 1, offset = 1){
  # parameters:
  # X, a d x n data matrix with training data with rows x_i^T
  # z, a d x m matrix of m data points to make new predictions on
  # alpha, an n vector \hat{alpha} which is the fitted parameters from 3.2b)
  # tau, the bandwidth of the Gaussian kernel, a real number > 0
  # offset, an offset parameter b which is a real number

  y_hat <- c()
  for (j in 1:dim(Z)[2]){
    sum <- offset
    for (i in 1:dim(X)[2]){
      sum <- sum + rbf_kernel(X[,i], Z[,j], tau) * alpha[i]
    }
    y_hat <- append(y_hat, sum)
  }
  return(y_hat)

}
```

b) Next we implement the fitKRR function. First, we write a helper function that builds our Gram Matrix:

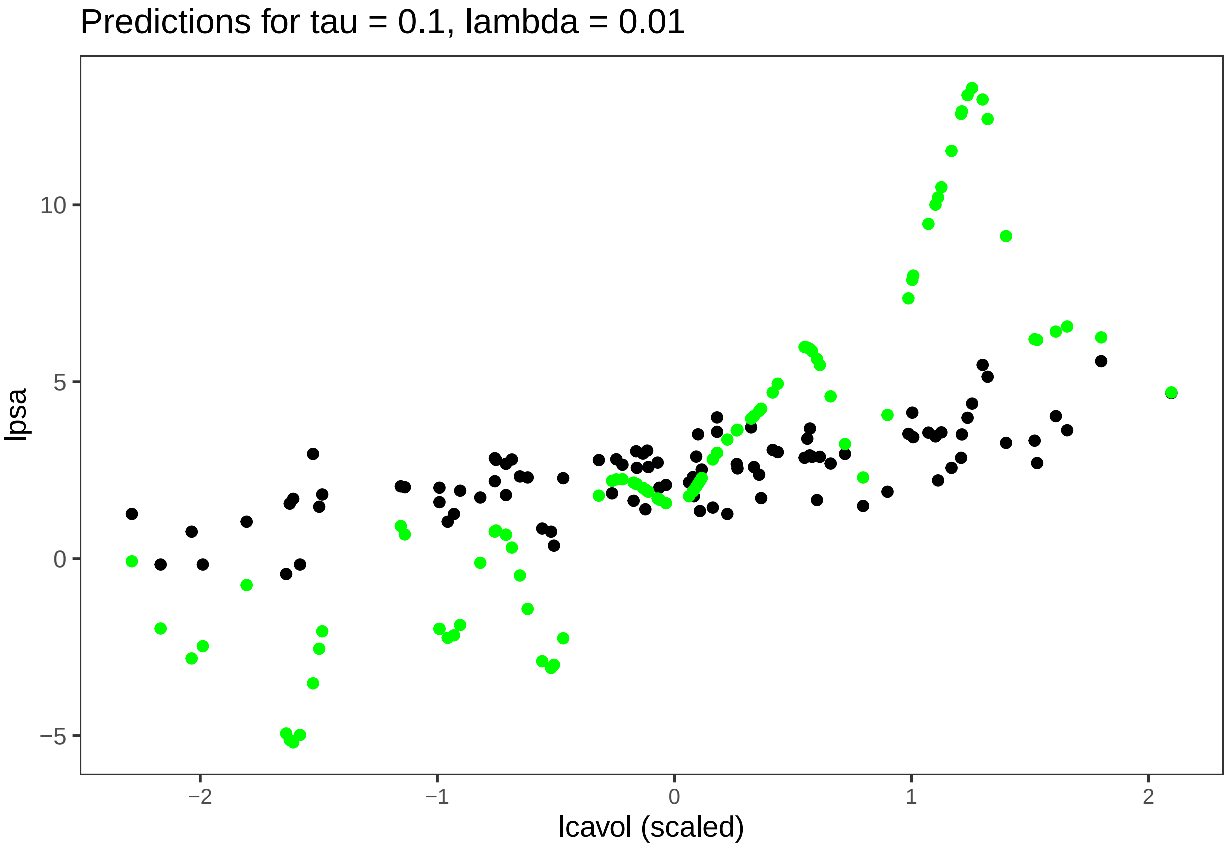
```
gram_matrix <- function(X = NULL, bandwidth){
  # parameters:
  # X, a d x n data matrix with training data with rows x_i^T
  # bandwidth, of the Gaussian kernel, a real number > 0
  G <- matrix(data = NA, nrow = dim(X)[2], ncol = dim(X)[2])
  for (j in 1:dim(X)[2]){
    for (i in 1:dim(X)[2]){
      G[i,j] <- rbf_kernel(X[,i], X[,j], bandwidth = bandwidth)
    }
  }
  return(G)
}
```

Next we write the fitKRR function:

```
fitKRR <- function(X = NULL, y = NULL,
                   lambda = 1, tau = 1){
  # parameters:
  # X, a d x n data matrix with training data with rows x_i^T
  # y, an n vector of responses
  # lambda, the regularization or shrinkage parameter
  # tau, the bandwidth of the Gaussian kernel, a real number > 0

  # getting alpha_hat
  y_centered <- y - mean(y)
  G <- gram_matrix(X, tau)
  inv_arg <- t(G) %*% G + lambda * G
  inv <- solve(inv_arg)
  GT_Y <- t(G) %*% y_centered
  alpha_hat <- inv %*% GT_Y

  # getting yMean
  y_hat_centered <- G %*% alpha_hat
  yMean <- mean(y_hat_centered)

  return(list(alpha_hat, yMean))
}
```

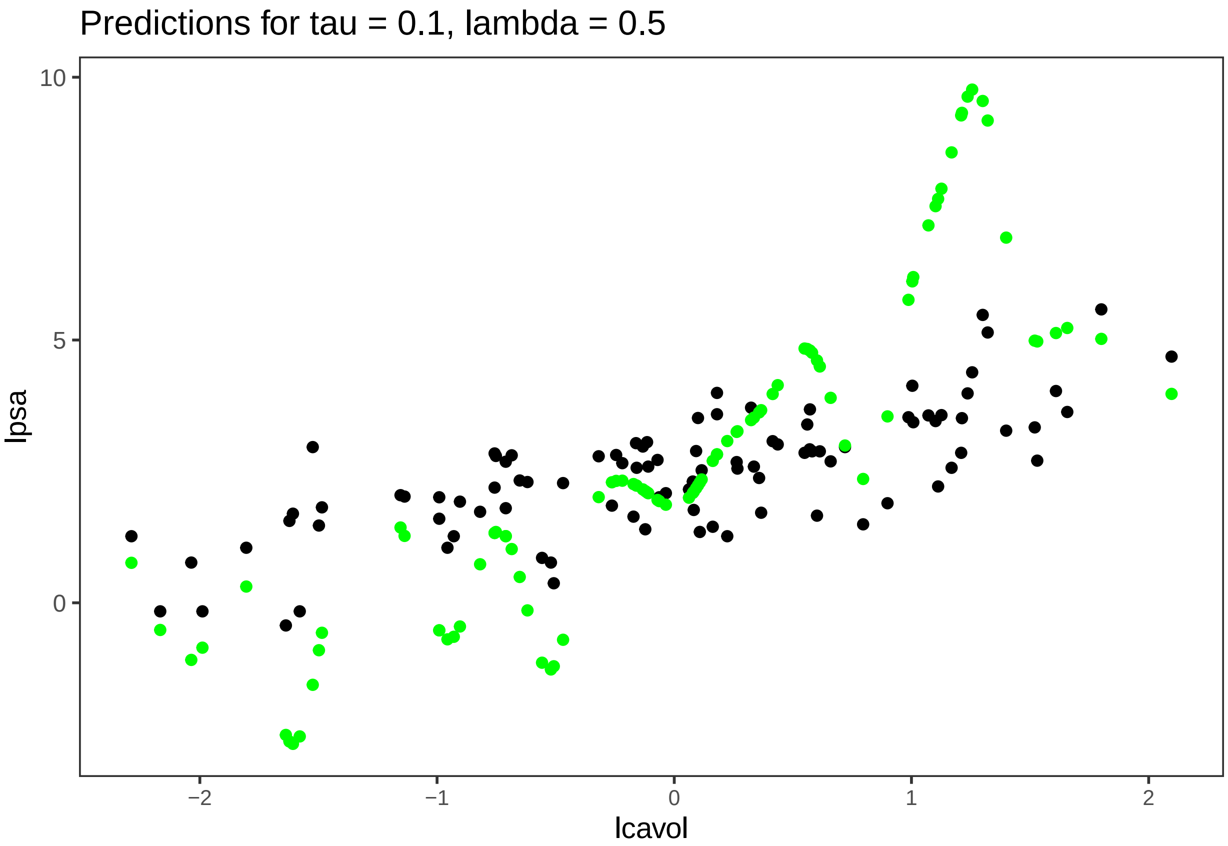
c) Next, we perform kernel ridge regression using lpsa as the response and lcavol as the only covariate x, with $\tau = 0.1$ fixed:

```
y = as.matrix(lprostate_data[,"lpsa"])

X = as.matrix(lprostate_data[,"lcavol"])
X_scaled = scale(X)

alpha_hat_1 <- unlist(fitKRR(X = X_scaled, y = t(y), tau = 0.1, lambda = 0.01)[1])

y_hat_1 <- predictKRR(X = t(X_scaled), Z = t(X_scaled), alpha = alpha_hat_1,
                      tau = 0.1, offset = mean(y))

plot_data_1 <- as.data.frame(cbind(X_scaled, y, y_hat_1))

ggplot(plot_data_1,
       aes(x = V1, y = V2)) +
  geom_point(aes(x = V1, y = V2))+
  geom_point(aes(x = V1,
              y = y_hat_1), color = "green")+
  xlab("lcavol (scaled)") +
  ylab("lpsa") +
  theme_bw() +
  theme(axis.text.x=element_text(size=8),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())+
  ggtitle("Predictions for tau = 0.1, lambda = 0.01")
```



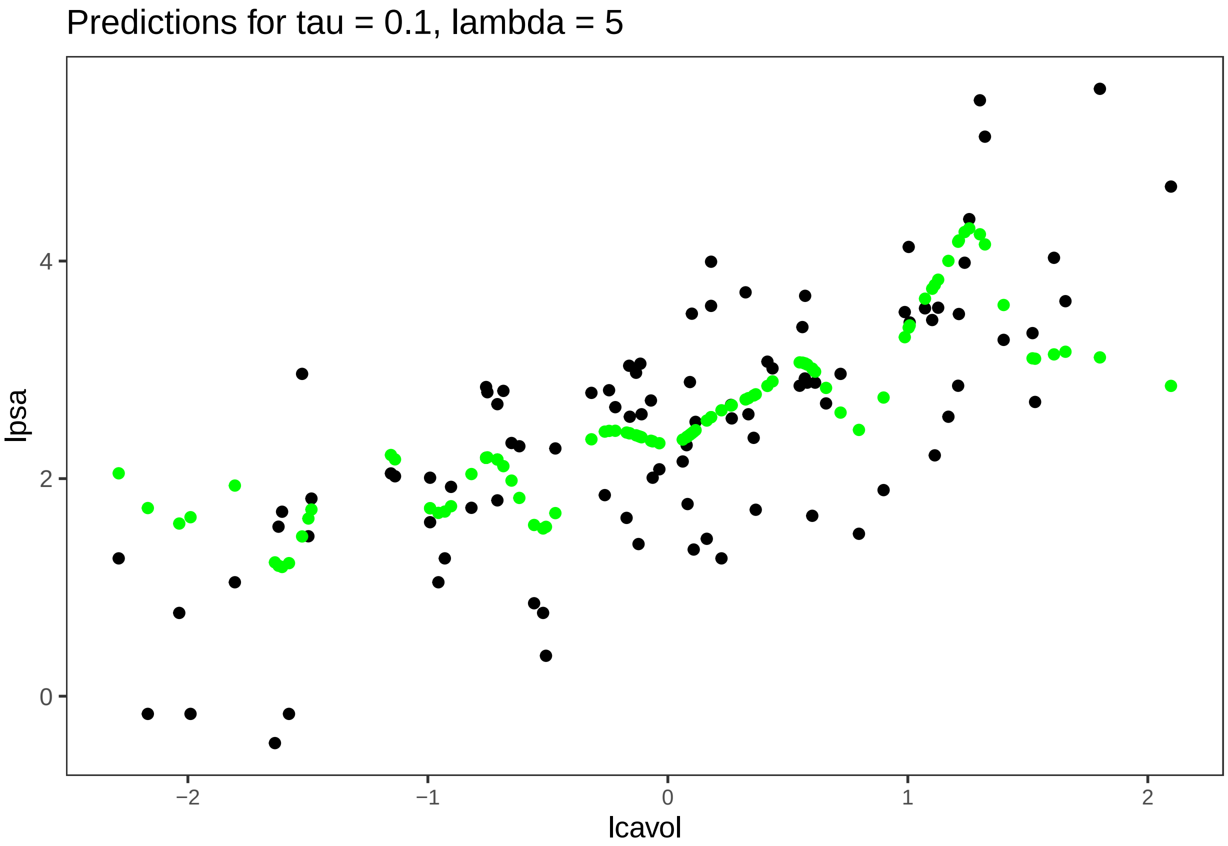
```
y = as.matrix(lprostate_data[,"lpsa"])

X = as.matrix(lprostate_data[,"lcavol"])
X_scaled = scale(X)

alpha_hat_1 <- unlist(fitKRR(X = X_scaled, y = t(y), tau = 0.1, lambda = 0.5)[1])

y_hat_1 <- predictKRR(X = t(X_scaled), Z = t(X_scaled), alpha = alpha_hat_1,
                      tau = 0.1, offset = mean(y))
```

```
plot_data_1 <- as.data.frame(cbind(X_scaled, y, y_hat_1))

ggplot(plot_data_1,
       aes(x = V1, y = V2)) +
  geom_point(aes(x = V1, y = V2))+
  geom_point(aes(x = V1,
              y = y_hat_1), color = "green")+
  xlab("lcavol") +
  ylab("lpsa") +
  theme_bw() +
  theme(axis.text.x=element_text(size=8),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())+
  ggtitle("Predictions for tau = 0.1, lambda = 0.5")
```



```
y = as.matrix(lprostate_data[,"lpsa"])

X = as.matrix(lprostate_data[,"lcavol"])
X_scaled = scale(X)

alpha_hat_1 <- unlist(fitKRR(X = X_scaled, y = t(y), tau = 0.1, lambda = 5)[1])

y_hat_1 <- predictKRR(X = t(X_scaled), Z = t(X_scaled), alpha = alpha_hat_1,
                      tau = 0.1, offset = mean(y))

plot_data_1 <- as.data.frame(cbind(X_scaled, y, y_hat_1))
```

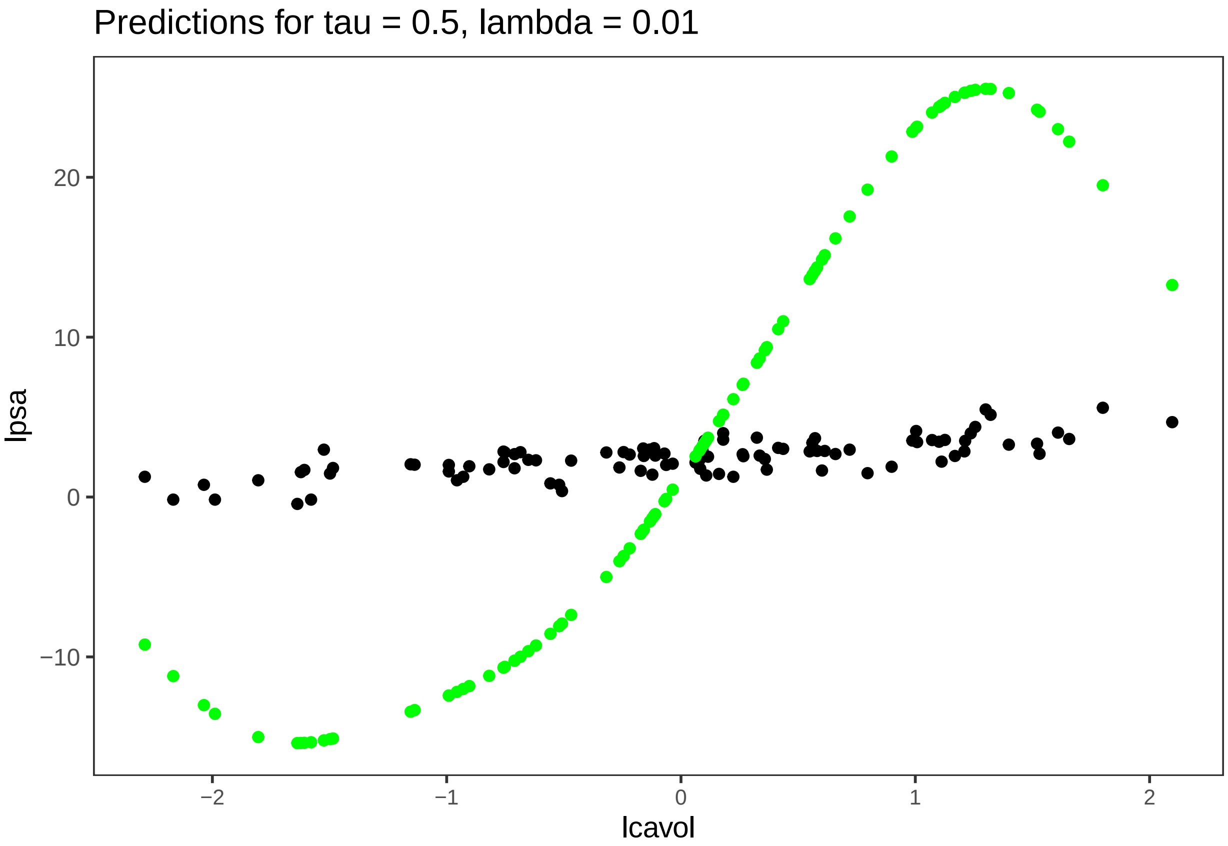
```
ggplot(plot_data_1,
       aes(x = V1, y = V2)) +
  geom_point(aes(x = V1, y = V2))+
  geom_point(aes(x = V1,
                 y = y_hat_1), color = "green")+
  xlab("lcavol") +
  ylab("lpsa") +
  theme_bw() +
  theme(axis.text.x=element_text(size=8),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())+
  ggtitle("Predictions for tau = 0.1, lambda = 5")
```



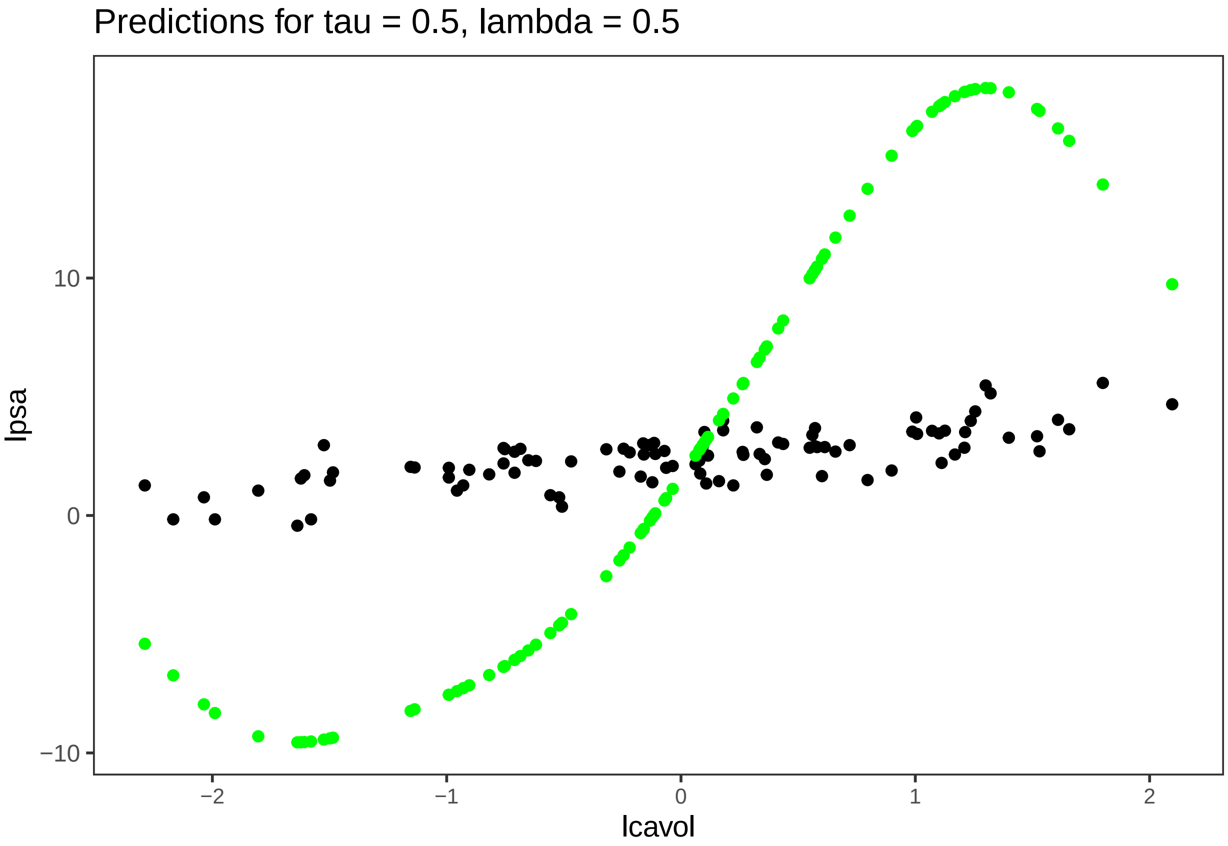
d) Next, we repeat the experiment above but with tau = 0.5

```
y = as.matrix(lprostate_data[,"lpsa"])

X = as.matrix(lprostate_data[,"lcavol"])
X_scaled = scale(X)

alpha_hat_1 <- unlist(fitKRR(X = X_scaled, y = t(y), tau = 0.5, lambda = 0.01)[1])

y_hat_1 <- predictKRR(X = t(X_scaled), Z = t(X_scaled), alpha = alpha_hat_1,
                      tau = 0.5, offset = mean(y))

plot_data_1 <- as.data.frame(cbind(X_scaled, y, y_hat_1))
```

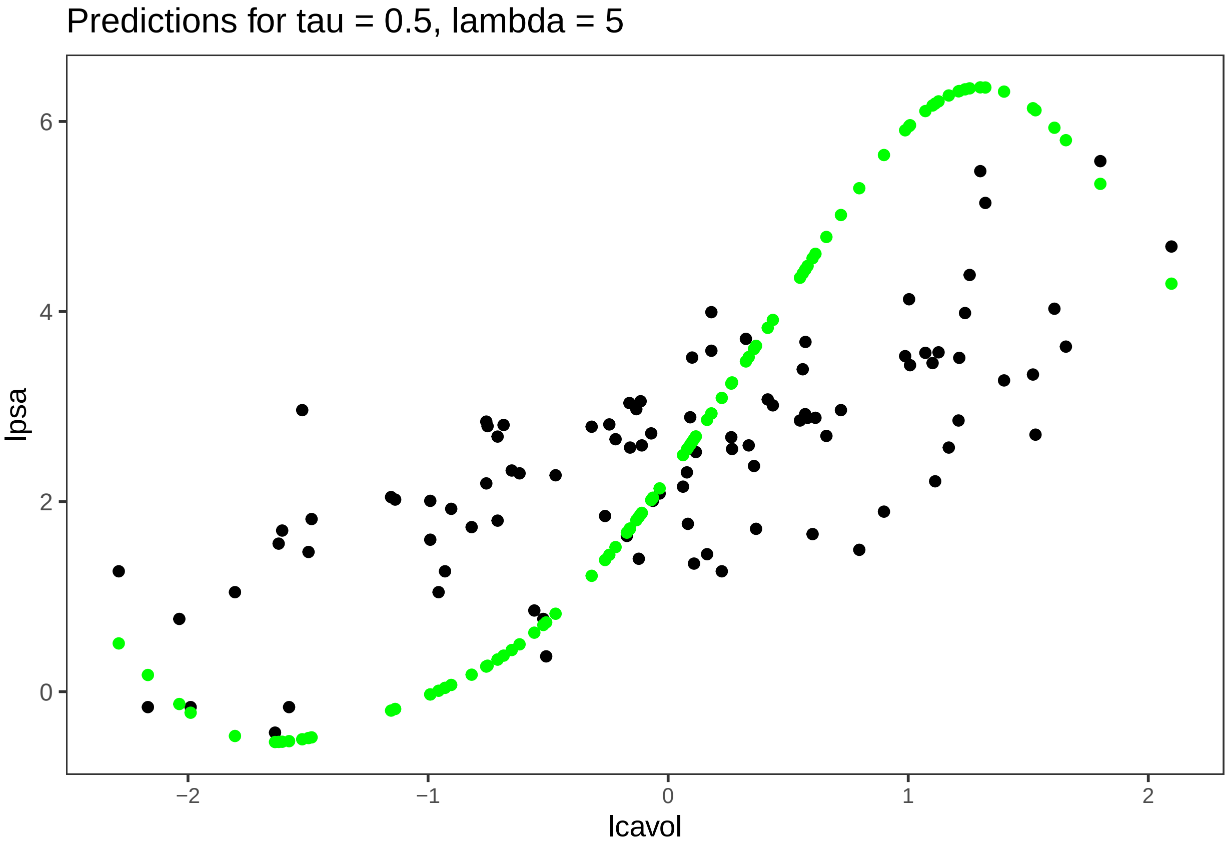
```
ggplot(plot_data_1,
       aes(x = V1, y = V2)) +
  geom_point(aes(x = V1, y = V2))+
  geom_point(aes(x = V1,
              y = y_hat_1), color = "green")+
  xlab("lcavol") +
  ylab("lpsa") +
  theme_bw() +
  theme(axis.text.x=element_text(size=8),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())+
  ggtitle("Predictions for tau = 0.5, lambda = 0.01")
```



```
y = as.matrix(lprostate_data[,"lpsa"])

X = as.matrix(lprostate_data[,"lcavol"])
X_scaled = scale(X)

alpha_hat_1 <- unlist(fitKRR(X = X_scaled, y = t(y), tau = 0.5, lambda = 0.5)[1])

y_hat_1 <- predictKRR(X = t(X_scaled), Z = t(X_scaled), alpha = alpha_hat_1,
                      tau = 0.5, offset = mean(y))

plot_data_1 <- as.data.frame(cbind(X_scaled, y, y_hat_1))

ggplot(plot_data_1,
       aes(x = V1, y = V2)) +
  geom_point(aes(x = V1, y = V2))+
  geom_point(aes(x = V1,
                y = y_hat_1), color = "green")+
  xlab("lcavol") +
  ylab("lpsa") +
  theme_bw() +
  theme(axis.text.x=element_text(size=8),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())+
  ggtitle("Predictions for tau = 0.5, lambda = 0.5")
```

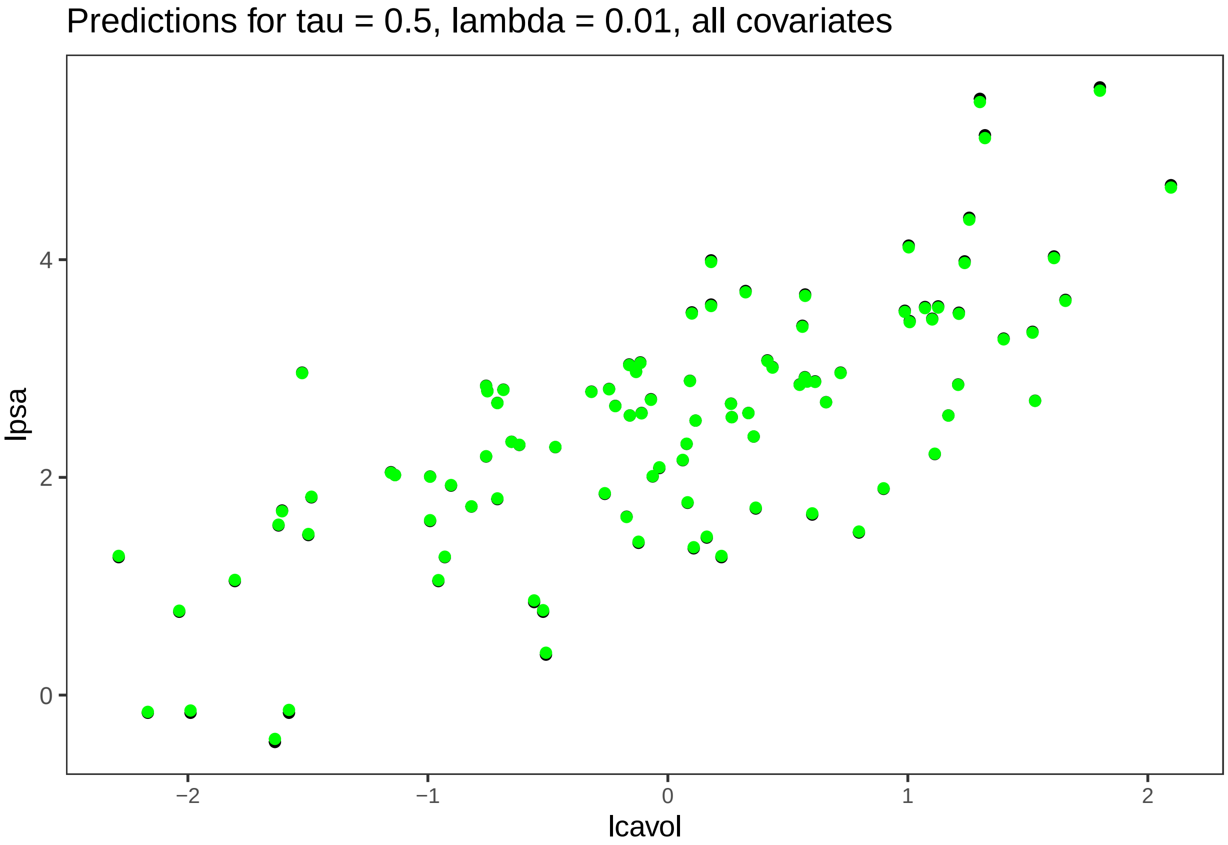


```
y = as.matrix(lprostate_data[,"lpsa"])

X = as.matrix(lprostate_data[,"lcavol"])
X_scaled = scale(X)

alpha_hat_1 <- unlist(fitKRR(X = X_scaled, y = t(y), tau = 0.5, lambda = 5)[1])

y_hat_1 <- predictKRR(X = t(X_scaled), Z = t(X_scaled), alpha = alpha_hat_1,
                      tau = 0.5, offset = mean(y))

plot_data_1 <- as.data.frame(cbind(X_scaled, y, y_hat_1))

ggplot(plot_data_1,
       aes(x = V1, y = V2)) +
  geom_point(aes(x = V1, y = V2))+
  geom_point(aes(x = V1,
                y = y_hat_1), color = "green")+
  xlab("lcavol") +
  ylab("lpsa") +
  theme_bw() +
  theme(axis.text.x=element_text(size=8),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())+
  ggtitle("Predictions for tau = 0.5, lambda = 5")
```

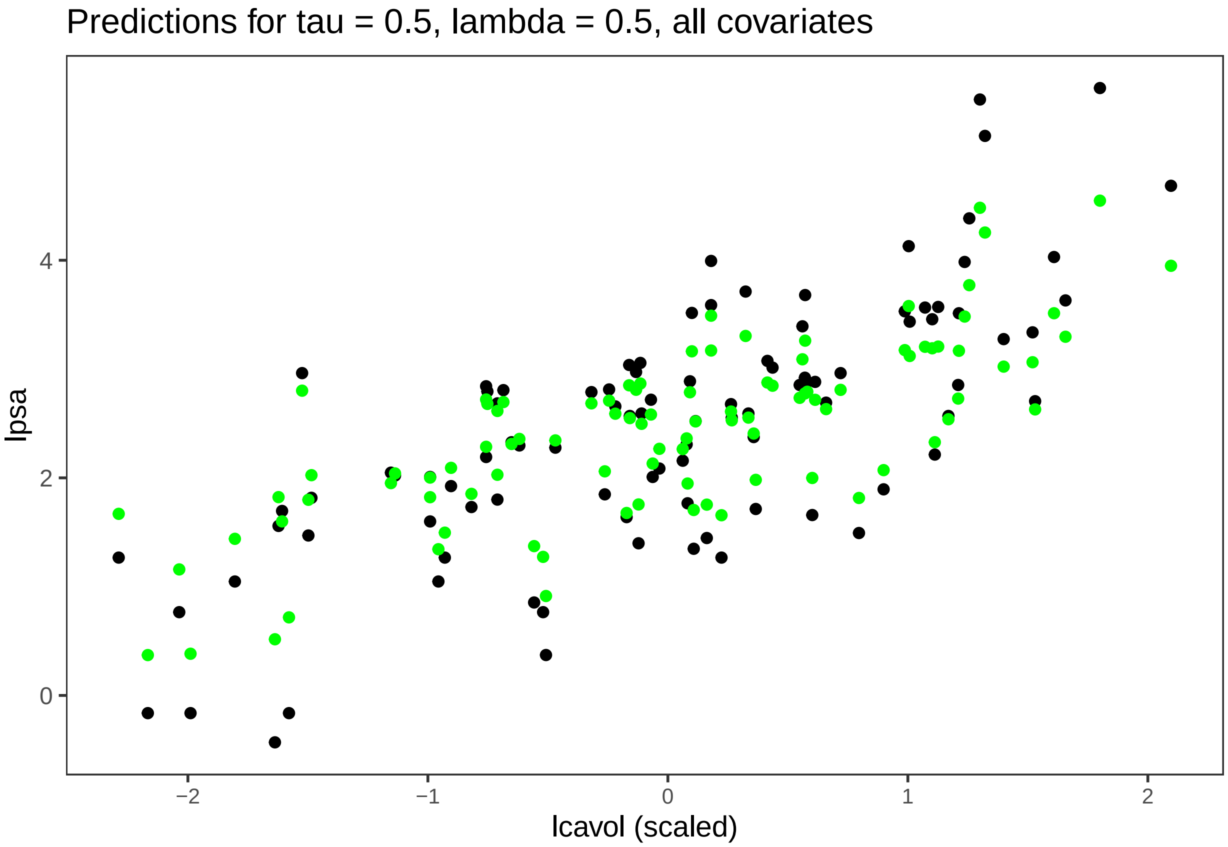


e) Now, we use all covariates to predict lpsa.

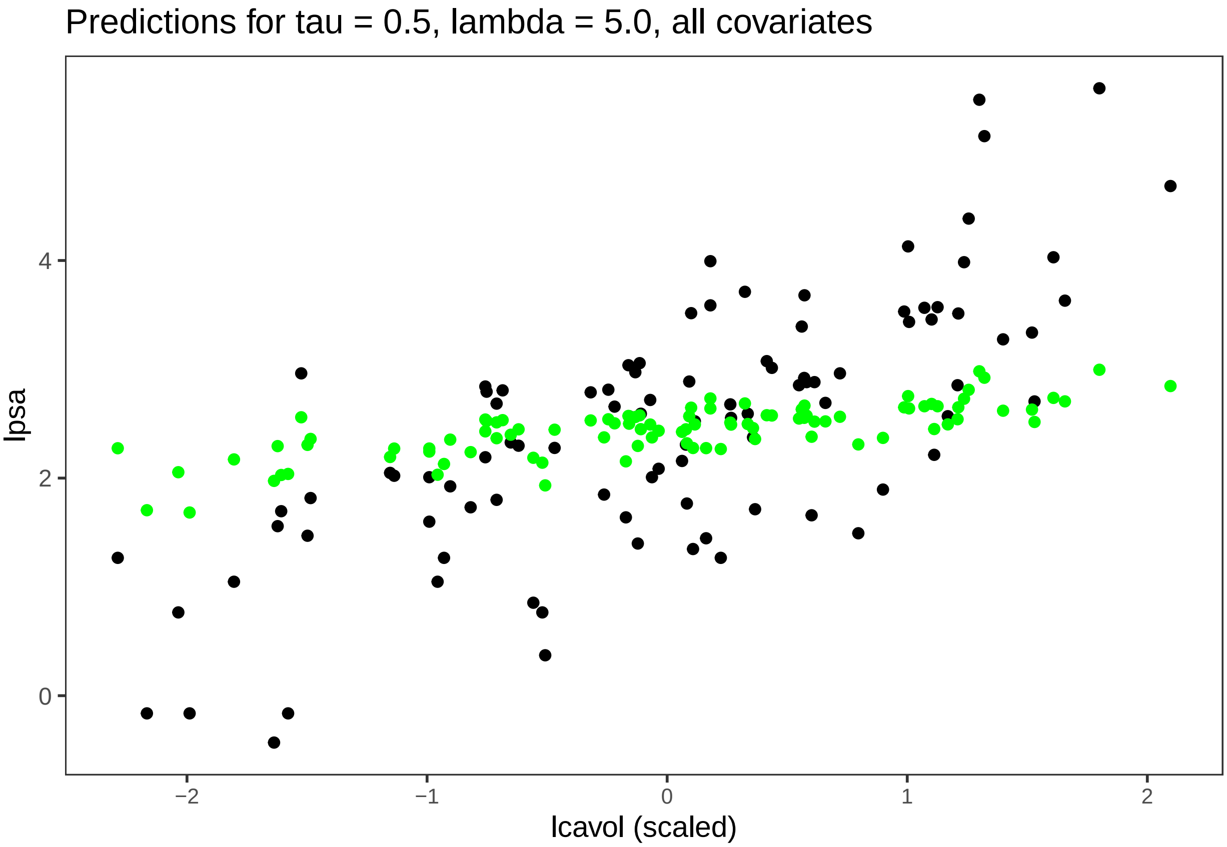
```
y = as.matrix(lprostate_data[,"lpsa"])

X = as.matrix(lprostate_data[,-9])
X_scaled = scale(X)

alpha_hat_1 <- unlist(fitKRR(X = t(X_scaled), y = y, tau = 0.5, lambda = 0.01)[1])

y_hat_1 <- predictKRR(X = t(X_scaled), Z = t(X_scaled), alpha = alpha_hat_1,
                      tau = 0.5, offset = mean(y))

plot_data_1 <- as.data.frame(cbind(X_scaled[,1], y, y_hat_1))

ggplot(plot_data_1,
       aes(x = V1, y = V2)) +
  geom_point(aes(x = V1, y = V2))+
  geom_point(aes(x = V1,
                 y = y_hat_1), color = "green")+
  xlab("lcavol") +
  ylab("lpsa") +
  theme_bw() +
  theme(axis.text.x=element_text(size=8),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())+
  ggtitle("Predictions for tau = 0.5, lambda = 0.01, all covariates")
```



```
y = as.matrix(lprostate_data[,"lpsa"])

X = as.matrix(lprostate_data[,-9])
X_scaled = scale(X)

alpha_hat_1 <- unlist(fitKRR(X = t(X_scaled), y = y, tau = 0.5, lambda = 0.5)[1])

y_hat_1 <- predictKRR(X = t(X_scaled), Z = t(X_scaled), alpha = alpha_hat_1,
                      tau = 0.5, offset = mean(y))

plot_data_1 <- as.data.frame(cbind(X_scaled[,1], y, y_hat_1))

ggplot(plot_data_1,
       aes(x = V1, y = V2)) +
  geom_point(aes(x = V1, y = V2))+
  geom_point(aes(x = V1,
                 y = y_hat_1), color = "green")+
  xlab("lcavol (scaled)") +
  ylab("lpsa") +
  theme_bw() +
  theme(axis.text.x=element_text(size=8),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())+
  ggtitle("Predictions for tau = 0.5, lambda = 0.5, all covariates")
```



```
y = as.matrix(lprostate_data[,"lpsa"])

X = as.matrix(lprostate_data[,-9])
X_scaled = scale(X)

alpha_hat_1 <- unlist(fitKRR(X = t(X_scaled), y = y, tau = 0.5, lambda = 5.0)[1])

y_hat_1 <- predictKRR(X = t(X_scaled), Z = t(X_scaled), alpha = alpha_hat_1,
                      tau = 0.5, offset = mean(y))

plot_data_1 <- as.data.frame(cbind(X_scaled[,1], y, y_hat_1))

ggplot(plot_data_1,
       aes(x = V1, y = V2)) +
  geom_point(aes(x = V1, y = V2))+
  geom_point(aes(x = V1,
                 y = y_hat_1), color = "green")+
  xlab("lcavol (scaled)") +
  ylab("lpsa") +
  theme_bw() +
  theme(axis.text.x=element_text(size=8),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())+
  ggtitle("Predictions for tau = 0.5, lambda = 5.0, all covariates")
```



f) First we calculate the sigma hat squared

```
y = as.matrix(lprostate_data[,"lpsa"])

X = as.matrix(lprostate_data[,-9])
X_scaled = scale(X)
```