a1q2

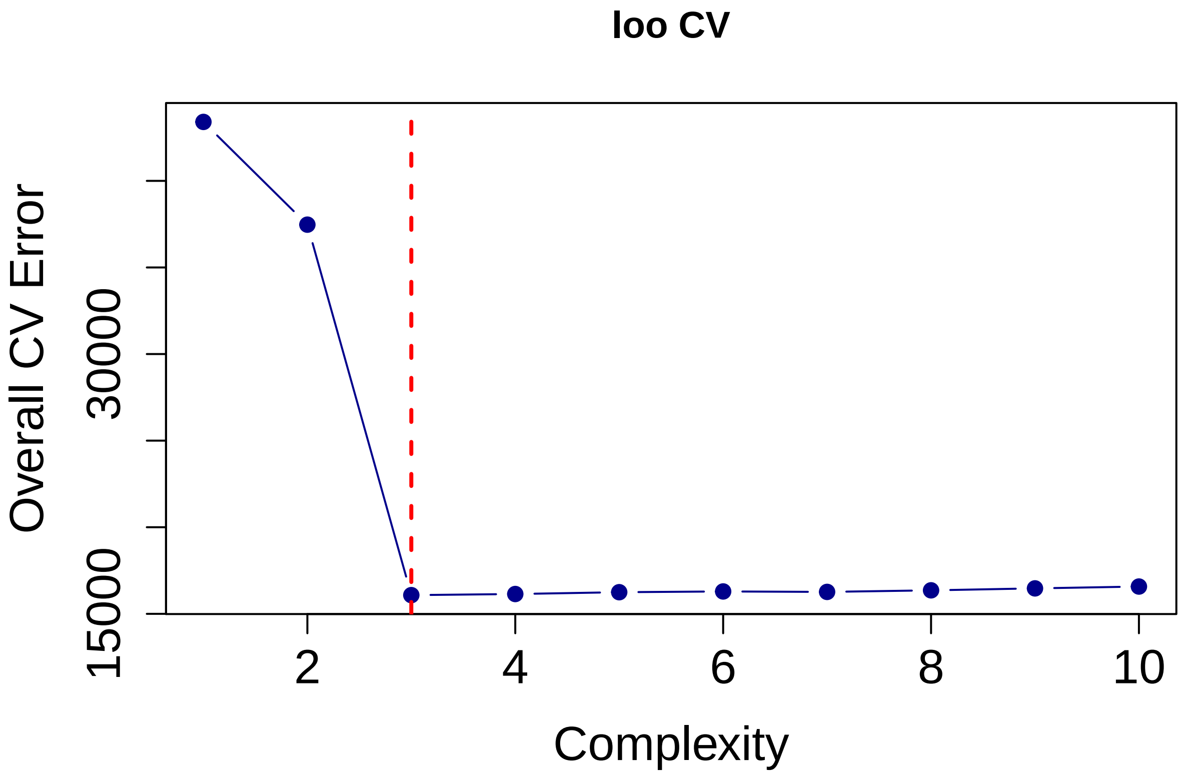
Mushi Wang

02/06/2020

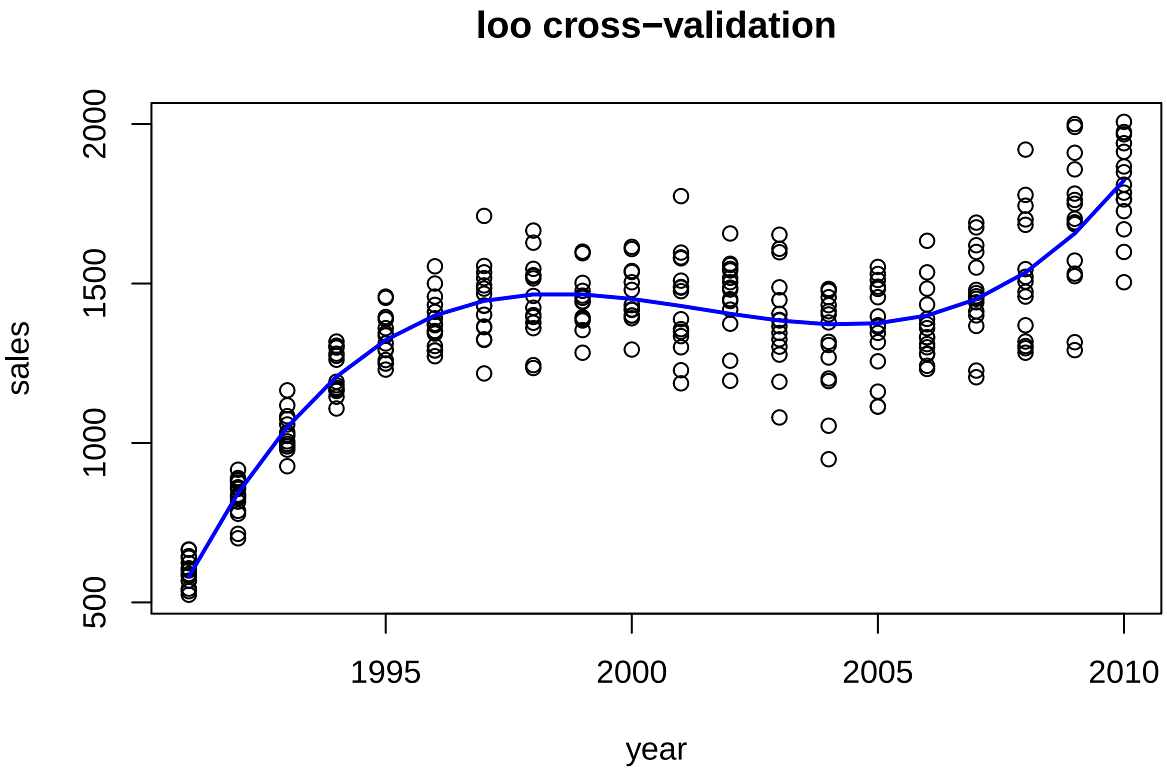
q2

```r
sales = read.table("../JaxSales.txt", header = TRUE)

# A function to generate the indices of the k-fold sets
kfold <- function(N, k=N, indices=NULL){
  # get the parameters right:
  if (is.null(indices)) {
    # Randomize if the index order is not supplied
    indices <- sample(1:N, N, replace=FALSE)
  } else {
    # else if supplied, force N to match its length
    N <- length(indices)
  }
  # Check that the k value makes sense.
  if (k > N) stop("k must not exceed N")
  #

  # How big is each group?
  gsize <- rep(round(N/k), k)

  # For how many groups do we need odjust the size?
  extra <- N - sum(gsize)

  # Do we have too few in some groups?
  if (extra > 0) {
    for (i in 1:extra) {
      gsize[i] <- gsize[i] +1
    }
  }
  # Or do we have too many in some groups?
  if (extra < 0) {
    for (i in 1:abs(extra)) {
      gsize[i] <- gsize[i] - 1
    }
  }

  running_total <- c(0,cumsum(gsize))

  # Return the list of k groups of indices
  lapply(1:k,
         FUN=function(i) {
           indices[seq(from = 1 + running_total[i],
                       to = running_total[i+1],
                       by = 1)
                   ]
         }
  )
}


# A function to form the k samples
getKfoldSamples <- function (x, y, k, indices=NULL){
  groups <- kfold(length(x), k, indices)
  #training sets
  Ssamples <- lapply(groups,
                     FUN=function(group) {
                       list(x=x[-group], y=y[-group])
                     })
  #test set
  Tsamples <- lapply(groups,
                     FUN=function(group) {
                       list(x=x[group], y=y[group])
                     })
  list(Ssamples = Ssamples, Tsamples = Tsamples)
}
```

```r
# For leave one out cross-validation
samples_loocv <-  getKfoldSamples(sales$Year, sales$Sales, k=length(sales$Sales))

# the degrees of freedom associated with each
complexity <- c(1:10) # These are the degrees of polynomials to be fitted

# Performing the Cross-Validation
Ssamples <- samples_loocv$Ssamples # change this accorcing to the number of folds
Tsamples <- samples_loocv$Tsamples # change this accorcing to the number of folds
CV.To.Plot = data.frame(Complexity=NA , MSE=NA)
for(i in 1:length(complexity)){
  MSE = c()
  for(j in 1:length(Ssamples)){
    x.temp = Ssamples[[j]]$x
    y.temp = Ssamples[[j]]$y
    model = lm(y.temp~poly(x.temp, complexity[i]))
    pred = predict(model, newdata=data.frame(x.temp=Tsamples[[j]]$x))
    MSE[j] = mean((Tsamples[[j]]$y-pred)^2)
  }
  CV.To.Plot[i,] = c(complexity[i], mean(MSE))
}


Title.Graph = "loo CV" # change this accorcing to the number of folds
plot(CV.To.Plot, pch=19, col="darkblue", type="b",
     cex.axis = 1.5, cex.lab=1.5, ylab="Overall CV Error")
indx = which.min(CV.To.Plot$MSE)
abline(v=indx, lty=2, lwd=2, col='red')
title(main=Title.Graph)
```

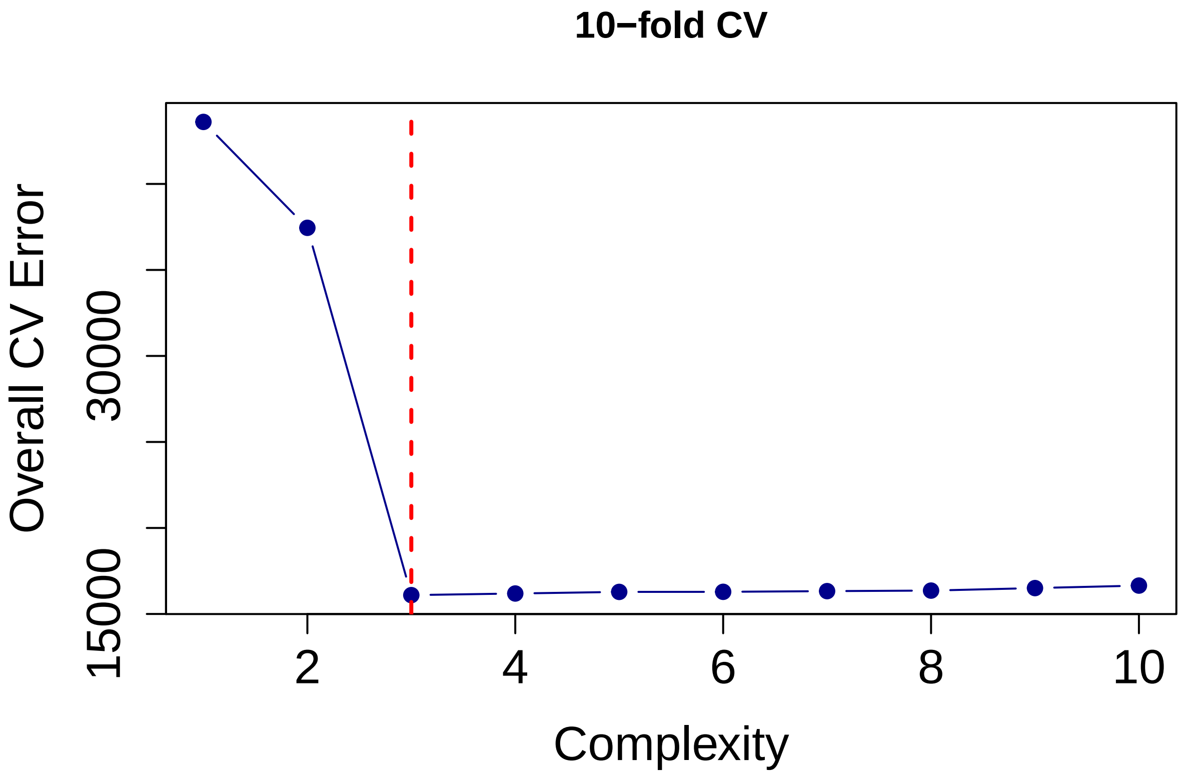


```
plot(sales$Year, sales$Sales, xlab = "year", ylab = "sales", main = "loo cross-validation")
lines(sales$Year, predict(lm(sales$Sales~poly(sales$Year,3))), type="l", col="blue", lwd=2)
```

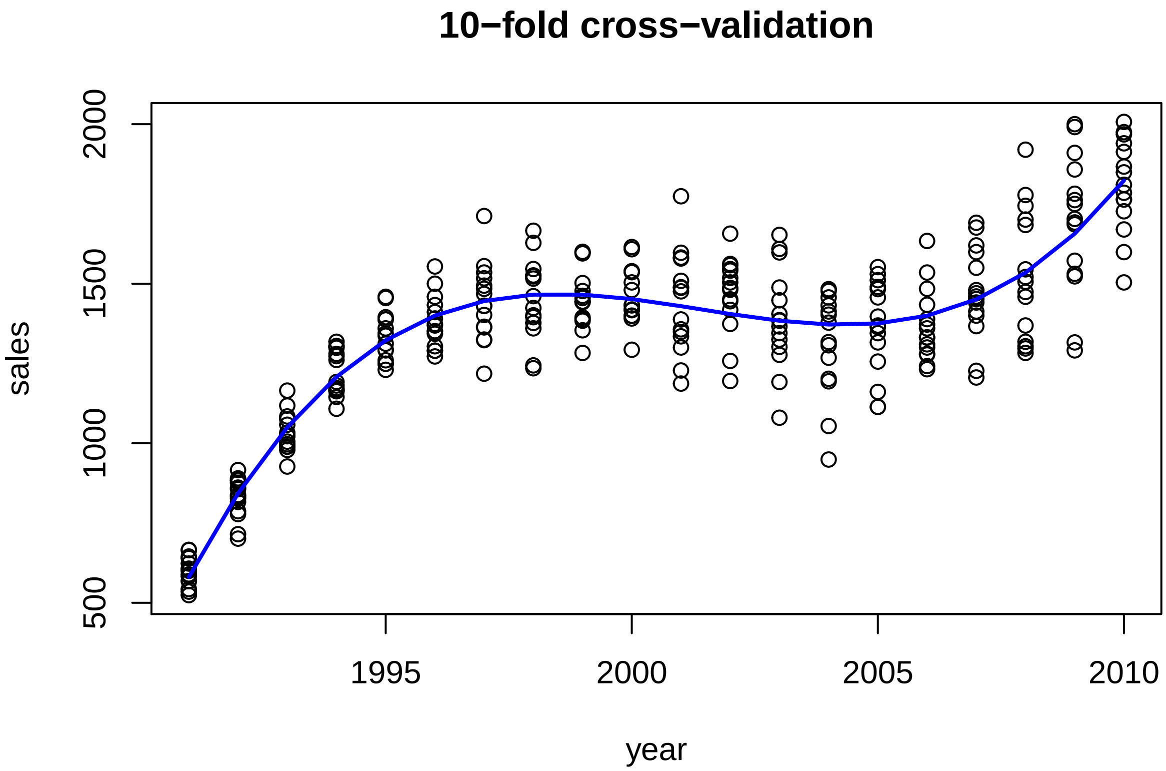

loo cross−validation
sales
2000
1500
1000
500
1995
2000
2005
2010
year

(b)

```
# For leave one out cross-validation
samples_10fold <- getKfoldSamples(sales$Year, sales$Sales, k=10)

# the degrees of freedom associated with each
complexity <- c(1:10) # These are the degrees of polynomials to be fitted

# Performing the Cross-Validation
Ssamples <- samples_10fold$Ssamples # change this accorcing to the number of folds
Tsamples <- samples_10fold$Tsamples # change this accorcing to the number of folds
CV.To.Plot = data.frame(Complexity=NA , MSE=NA)
for(i in 1:length(complexity)){
  MSE = c()
  for(j in 1:length(Ssamples)){
    x.temp = Ssamples[[j]]$x
    y.temp = Ssamples[[j]]$y
    model = lm(y.temp~poly(x.temp, complexity[i]))
    pred = predict(model, newdata=data.frame(x.temp=Tsamples[[j]]$x))
    MSE[j] = mean((Tsamples[[j]]$y-pred)^2)
  }
  CV.To.Plot[i,] = c(complexity[i], mean(MSE))
}


Title.Graph = "10-fold CV" # change this accorcing to the number of folds
plot(CV.To.Plot, pch=19, col="darkblue", type="b",
     cex.axis = 1.5, cex.lab=1.5, ylab="Overall CV Error")
indx = which.min(CV.To.Plot$MSE)
abline(v=indx, lty=2, lwd=2, col='red')
title(main=Title.Graph)
```



```
plot(sales$Year, sales$Sales, xlab = "year", ylab = "sales", main = "10-fold cross-validation")
lines(sales$Year, predict(lm(sales$Sales~poly(sales$Year, 3))), type="l", col="blue", lwd=2)
```


10−fold cross−validation
sales
500
1000
1500
2000
1995
2000
2005
2010
year

(c) The two models above result in same model where complexity is 3. I prefer $k = 10$. Even though they result in the same model and LOO cross-validation is approximately unbiased. However, LOO cross-validation causes high variance.