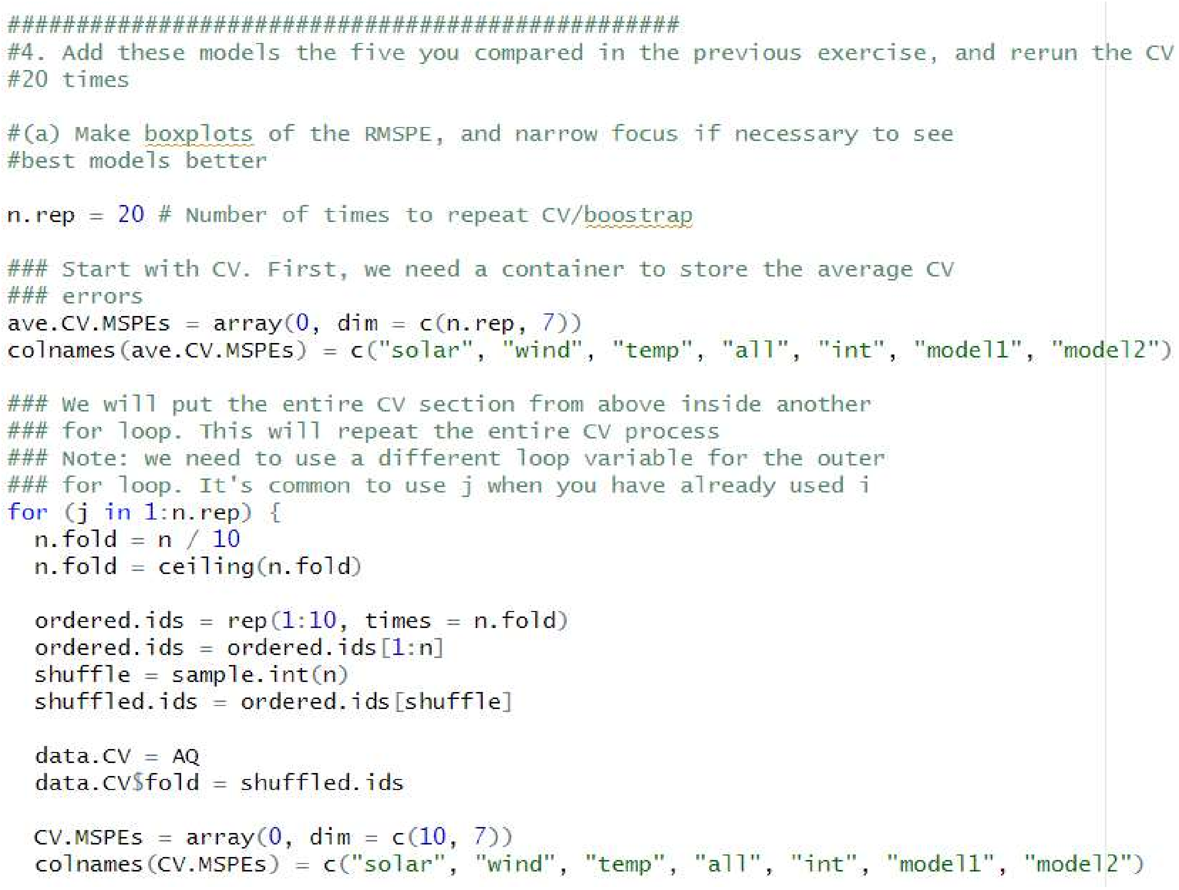
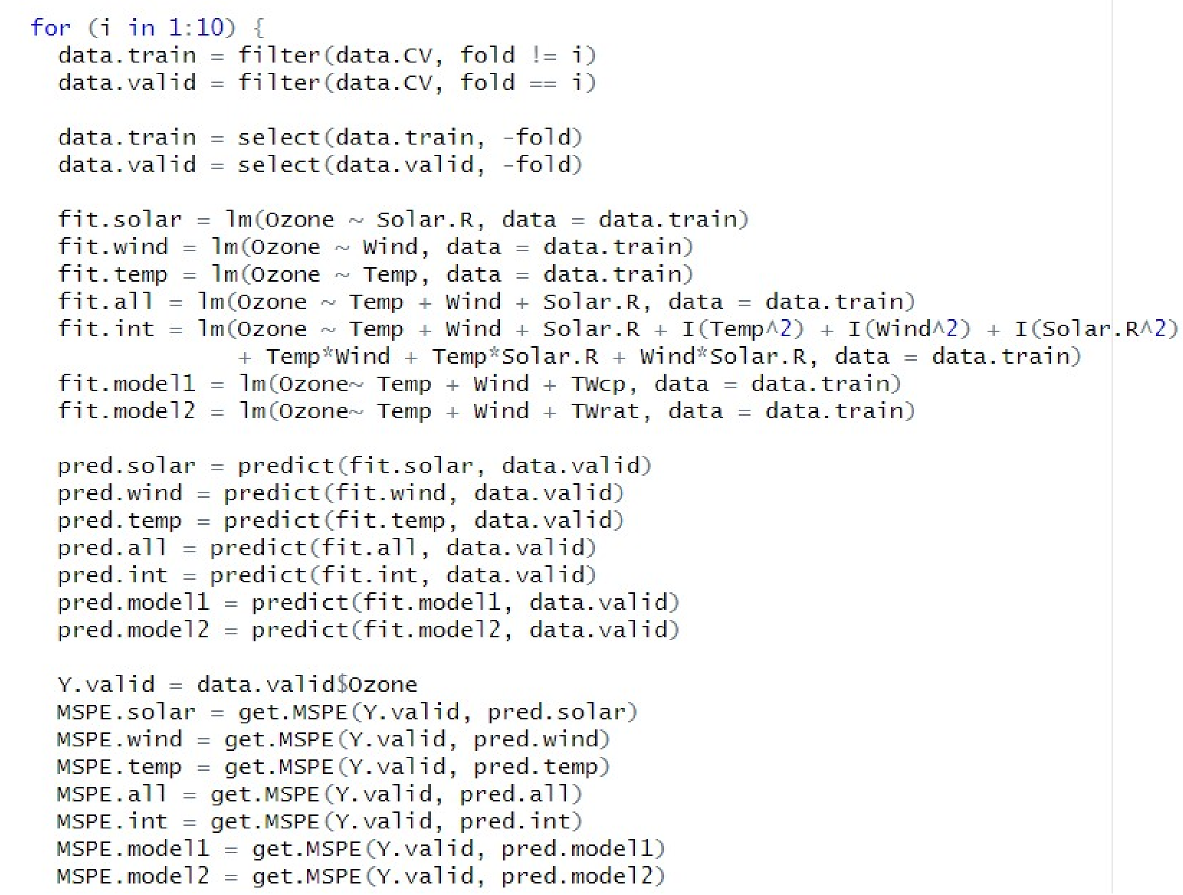
4. Add these models the five you compared in the previous exercise, and rerun the CV 20 times.

(a) **Make boxplots of the RMSPE, and narrow focus if necessary to see best models better.**

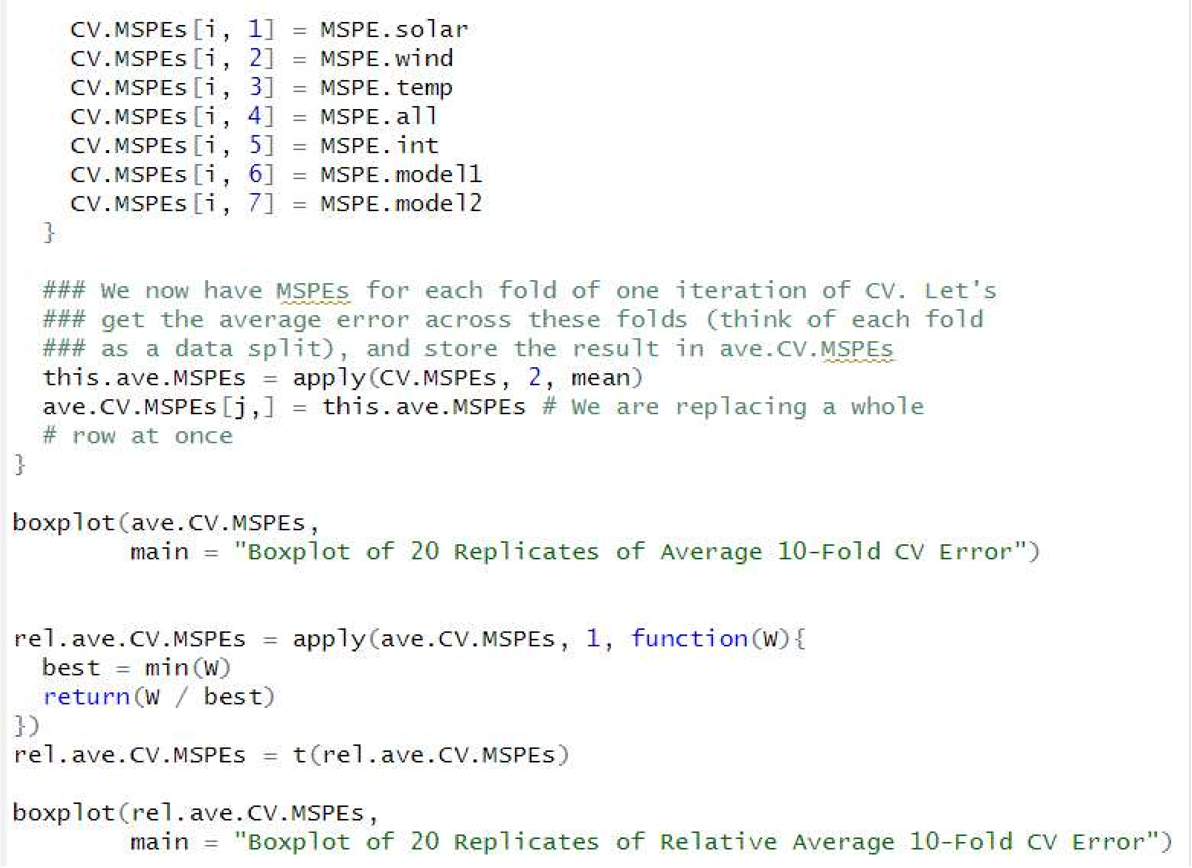
(b) Are any of the new models competitive, or even best? **(1 sentence)**



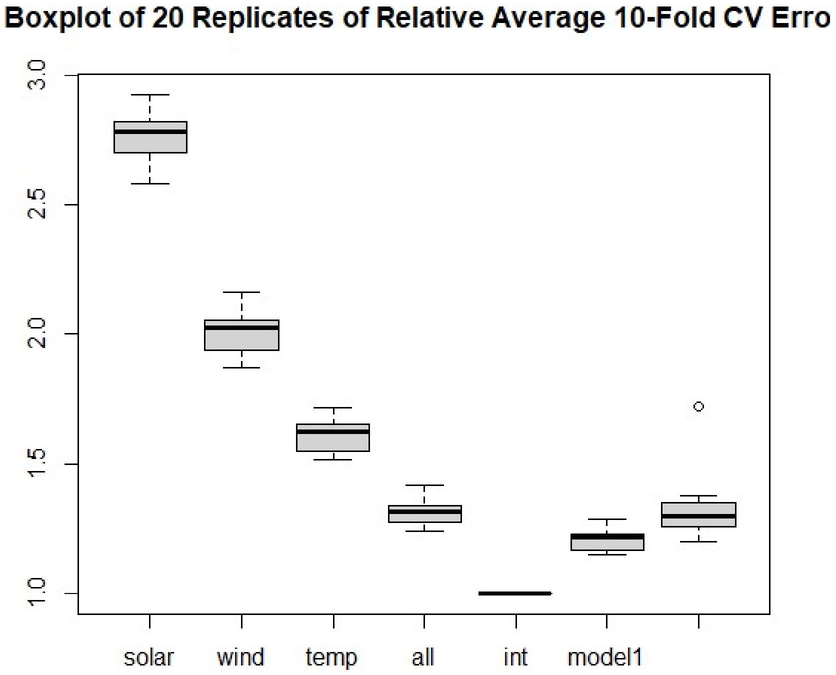
```
#################################################
#4. Add these models the five you compared in the previous exercise, and rerun the CV
#20 times

#(a) Make boxplots of the RMSPE, and narrow focus if necessary to see
#best models better

n.rep = 20 # Number of times to repeat CV/boostrap

### Start with CV. First, we need a container to store the average CV
### errors
ave.CV.MSPEs = array(0, dim = c(n.rep, 7))
colnames(ave.CV.MSPEs) = c("solar", "wind", "temp", "all", "int", "model1", "model2")

### We will put the entire CV section from above inside another
### for loop. This will repeat the entire CV process
### Note: we need to use a different loop variable for the outer
### for loop. It's common to use j when you have already used i
for (j in 1:n.rep) {
  n.fold = n / 10
  n.fold = ceiling(n.fold)

  ordered.ids = rep(1:10, times = n.fold)
  ordered.ids = ordered.ids[1:n]
  shuffle = sample.int(n)
  shuffled.ids = ordered.ids[shuffle]

  data.CV = AQ
  data.CV$fold = shuffled.ids

  CV.MSPEs = array(0, dim = c(10, 7))
  colnames(CV.MSPEs) = c("solar", "wind", "temp", "all", "int", "model1", "model2")
```



```r
  for (i in 1:10) {
    data.train = filter(data.CV, fold != i)
    data.valid = filter(data.CV, fold == i)

    data.train = select(data.train, -fold)
    data.valid = select(data.valid, -fold)

    fit.solar = lm(Ozone ~ Solar.R, data = data.train)
    fit.wind = lm(Ozone ~ Wind, data = data.train)
    fit.temp = lm(Ozone ~ Temp, data = data.train)
    fit.all = lm(Ozone ~ Temp + Wind + Solar.R, data = data.train)
    fit.int = lm(Ozone ~ Temp + Wind + Solar.R + I(Temp^2) + I(Wind^2) + I(Solar.R^2)
                 + Temp*Wind + Temp*Solar.R + Wind*Solar.R, data = data.train)
    fit.model1 = lm(Ozone~ Temp + Wind + TWcp, data = data.train)
    fit.model2 = lm(Ozone~ Temp + Wind + TWrat, data = data.train)

    pred.solar = predict(fit.solar, data.valid)
    pred.wind = predict(fit.wind, data.valid)
    pred.temp = predict(fit.temp, data.valid)
    pred.all = predict(fit.all, data.valid)
    pred.int = predict(fit.int, data.valid)
    pred.model1 = predict(fit.model1, data.valid)
    pred.model2 = predict(fit.model2, data.valid)

    Y.valid = data.valid$Ozone
    MSPE.solar = get.MSPE(Y.valid, pred.solar)
    MSPE.wind = get.MSPE(Y.valid, pred.wind)
    MSPE.temp = get.MSPE(Y.valid, pred.temp)
    MSPE.all = get.MSPE(Y.valid, pred.all)
    MSPE.int = get.MSPE(Y.valid, pred.int)
    MSPE.model1 = get.MSPE(Y.valid, pred.model1)
    MSPE.model2 = get.MSPE(Y.valid, pred.model2)
```



```r
    CV.MSPEs[i, 1] = MSPE.solar
    CV.MSPEs[i, 2] = MSPE.wind
    CV.MSPEs[i, 3] = MSPE.temp
    CV.MSPEs[i, 4] = MSPE.all
    CV.MSPEs[i, 5] = MSPE.int
    CV.MSPEs[i, 6] = MSPE.model1
    CV.MSPEs[i, 7] = MSPE.model2
  }

  ### We now have MSPEs for each fold of one iteration of CV. Let's
  ### get the average error across these folds (think of each fold
  ### as a data split), and store the result in ave.CV.MSPEs
  this.ave.MSPEs = apply(CV.MSPEs, 2, mean)
  ave.CV.MSPEs[j,] = this.ave.MSPEs # We are replacing a whole
  # row at once
}

boxplot(ave.CV.MSPEs,
        main = "Boxplot of 20 Replicates of Average 10-Fold CV Error")


rel.ave.CV.MSPEs = apply(ave.CV.MSPEs, 1, function(W){
  best = min(W)
  return(W / best)
})
rel.ave.CV.MSPEs = t(rel.ave.CV.MSPEs)

boxplot(rel.ave.CV.MSPEs,
        main = "Boxplot of 20 Replicates of Relative Average 10-Fold CV Error")
```



```
#(b) Are any of the new models competitive, or even best? (1 sentence)
# No the existing model that are with interaction is best
```