Hwk 07 Solutions L10

```
################################
### Import and process data ###
################################

### Import and clean the air quality data
data("airquality")
AQ.raw = na.omit(airquality[,1:4])

### Construct new variables
AQ = AQ.raw
AQ$TWcp = with(AQ.raw, Temp * Wind)
AQ$TWrat = with(AQ.raw, Temp / Wind)

########################
### Helper Functions ###
########################

### Create function to compute MSPEs
get.MSPE = function(Y, Y.hat){
  return(mean((Y - Y.hat)^2))
}

### Create function which constructs folds for CV
### n is the number of observations, K is the number of folds
get.folds = function(n, K) {
  ### Get the appropriate number of fold labels
  n.fold = ceiling(n / K) # Number of observations per fold (rounded up)
  fold.ids.raw = rep(1:K, times = n.fold)
  fold.ids = fold.ids.raw[1:n]

  ### Shuffle the fold labels
  folds.rand = fold.ids[sample.int(n)]

  return(folds.rand)
}
```

Applications

Question 1

Part (a)

```r
### Extract predictor and response variables
X = AQ$Temp
Y = AQ$Ozone

### Fit smoothing splines with various DF
fit.smooth.5 = smooth.spline(X, Y, df = 5)
fit.smooth.7 = smooth.spline(X, Y, df = 7)
fit.smooth.9 = smooth.spline(X, Y, df = 9)
fit.smooth.20 = smooth.spline(X, Y, df = 20)

### Construct plot
with(AQ, plot(Temp, Ozone, xlab = "Temperature"))
lines(fit.smooth.5, col = 2, lwd = 2)
lines(fit.smooth.7, col = 3, lwd = 2)
lines(fit.smooth.9, col = 4, lwd = 2)
lines(fit.smooth.20, col = 5, lwd = 2)


### Add legend
legend("topleft", legend = c("Smoothing Spline - 5", "Smoothing Spline - 7",
  "Smoothing Spline - 9", "Smoothing Spline - 20"),
  col = 2:5, lty = 1, lwd = 2)
```

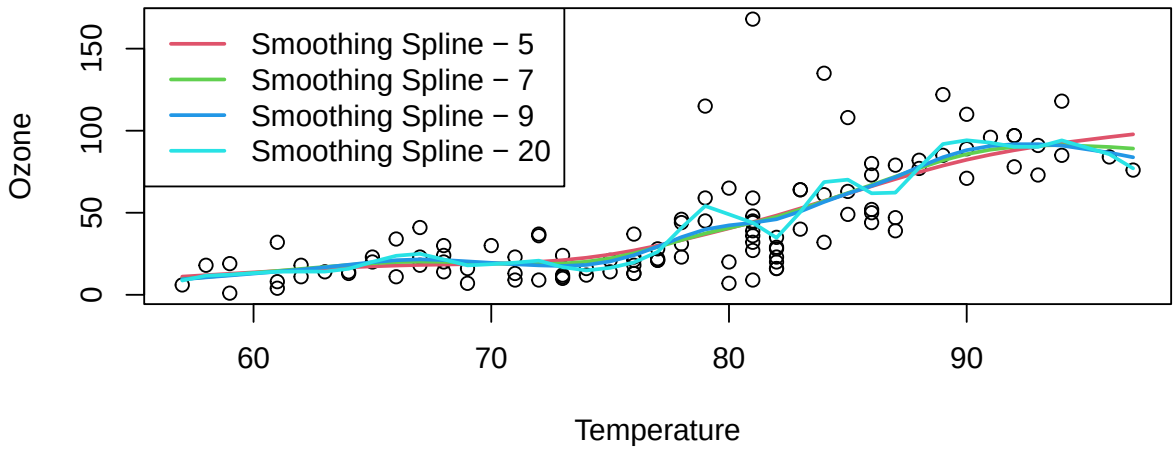

Figure 1: Smoothing Splines

See Figure 1 for a comparison of smoothing splines with various degrees of freedom. If it had to choose one model, it would be the one with 5 degrees of freedom (red), because this model does a pretty good job of following the overall trend of the data without being too wiggly.

Part (b)

```
### Fit models
fit.smooth.CV = smooth.spline(X, Y, cv = T)
fit.smooth.GCV = smooth.spline(X, Y, cv = F)

### Extract df
df.CV = fit.smooth.CV$df
df.GCV = fit.smooth.GCV$df

### Create Plot
with(AQ, plot(Temp, Ozone, xlab = "Temperature"))
lines(fit.smooth.CV, col = 2, lwd = 2)
lines(fit.smooth.GCV, col = 3, lwd = 2)

### Add legend
legend("topleft", legend = c("Smoothing Spline - CV", "Smoothing Spline - GCV"), col = 2:3, lty = 1, lw
```

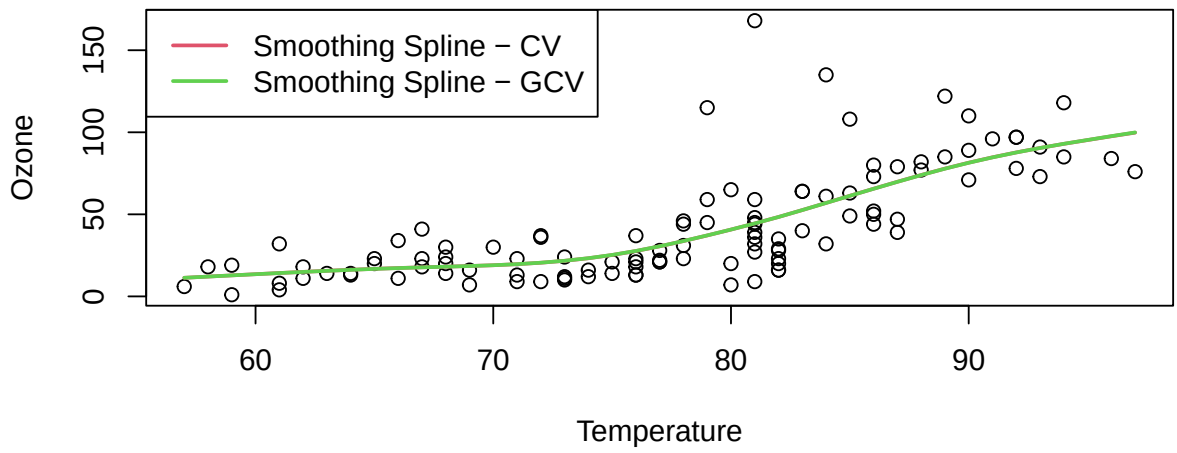

Figure 2: Tuned Smoothing Splines

See Figure 2 for a comparison of smoothing splines with degrees of freedom chosen by tuning. The two models are nearly identical, and both fit the data quite well.

Question 2

```
### Fit models
fit.loess.5 = loess(Ozone ~ Temp, data = AQ, enp.target = 5)
fit.loess.7 = loess(Ozone ~ Temp, data = AQ, enp.target = 7)
fit.loess.9 = loess(Ozone ~ Temp, data = AQ, enp.target = 9)
fit.loess.20 = loess(Ozone ~ Temp, data = AQ, enp.target = 20)
```

```r
### Construct grid in Temp
seq.temp = data.frame(Temp = seq(57, 97, by = 0.5))
pred.loess.5 = predict(fit.loess.5, seq.temp)
pred.loess.7 = predict(fit.loess.7, seq.temp)
pred.loess.9 = predict(fit.loess.9, seq.temp)
pred.loess.20 = predict(fit.loess.20, seq.temp)

### Create plot
with(AQ, plot(Temp, Ozone, xlab = "Temperature"))
lines(seq.temp$Temp, pred.loess.5, col = 2, lwd = 2)
lines(seq.temp$Temp, pred.loess.7, col = 3, lwd = 2)
lines(seq.temp$Temp, pred.loess.9, col = 4, lwd = 2)
lines(seq.temp$Temp, pred.loess.20, col = 5, lwd = 2)

### Add legend
legend("topleft", legend = c("LOESS - 5", "LOESS - 7", "LOESS - 9", "LOESS - 20"), col = 2:5, lty = 1,
```

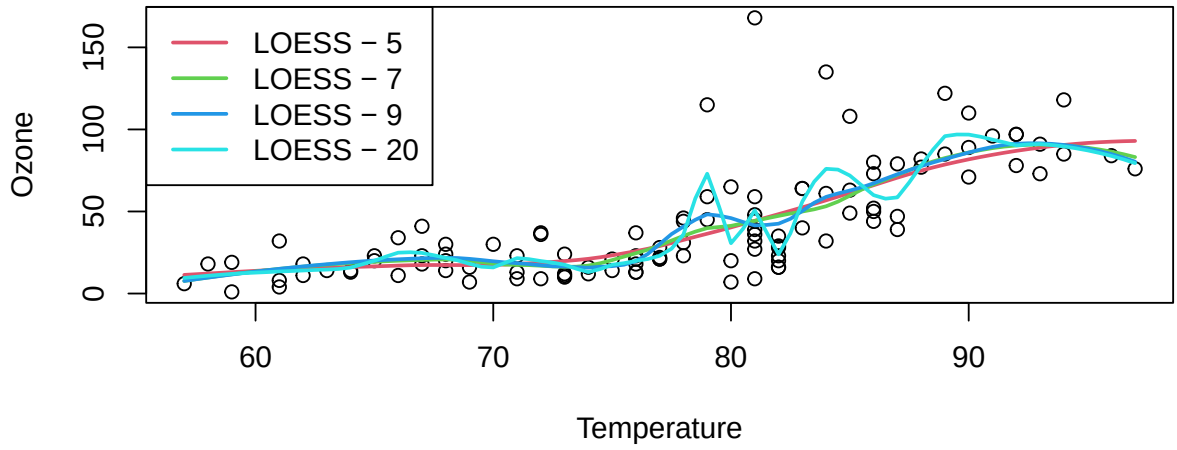

Figure 3: LOESS

See Figure 3 for a comparison of loess models with various degrees of freedom. I especially prefer the model with 5 degrees of freedom here, because the other models all curve down dramatically at the end and I don't think that the little data available there should have such a dramatic impact on the fit.