# Setup
library(MASS)
data(geyser)
summary(geyser)

1. **Answer:**

   # Plot the data points
   plot(geyser$duration, geyser$waiting, cex = 0.5, pch = 16, 
       main = "Waiting time as function of geyser duration",
       xlab = "Duration (minutes)", ylab = "Waiting time (minutes)"
   mtext("Black line = LS regression line")
   # Build linear model:
   lm.1 = lm(waiting ~ duration, data = geyser)
   # Add the regression line to the plot
   abline(lm.1)

   See Figure 1. Values of duration seem to cluster around 2 minutes and 4-5 minutes. The cluster on the right has more variation in waiting times compared to the cluster on the left.

2. **Answer:**

   # Plot the squared residuals against duration
   plot(geyser$duration, lm.1$residuals^2, cex = 0.5, pch = 16, 
       main = "Squared residuals versus geyser duration",
       xlab = "Geyser duration", ylab = "Squared residuals from linear model")

   See Figure 2. The squared residuals seem to be largest, on average, near duration = 4.5, and substantially smaller for the cluster of observations near duration = 2.

3. **Answer:**

   # To estimate the variance, I will use the same kernel regression function
   # from HW Assignment 2:
Figure 1:
require(np)
reg.var3 = npreg(lm.1$residuals^2 ~ geyser$duration)
# Pair each observation, indexed by duration, with its fitted value
fits = cbind(geyser$duration, reg.var3$mean)
# Order the observation-fit pairs by duration (column 1)
fits3 = fits[order(fits[,1]),]
# Add the fits to the plot
lines(fits3, col=2, lwd = 2.5)
mtext("Red curve: kernel regression function", col = 2)

Again see Figure 2. The regression curve estimates the variance of \textit{waiting|duration} (under the assumption that the expectation of \textit{waiting} at each fixed value \textit{duration} is given by the linear regression line in Figure 1). The variance estimate changes substantially across changes in \textit{duration}, and this indicates that the noise is heteroskedastic.

Figure 2:

4. \textbf{Answer:}

# Build a weighted linear model:
lm.4 = lm(waiting ~ duration, data = geyser, weights = 1/reg.var3$mean)

# Find the coefficient for the unweighted linear model
> summary(lm.1)$coefficients #output includes the following:
   Estimate Std. Error
(Intercept) 99.309856 1.9569392
duration   -7.800326 0.5367926

# Find the coefficient for the weighted linear model
> summary(lm.4)$coefficients #output includes the following:
   Estimate Std. Error
(Intercept) 98.896429 1.3303701
duration    -7.808154 0.4320001

The difference between the two coefficients for duration is only about
0.008, while the corresponding standard errors are at least 50 times as
large, indicating that the difference in slopes is not statistically significant.
Even if this difference were statistically significant, the difference is too
small to matter much, at least from the perspective of a casual tourist
who wants to know when the next eruption will occur.

The difference between intercepts is quite a bit larger, at about 0.4, but
again the standard errors are at least several times larger.

5. Answer:

# Do a nonparametric kernel regression of waiting on duration
regression5 = npreg(waiting ~ duration, data = geyser, residuals = T)

# Plot the data and then plot the regression results
plot(geyser$duration, geyser$waiting, cex = 0.5, pch = 16,
     main = "Waiting time as function of geyser duration
            with a kernel regression curve",
     xlab = "Duration (minutes)", ylab = "Waiting time (minutes)"
)
# Pair each observation, indexed by duration, with its fitted value
fits = cbind(geyser$duration, regression5$mean)
# Order the observation-fit pairs by duration (column 1)
fits5 = fits[order(fits[,1]),]
# Add the fits to the plot
lines(fits, col=2, lwd = 2.5)

See Figure 3. The kernel regression (the red curve) is not restricted to
be linear, resulting in a curvy fit. The kernel regression suggests that the
dependence of waiting on duration is rather flat for duration between 0
and 3 minutes, and then decreases sharply for larger values of duration.

6. Answer:
Waiting time as function of geyser duration
with a kernel regression curve

Figure 3:
# Do a kernel regression of the squared residuals (from part 5) on duration
reg.var6 = npreg(regression5$resid^2 ~ geyser$duration)

# Pair each observation, indexed by duration, with its fitted value
fits = cbind(geyser$duration, reg.var6$mean)

# Order the observation-fit pairs by duration (column 1)
fits6 = fits[order(fits[,1]),]

# Plot the fits as a blue dashed curve
plot(fits6, type = "l", col = 4, lty = 2, lwd = 2.5,
xlab = "Duration (minutes)",
ylab = "Estimated variance of waiting given duration",
main = "Estimated variance of waiting given duration:"
Solid black is for unweighted linear regression,
and dashed blue is for kernel regression"

# Add to the plot the fits from part 3
lines(fits3, lwd = 2)

See Figure 4. Observe that the dashed blue curve is almost everywhere
(not in the measure theory sense) below the black curve. HOWEVER,
it would be wrong to take this plot as proof that the kernel regression
produces lower-variance estimates. All of our variance estimates here are
biased because we are considering in-sample model fits instead of computing
residuals for out-of-sample data. To do a real comparison of variance
between the two models, we could implement cross-validation.

Plotting in-sample squared residuals allows us to do an approximate com-
parison of heteroskedasticity between models. We see that using kernel
regression has not substantially reduced heteroskedasticity. Prediction
uncertainty for waiting is very much a function of duration.

7. Answer:

require(np)

# Produce a bandwidth specification
bw = npcdensbw(waiting~duration, data = geyser, tol=.1, ftol=.1)

# Produce a conditional density estimate and plot it
npc.dens7 = npcdens(bws = bw)
npplot(bws=bw, theta = -30, phi = 30, view = "fixed",
main = "Conditional density estimate for waiting given duration")

See Figure 5.

8. Answer: To the extent that conditional means match up with conditional
modes, the projection of the “maximal ridge” (running from the left to the
right in Figure 5) onto the waiting-duration plane should approximately
match up with kernel regression curve in Figure 3.

In comparing Figure 5 to the variance estimate plotted as the dashed blue
curve in Figure 3 notice that low variance estimates correspond to narrow,
Estimated variance of waiting given duration:
Solid black is for unweighted linear regression, and dashed blue is for kernel regression

Figure 4:
Conditional density estimate for waiting given duration

Figure 5:
peaked conditional distributions in Figure 5. This feature is especially noticeable at extreme values of duration.

9. Answer: We have two criteria for sorting out good models from bad ones. The first objective is to produce estimates of waiting times with low bias. The second objective is to minimize the error margin needed for, say, 90% coverage of observation times. Without a strong personal connection with the Oracle of Regression, there is no way to definitively identify the model that best achieves either of these objectives. Even if we identify the true regression function, it will not necessarily be the function with the smallest error margin.

I recommend that the Park Service use the kernel regression model for producing estimates for waiting times. I prefer the kernel regression because it allows for non-linearity in the observations, which don’t look very linear to me.

To produce a margin of error with approximately 90% coverage, one could use leave-one-out cross-validation to sample from the space of residuals for the kernel regression model and then estimate the 90th percentile of absolute residual values, taking this as an error margin.

If the Park Service wants to report separate error margins for each prediction, one could use the estimated conditional distributions of waiting times, discarding the 5% tails.