

Problem Set 1

1. The data were generated in a way that the response is linearly and additively related to the covariates. The linear regression output implies linearity in the coefficients, but the CART output implies a complicated step functions that includes several interaction effects as well as main effects. When running a simple linear regression model of the observed versus predicted outcomes using `lm()` and `rpart()`, we see that the fits are roughly the same. But CART can be misleading when $f(X)$ is linear and additive because CART constructs step functions and interaction effects whether they are appropriate not. It can gets the $f(X)$ very wrong. Summary output from the linear regression model and a regression tree are shown below.

```
> summary(out.lm)
```

```
Call:
```

```
lm(formula = y1 ~ x1 + x2)
```

```
Residuals:
```

	Min	1Q	Median	3Q	Max
	-5.43282	-1.43852	-0.07482	1.19843	5.45674

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.8976	0.1122	8.003	2.76e-14 ***
x1	1.9696	0.1077	18.290	< 2e-16 ***
x2	2.9279	0.1085	26.995	< 2e-16 ***

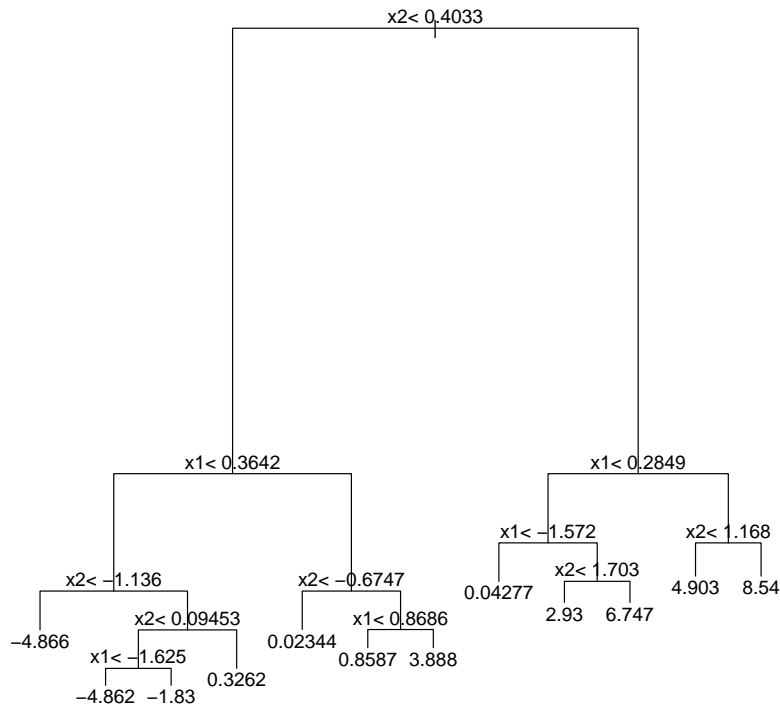
```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

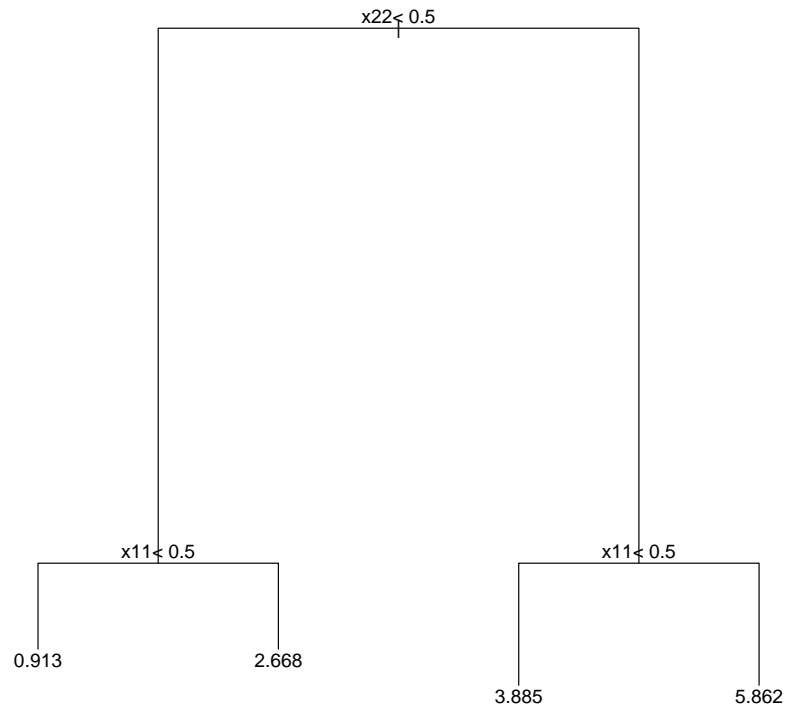
```
Residual standard error: 1.938 on 297 degrees of freedom
```

```
Multiple R-Squared: 0.762, Adjusted R-squared: 0.7604
```

```
F-statistic: 475.6 on 2 and 297 DF, p-value: < 2.2e-16
```



2. When $f(X)$ is a step function and additive, the procedures identify step functions with nearly identical fits. Given the way that the data were generated, CART and linear regression with main effects are essentially equivalent.



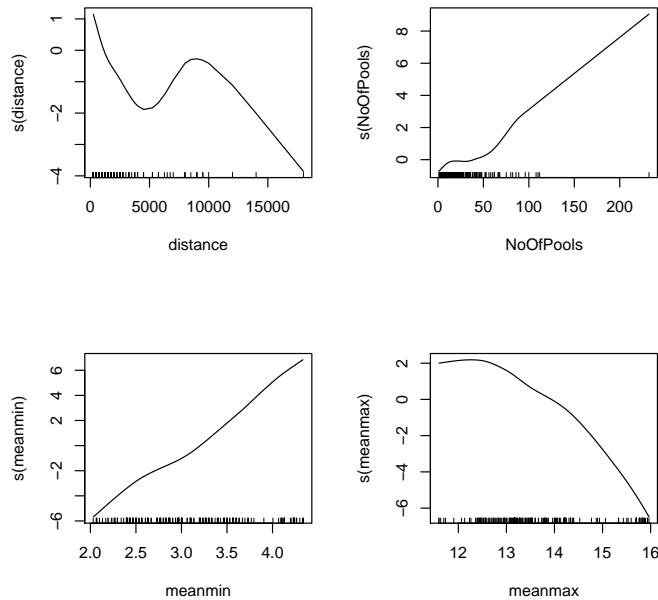
3. CART is a data-adaptive procedure, and as such, it can inductively arrive at step function main effects and/or multi-way interaction effects. If nature generates the data in a way such that these kinds of relationships exist between the response and predictors, CART is likely to perform better than linear regression. It will fit the data better and produce a $f(X)$ that likely to capture better how the data were generated. Using linear regression, it is feasible to consider step functions and interactions, but they need to be specified before the analysis begins or they must be gradually induced by trial and error.

Problem Set 2

1. The following predictors seem to matter: *distance*, *NoOfPools*, and *meanmin*. Recognizing that these relationships in no way imply causality, it makes sense that frogs will be more likely to exist in areas that are close to extant populations of frogs. As the number of potential breeding pools increases, it is reasonable to expect the odds of frog presence to increase as well. The positive sign for the *meanmin* coefficient suggests that there is a positive association between the presence of frogs and the mean minimum Spring temperature. In practice, it would be necessary to confirm that this is sensible, as with all of the predictor coefficients.
2. In addition to *distance*, *NoOfPools*, and *meanmin*, *meanmax* also turns out to be important when applying `stepAIC()` to identify important predictors. The signs of *distance*, *NoOfPools*, and *meanmin* are the same as in (1) above. This sign of *meanmax* is negative. Broadly speaking, the coefficients corresponding to *meanmin* and *meanmax* suggest that the presence of frogs is more likely in warm – but not too warm – Spring temperatures. Again, further investigation is needed to confirm that the coefficients’ signs are sensible.
3. The overall error (i.e., fraction of cases classified incorrectly) is $(38 + 10)/(123 + 41 + 38 + 10) = 22.6\%$. Roughly half of the true presences are classified incorrectly $38/(38 + 41) = 48.1\%$, and $10/(123 + 10) = 7.5\%$ of true absences are identified incorrectly. The rationale behind using 0.5 as a threshold for class assignment is to impose equal (1 to 1) costs of misclassifying the presence and absence of frogs.

	0 (predicted)	1 (predicted)	Model Error
0 (observed)	123	10	.08
1 (observed)	38	41	.48
Use Error	.24	.20	Overall Error = .23

4. Comparing results from GLM subject to AIC and GAM, we see that the residual deviance is lower when using GAM (179.9 versus 199.6) but slightly higher AIC (209.6 versus 213.9). The extra degrees of freedom used by GAM does not seem worth it. Unlike the GLM summary output, the summary GAM output does not include coefficients. The smoother plots show nonlinear relationships between the response and *distance* and *meanmax*. With respect to *distance*, the probability of frog presence decreases until around 5000 meters, at which the presence increases until around 9000 meters. For distances greater than 9000 meters, the probability decreases once again, though this decrease is based on a small number of observations. With respect to *meanmax*, we see a concave-down half-parabola. The *NoOfPools* smoother suggests a positive, piecewise relationship, with three noticeable changes in slope. The relationship between the presence of frogs and *meanmin* is approximately linear. But overall, the non-linear plots would be pretty well summarized by straight lines. So, it again becomes a question of whether the nonlinear relationships are substantively sensible and with the trouble.



5. The overall error (i.e., fraction of cases classified incorrectly) is $(9 + 31)/(124 + 9 + 31 + 48) = 18.9\%$. Roughly half of the true presences are classified incorrectly $9/(124 + 9) = 6.8\%$, and 39.2% $31/(48 + 31)$ of true absences are identified incorrectly. Compared to the confusion table

using GLM and still using 1 to 1 costs, all misclassification errors are lower when using GAM. But the improvement is not dramatic. Again, the fits GLM and GAM are quite similar. The choice between them would be significantly affected by substantive issues.

	0 (predicted)	1 (predicted)	Model Error
0 (observed)	124	9	.07
1 (observed)	31	48	.39
Use Error	.20	.16	Overall Error = .19

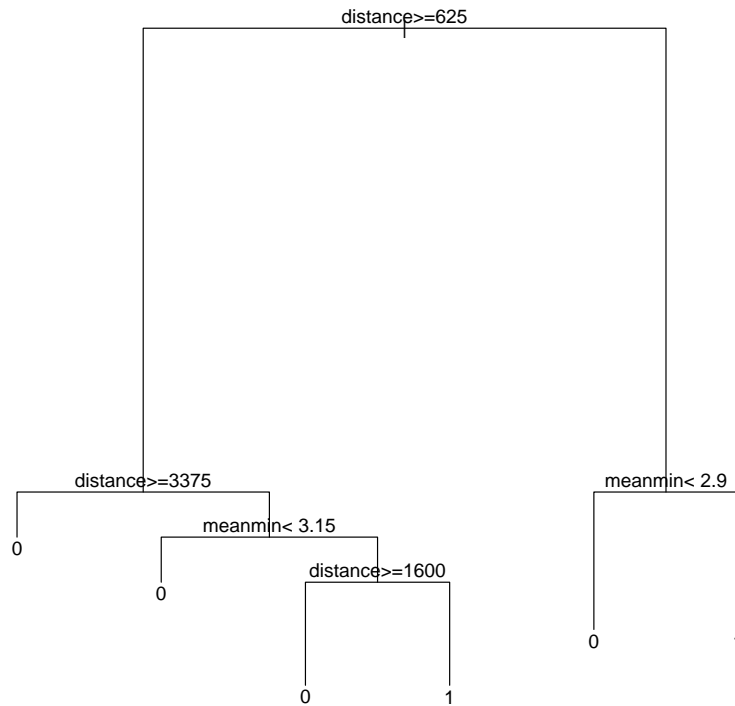
6. For each split you get the variable on which the partition is constructed, the number of observations in the node, the number of misclassified observations, the fitted values of the response and the associated probabilities. For example, the second partition is based on distance greater to or equal to 625, the sample size the node is 137, 28 observations are misclassified, the fitted response value is 0, and the probabilities for the two response values are .796 and .204. Terminal nodes are indicated with asterisks.

The predictor *distance* appears to be important. Of the five splits, *distance* is the splitting variable three times. It is also the basis for the first split. The variable *meanmin* is also important, as it is used to split the data twice. Interestingly, *meanmax* and *NoOfPools* do not appear in the CART model. Just as in the generalized additive model and generalized linear model, there is a negative association between distance and the presence of frogs, and there is a positive association between the response and the mean minimum spring temperature.

```
> print(frogs.rpart)
n= 212

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 212 79 0 (0.62735849 0.37264151)
 2) distance<=625 137 28 0 (0.79562044 0.20437956)
   4) distance<=3375 30 1 0 (0.96666667 0.03333333) *
   5) distance< 3375 107 27 0 (0.74766355 0.25233645)
     10) meanmin< 3.15 76 14 0 (0.81578947 0.18421053) *
     11) meanmin>=3.15 31 13 0 (0.58064516 0.41935484)
       22) distance<=1600 15 2 0 (0.86666667 0.13333333) *
       23) distance< 1600 16 5 1 (0.31250000 0.68750000) *
 3) distance< 625 75 24 1 (0.32000000 0.68000000)
   6) meanmin< 2.9 12 2 0 (0.83333333 0.16666667) *
   7) meanmin>=2.9 63 14 1 (0.22222222 0.77777778) *
```



7. Using CART, 17.9% of cases are misclassified. Classification is more accurate in identifying the absence of frogs (14.3% versus 24.1%). Among the three procedures explored in this problem set, the overall error is lowest when employing CART (22.6% using GLM and 18.9% using GAM). The false absence rate is lowest using GAM (6.7%), and the false presence rate is lowest using CART (24.1%). In terms of misclassification rates, GLM performs the least favorable. In general, we obtain different results because each procedure employs distinct model construction criteria. GLM is the least flexible procedure, GAM is more flexible, and CART is the most flexible among the three methods. Some caution is needed, however, in drawing any conclusions about which procedures works best. We are looking how well the model fits the data and other things equal, more flexible procedures will fit better. One has to worry about overfitting with more flexible models.

	0 (predicted)	1 (predicted)	Model Error
0 (observed)	114	19	.14
1 (observed)	19	60	.24
Use Error	.14	.24	Overall Error = .18

8. In #7 above, we see that the ratio of falsely predicted absences to falsely predicted presences is exactly 1 to 1. There, the unaltered prior is the empirical distribution of the response (62.7 percent absence, 37.3 percent presence), and the overall error is approximately 18 percent.

The confusion tables using an altered prior presence of 0.5 and 0.3, respectively, are shown below.

	0 (predicted)	1 (predicted)	Model Error
0 (observed)	105	28	.21
1 (observed)	16	63	.20
Use Error	.13	.31	Overall Error = .21

Using balanced priors, the above confusion table shows a higher overall error rate and a higher model error rate among locations with no frogs present. At the same time, there are now fewer falsely predicted absences (16 instead of 19) and more correctly predicted presences (63 instead of 60). This is because frog presence is given additional weight in the prior (up nearly 13 percent from the empirical amount).

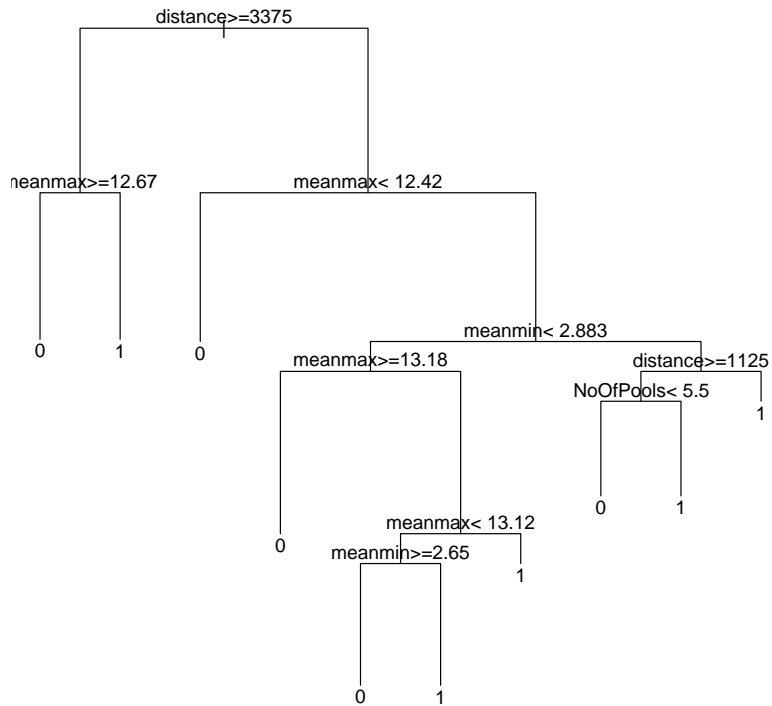
	0 (predicted)	1 (predicted)	Model Error
0 (observed)	112	21	.16
1 (observed)	13	66	.16
Use Error	.10	.24	Overall Error = .16

When the priors are changed such that the rate of frog presence is .3, the number of falsely predicted absences increases, but so too does the number of true absences. In comparison to both #7 and the first CART confusion table in this problem, frog absence is given additional weight (albeit a small increase from the empirical rate of 63 percent).

9. When false negatives are considered 10 times more costly than false positives, the false negative rate jumps to 53%, but the false positive rate goes to 0. All 79 locations with frogs are identified correctly. From the tree diagram, we see more splits in the data. The *meanmax* is now in the model; it is used to split the data four times. This predictor is needed to misclassify the presence of frogs less frequently. The predictors *distance* and *meanmin* are also used, though distance is now used to split the data

two times (compared to three times with 1 to 1 costs). *NoOfPools* is also in the tree once, whereas with 1 to 1 costs, it was not in the model as well.

	0 (predicted)	1 (predicted)
0 (observed)	62	71
1 (observed)	0	79



10. (Tree structures may vary because sampling is involved) The two trees based on bootstrap samples show distinct models, and each of these are different than the original tree diagram shown in question (6). Each tree has a distinct set of predictors used, as well as different splitting locations. These results suggest that CART models can be too sensitive to idiosyncratic features of the data on hand. Trees can differ substantially across random samples from the same population, which represent instability in CART results and is a manifestation of overfitting.

```

> print(frogs.sample1.rpart) #First sample
n= 212

node), split, n, loss, yval, (yprob)
* denotes terminal node

```

```

1) root 212 87 0 (0.5896226 0.4103774)
2) distance>=625 138 35 0 (0.7463768 0.2536232)
4) NoOfPools< 84 127 25 0 (0.8031496 0.1968504)
8) meanmin< 3.216667 93 11 0 (0.8817204 0.1182796)
16) meanmin< 2.483333 35 0 0 (1.0000000 0.0000000) *
17) meanmin>=2.483333 58 11 0 (0.8103448 0.1896552)
34) altitude< 1665 51 6 0 (0.8823529 0.1176471) *
35) altitude>=1665 7 2 1 (0.2857143 0.7142857) *
9) meanmin>=3.216667 34 14 0 (0.5882353 0.4117647)
18) distance>=1625 18 0 0 (1.0000000 0.0000000) *
19) distance< 1625 16 2 1 (0.1250000 0.8750000) *
5) NoOfPools>=84 11 1 1 (0.0909091 0.9090909) *
3) distance< 625 74 22 1 (0.2972973 0.7027027)
6) altitude>=1610 9 1 0 (0.8888889 0.1111111) *
7) altitude< 1610 65 14 1 (0.2153846 0.7846154)
14) avrain< 126.6667 8 2 0 (0.7500000 0.2500000) *
15) avrain>=126.6667 57 8 1 (0.1403509 0.8596491) *

```

```

> print(frogs.sample2.rpart) #Second Sample
n= 212

```

```

node), split, n, loss, yval, (yprob)
* denotes terminal node

```

```

1) root 212 73 0 (0.65566038 0.34433962)
2) distance>=875 112 10 0 (0.91071429 0.08928571) *
3) distance< 875 100 37 1 (0.37000000 0.63000000)
6) meanmin< 3.116667 34 11 0 (0.67647059 0.32352941)
12) NoOfPools< 60 27 5 0 (0.81481481 0.18518519) *
13) NoOfPools>=60 7 1 1 (0.14285714 0.85714286) *
7) meanmin>=3.116667 66 14 1 (0.21212121 0.78787879)
14) distance>=625 13 5 0 (0.61538462 0.38461538) *
15) distance< 625 53 6 1 (0.11320755 0.88679245) *

```

11. (The tree structures that result may vary so the details of the answer may vary as well.) Setting the minimum terminal node size to 50, there is just one split in each tree on *distance*. By that criterion, the trees are the same. So, smaller trees can produce more stable results for a given sample size. But in this case, some differences across trees remain. The full data set and one of the samples split on a distance of 625, while the other sample splits on a distance of 875. In short, overfitting may still be a problem insofar as there is an important substantive difference between a split at 625 and a split at 875.

```

#Population of frog data, min. terminal node size of 50

```

```

> print(frogs.rpart4)
n= 212

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 212 79 0 (0.6273585 0.3726415)
  2) distance>=625 137 28 0 (0.7956204 0.2043796) *
  3) distance< 625 75 24 1 (0.3200000 0.6800000) *

#Sample with replacement of frog data, min. terminal node size of 50
> print(frogs.sample1.rpart2)
n= 212

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 212 87 0 (0.5896226 0.4103774)
  2) distance>=625 138 35 0 (0.7463768 0.2536232) *
  3) distance< 625 74 22 1 (0.2972973 0.7027027) *

#Another sample with replacement of frog data, min. terminal node size of 50
> print(frogs.sample2.rpart2)
n= 212

node), split, n, loss, yval, (yprob)
      * denotes terminal node

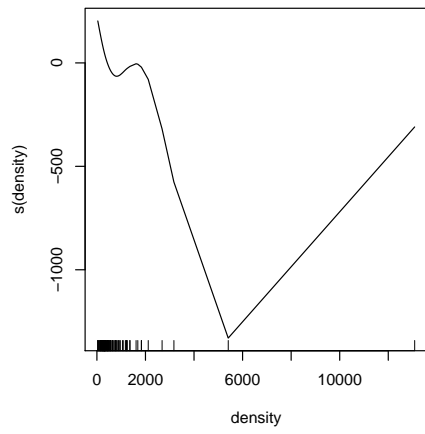
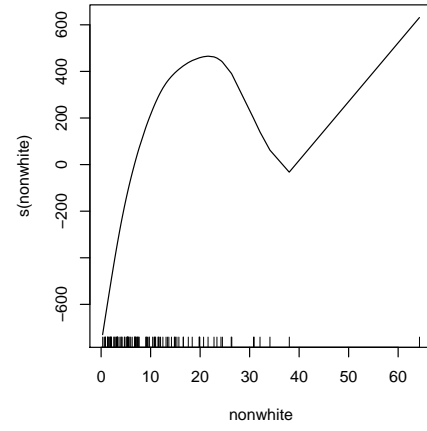
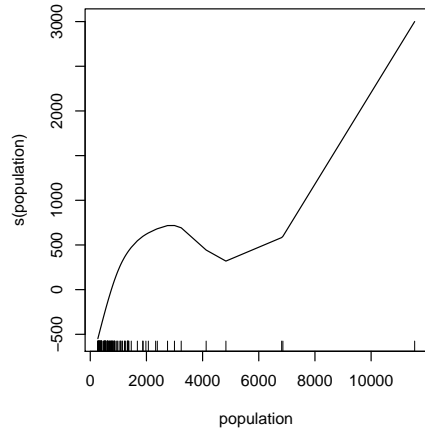
1) root 212 73 0 (0.65566038 0.34433962)
  2) distance>=875 112 10 0 (0.91071429 0.08928571) *
  3) distance< 875 100 37 1 (0.37000000 0.63000000) *

```

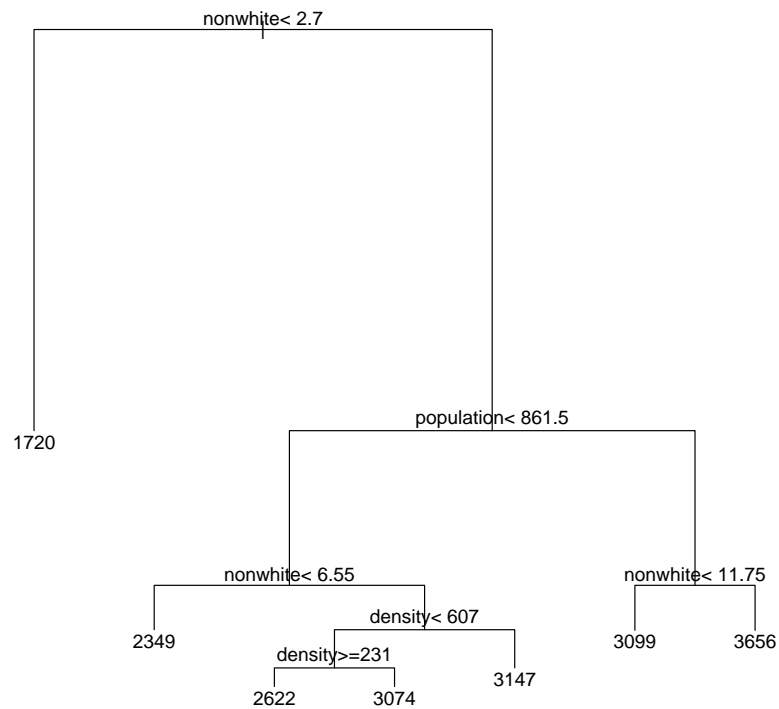
Problem Set 3

1. The GAM plots show nonlinear relationships between the crime rate and each of the predictors, as well as sharp inflection points. However, most of the dramatic nonlinearities occur where there is not much data. Support is very sparse in those regions (look at the rug plots). It might well be prudent to ignore the fitted values in those regions.

But taken at face value, as the population increases, so too does the crime rate, with a small dip around 2.5 million people. As the proportion of non-whites increases, the crime rate increases as well until roughly 20 percent, at which point there is a decrease in crime rate until 35-40 percent. After that the relationship is again positive. Finally, as population density increases, the crime rate decreases for population densities less than roughly 6,000 people per square mile. For higher densities, the association between crime rate and density is positive.



2. In both CART and GAM, it is possible to identify nonlinearity between crime rate and the predictors. However, an advantage to using CART is that it is easier to identify interaction effects as well (such can be done using GAM only for a small number of dimensions). As the proportion of non-whites increases, the crime rate increases as well according to the tree diagram. As the population increases, so too does the crime rate. What we could not see from the GAM plots is that there is an interaction effect between *nonwhite*, *population*, and *density*. For a specific subset of the data, a density greater than 607 suggests a higher crime rate. Still, one must keep in mind the fact that this terminal node (with a crime rate of 3147 crimes per hundred thousand people) is based on only eight metropolitan areas.



3. If the fitted values corresponded perfectly, we would see a straight line of CART predictions against GAM predictions, with a slope of 1.0 and an intercept of 0.0. Given that CART yields seven distinct predicted values – one for each terminal node – whereas GAM produces many distinct values, we would see two things if the fitted values were identical across procedures: 1) a straight line with a slope of 1 and intercept of 0, and 2) the GAM predictions would “straddle” the CART predictions so that the predicted values were, on average, the same. While we see a strong, positive relationship between the two sets of predictions, the distribution of each terminal node’s GAM predictions tends to be on one side or the other of the (dashed) 45-degree line. For larger fitted values, the points tend to be to the left of the 1 to 1 line. GAM is more likely to produce smaller fitted values than CART. For smaller fitted values the reverse is true. Any given city could have two rather different fitted values depending on which procedure was used. Whether this matters would be depend on the use to be made of the results.
4. (See figure in question (3)) If the slope were 1.0 and intercept were 0.0, this would indicate that the two procedures are equivalent (at least for this data set). Because the actual slope and intercept are relatively close to the 45-degree line, the overlaid regression line indicates that, on average, the two produce very similar predictions. From the previous question, however, there could be important differences between the two procedures for a given city.
5. The four scatter plots include smoothers with spans of $1/5$, $1/3$, $2/3$, and $4/5$. Each of these show a positive relationship. Smaller spans lead to a rougher smooth, but the overall story is about the same for each. For the regions where there are data, the fit is quite linear. The conclusions from the previous question are still sound.
6. Among `rpart()` and `gam()`, the `gam()` procedure produces fitted values that are correlated slightly higher with the observed crime rates (0.669 versus 0.632). The correlation is an instructive statistic for measuring which modeling approach is superior if a) the sum of the squared residuals is the “gold standard” for the quality of fit, and b) the objective is entirely descriptive, and some overall measure of fit is what matters, not the fitted for any particular city or small set of cities. The same hold for the square of the correlation is R^2 . These statistics do not take into account model complexity, prediction error, and performance with test data sets. In short, having a high correlation between the fitted and the observed values is often a good thing, hardly the full story.

