Assignment

Write Up:

Data 621 Group 2 HW 3: Crime

Members: Omar Pineda, Jeff Littlejohn, Sergio Ortega Cruz, Chester Poon, Simon Ustoyev

Due: October 30, 2019

Assignment

Build a binary logistic regression model on the training data set to predict whether the neighborhood will be at risk for high crime levels. You will provide classifications and probabilities for the evaluation data set using your binary logistic regression model. You can only use the variables given to you (or variables that you derive from the variables provided). Use 0.5 threshold. Below is a short description of the variables of interest in the data set:

- zn: proportion of residential land zoned for large lots (over 25000 square feet) (predictor variable)
- indus: proportion of non-retail business acres per suburb (predictor variable)
- chas: a dummy var. for whether the suburb borders the Charles River (1) or not (0) (predictor variable)
- nox: nitrogen oxides concentration (parts per 10 million) (predictor variable)
- rm: average number of rooms per dwelling (predictor variable)
- age: proportion of owner-occupied units built prior to 1940 (predictor variable)
- dis: weighted mean of distances to five Boston employment centers (predictor variable)
- rad: index of accessibility to radial highways (predictor variable)
- tax: full-value property-tax rate per $10,000 (predictor variable)
- ptratio: pupil-teacher ratio by town (predictor variable)
- black: 1000(Bk - 0.63)2 where Bk is the proportion of blacks by town (predictor variable)
- lstat: lower status of the population (percent) (predictor variable)
- medv: median value of owner-occupied homes in $1000s (predictor variable)
- target: whether the crime rate is above the median crime rate (1) or not (0) **(response variable)**

Write Up:

1. Data Exploration

The dataset includes information on 466 neighborhoods in the city of Boston. Despite its East Coast location and reputation as a bastion of liberalism, Boston is among the most racially segregated of American cities. Attempts to integrate the schools using busing in the 1970s led to sustained violence (https://en.wikipedia.org/wiki/Boston_desegregation_busing_crisis), including deaths. Recent scholarship has highlighted the widespread use of redlining, a process by which institutions such as banks refused to offer mortgages or other financial services to people of certain races if they wished to purchase a home in certain neighborhoods despite creditworthiness.

In short, one would probably not want to construct a model to predict crime by neighborhood that uses variables such as race without having a clear idea of the model's intended use and an ethical framework for evaluating said model. This, however, is an academic exercise, so we proceed. Let's preview the data.

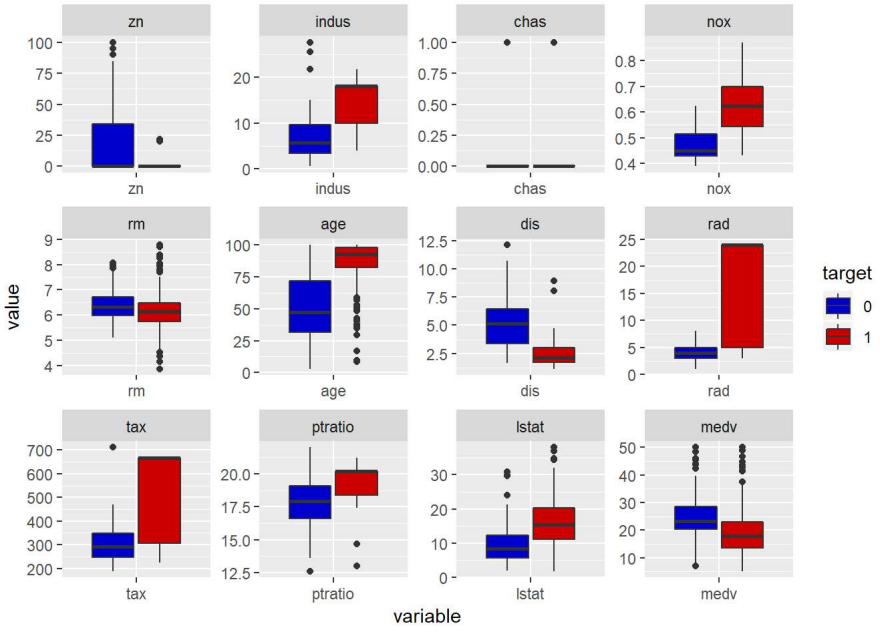
zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv	target
0	19.58	0	0.605	7.929	96.2	2.0459	5	403	14.7	3.70	50.0	1
0	19.58	1	0.871	5.403	100.0	1.3216	5	403	14.7	26.82	13.4	1
0	18.10	0	0.740	6.485	100.0	1.9784	24	666	20.2	18.85	15.4	1
30	4.93	0	0.428	6.393	7.8	7.0355	6	300	16.6	5.19	23.7	0
0	2.46	0	0.488	7.155	92.2	2.7006	3	193	17.8	4.82	37.9	0
0	8.56	0	0.520	6.781	71.3	2.8561	5	384	20.9	7.67	26.5	0
0	18.10	0	0.693	5.453	100.0	1.4896	24	666	20.2	30.59	5.0	1
0	18.10	0	0.693	4.519	100.0	1.6582	24	666	20.2	36.98	7.0	1
0	5.19	0	0.515	6.316	38.1	6.4584	5	224	20.2	5.68	22.2	0
80	3.64	0	0.392	5.876	19.1	9.2203	1	315	16.4	9.25	20.9	0
22	5.86	0	0.431	6.438	8.9	7.3967	7	330	19.1	3.59	24.8	0
0	12.83	0	0.437	6.286	45.0	4.5026	5	398	18.7	8.94	21.4	0
0	18.10	0	0.532	7.061	77.0	3.4106	24	666	20.2	7.01	25.0	1
22	5.86	0	0.431	8.259	8.4	8.9067	7	330	19.1	3.54	42.8	1
0	2.46	0	0.488	6.153	68.8	3.2797	3	193	17.8	13.15	29.6	0

Expected variables are present. Note that, as indicated in the variables' descriptions, many of the variables have already been scaled or transformed in some way. Let's calculate summary statistics and generate a box plot for further review.

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	
zn	1	466	11.5772532	23.3646511	0.00000	5.3542781	0.0000000	0.0000	100.0000	100.0000	2.1768152	3.8135765	1.0823
indus	2	466	11.1050215	6.8458549	9.69000	10.9082353	9.3403800	0.4600	27.7400	27.2800	0.2885450	-1.2432132	0.3171
chas	3	466	0.0708155	0.2567920	0.00000	0.0000000	0.0000000	0.0000	1.0000	1.0000	3.3354899	9.1451313	0.0118
nox	4	466	0.5543105	0.1166667	0.53800	0.5442684	0.1334340	0.3890	0.8710	0.4820	0.7463281	-0.0357736	0.0054
rm	5	466	6.2906738	0.7048513	6.21000	6.2570615	0.5166861	3.8630	8.7800	4.9170	0.4793202	1.5424378	0.0326
age	6	466	68.3675966	28.3213784	77.15000	70.9553476	30.0226500	2.9000	100.0000	97.1000	-0.5777075	-1.0098814	1.3119
dis	7	466	3.7956929	2.1069496	3.19095	3.5443647	1.9144814	1.1296	12.1265	10.9969	0.9988926	0.4719679	0.0976
rad	8	466	9.5300429	8.6859272	5.00000	8.6978610	1.4826000	1.0000	24.0000	23.0000	1.0102788	-0.8619110	0.4023
tax	9	466	409.5021459	167.9000887	334.50000	401.5080214	104.5233000	187.0000	711.0000	524.0000	0.6593136	-1.1480456	7.7778
ptratio	10	466	18.3984979	2.1968447	18.90000	18.5970588	1.9273800	12.6000	22.0000	9.4000	-0.7542681	-0.4003627	0.1017
lstat	11	466	12.6314592	7.1018907	11.35000	11.8809626	7.0720020	1.7300	37.9700	36.2400	0.9055864	0.5033688	0.3289
medv	12	466	22.5892704	9.2396814	21.20000	21.6304813	6.0045300	5.0000	50.0000	45.0000	1.0766920	1.3737825	0.4280
target	13	466	0.4914163	0.5004636	0.00000	0.4893048	0.0000000	0.0000	1.0000	1.0000	0.0342293	-2.0031131	0.0231

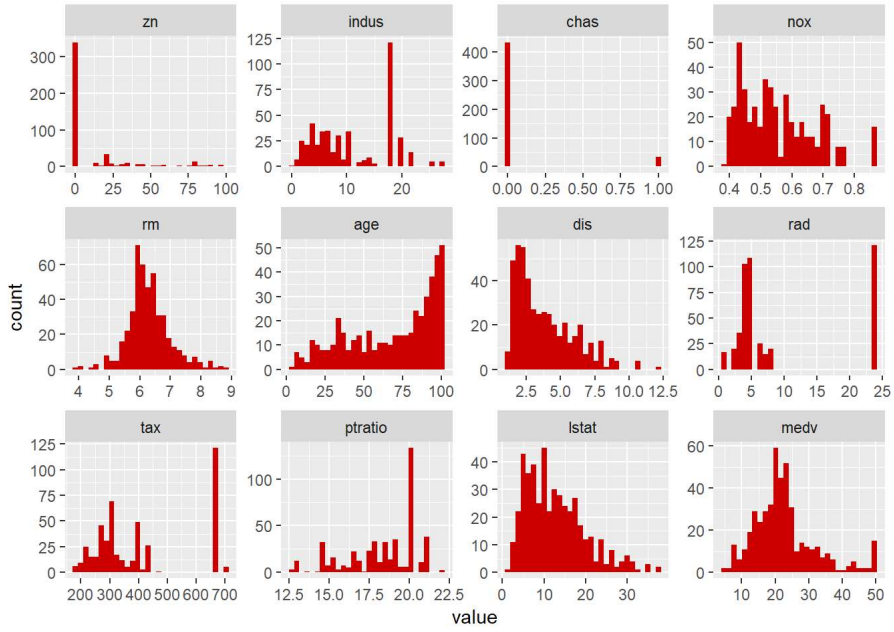
We see 466 records in our training set and no missing values for any variable. Other than that, without a specific question in mind, it's difficult to draw any conclusions from this big table of numbers. We see no missing values that would require imputation using medians or other methods.

Now, we visualize using box plots. We'll separate the box plots by the target value, which signifies whether or not the neighborhood is high crime. And we'll approximate Boston Red Sox colors.



The dummy variable (chas) that represents proximity to the Charles River is not meaningful, but clear distinctions in distributions between the neighborhoods in which the crime rate is below and above the median - the target variable by which the box plots are split. We might later look at these values after transformations such as logs.

To check for skewness, let's examine the distribution of each variable independent of target variable value.

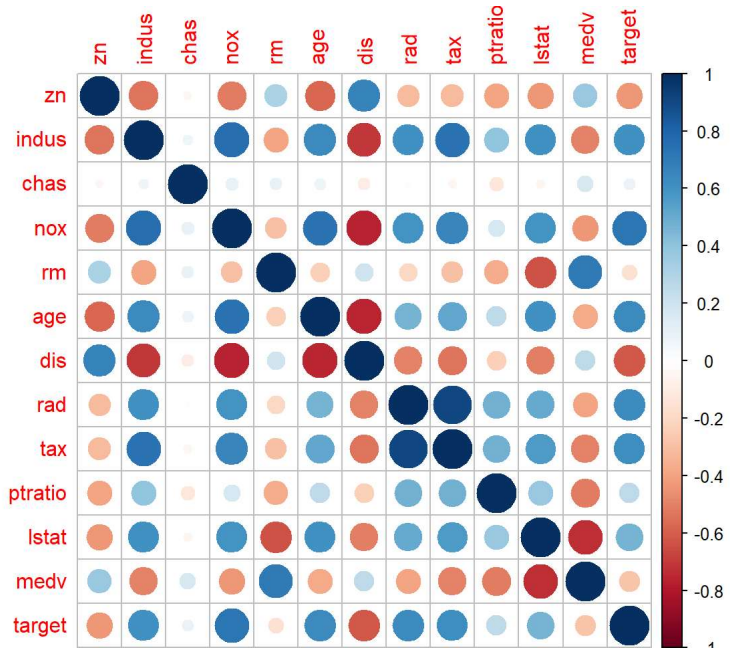


Skewness abounds. We will file this away for now and revisit in the Data Preparation part of the project. In particular, zn, nox, age, dis, ptratio, and lstat seem likely candidates for transformations.

We will now check for covariance.

	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv	t
zn	1.0000000	-0.5382664	-0.0401620	-0.5170452	0.3198141	-0.5725805	0.6601243	-0.3154812	-0.3192841	-0.3910357	-0.4329925	0.3767171	-0.431
indus	-0.5382664	1.0000000	0.0611832	0.7596301	-0.3927118	0.6395818	-0.7036189	0.6006284	0.7322292	0.3946898	0.6071102	-0.4961743	0.604
chas	-0.0401620	0.0611832	1.0000000	0.0974558	0.0905098	0.0788837	-0.0965771	-0.0159004	-0.0467648	-0.1286606	-0.0514232	0.1615653	0.080
nox	-0.5170452	0.7596301	0.0974558	1.0000000	-0.2954897	0.7351278	-0.7688840	0.5958298	0.6538780	0.1762687	0.5962426	-0.4301227	0.726
rm	0.3198141	-0.3927118	0.0905098	-0.2954897	1.0000000	-0.2328125	0.1990158	-0.2084457	-0.2969343	-0.3603471	-0.6320245	0.7053368	-0.152
age	-0.5725805	0.6395818	0.0788837	0.7351278	-0.2328125	1.0000000	-0.7508976	0.4603143	0.5121245	0.2554479	0.6056200	-0.3781560	0.630
dis	0.6601243	-0.7036189	-0.0965771	-0.7688840	0.1990158	-0.7508976	1.0000000	-0.4949919	-0.5342546	-0.2333394	-0.5075280	0.2566948	-0.618
rad	-0.3154812	0.6006284	-0.0159004	0.5958298	-0.2084457	0.4603143	-0.4949919	1.0000000	0.9064632	0.4714516	0.5031013	-0.3976683	0.628
tax	-0.3192841	0.7322292	-0.0467648	0.6538780	-0.2969343	0.5121245	-0.5342546	0.9064632	1.0000000	0.4744223	0.5641886	-0.4900329	0.611
ptratio	-0.3910357	0.3946898	-0.1286606	0.1762687	-0.3603471	0.2554479	-0.2333394	0.4714516	0.4744223	1.0000000	0.3773560	-0.5159153	0.250
lstat	-0.4329925	0.6071102	-0.0514232	0.5962426	-0.6320245	0.6056200	-0.5075280	0.5031013	0.5641886	0.3773560	1.0000000	-0.7358008	0.469
medv	0.3767171	-0.4961743	0.1615653	-0.4301227	0.7053368	-0.3781560	0.2566948	-0.3976683	-0.4900329	-0.5159153	-0.7358008	1.0000000	-0.270
target	-0.4316818	0.6048507	0.0800419	0.7261062	-0.1525533	0.6301062	-0.6186731	0.6281049	0.6111133	0.2508489	0.4691270	-0.2705507	1.000

We see some very high positive and negative correlations between variables. Let's construct a more effective visualization.



We see candidates for combination due to covariance.

As a final step, let's look just a correlation between the independent variables and the target variables.

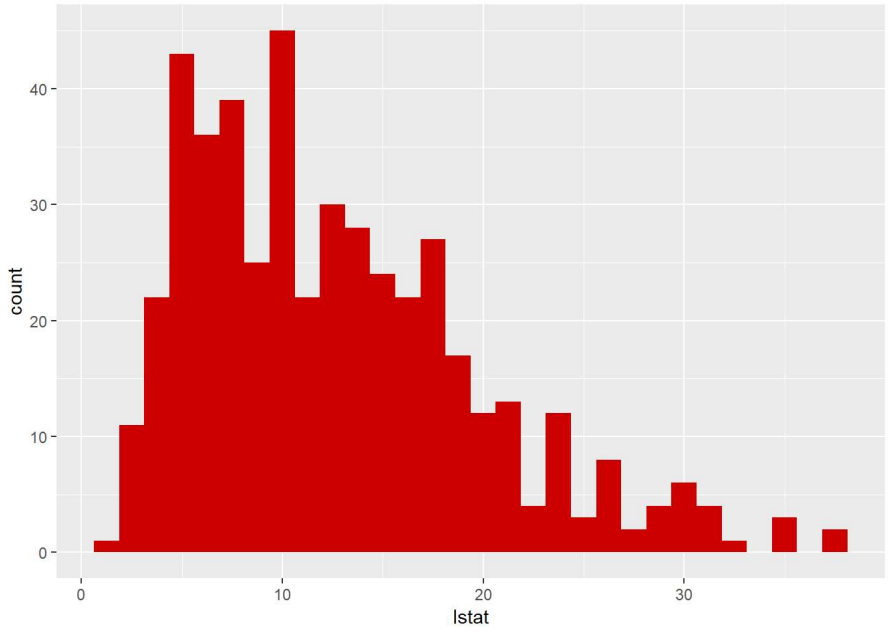
	target
zn	-0.4316818
indus	0.6048507
chas	0.0800419
nox	0.7261062
rm	-0.1525533
age	0.6301062
dis	-0.6186731
rad	0.6281049
tax	0.6111133
ptratio	0.2508489
lstat	0.4691270
medv	-0.2705507
target	1.0000000

Nox, or the concentration of nitrogen oxide, a significant pollutant that's harmful to human health, in a neighborhood, shows the closest correlation with the target variable at .73. Next, age, rad, tax, and indus all correlate with the target value just above .6. Zn showed the largest negative correlation with the target at -.43. Zn represents the percentage of residential lots zoned for large lots, which may be an indicator large rental housing - apartments.

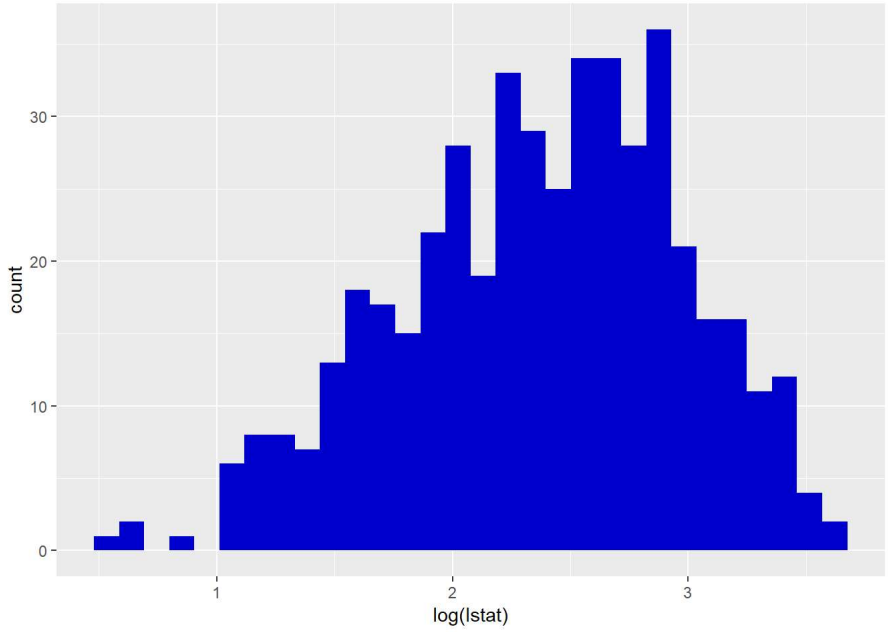
2. Data Preparation

Prior to modelling the training data set, we must prepare the data. We do not have any missing values, so imputation is not required. We will probably actually start with a model that uses all variables regardless of skew or covariance. However, we will definitely progress to using transformations and will also combine variables due to covariance in seeking the construction of accurate and valid models.

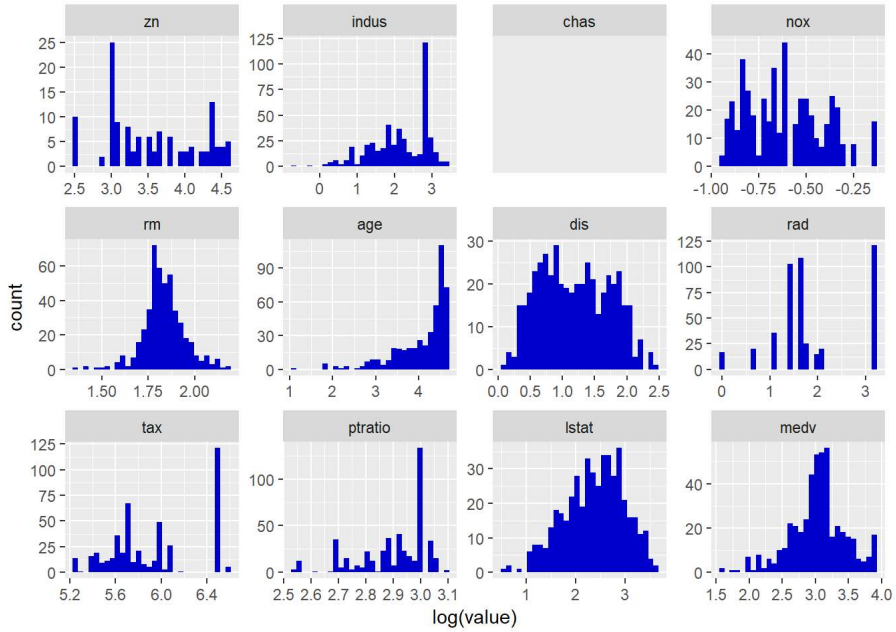
Let's look at how transformations might solve distribution issues with some of our variables. Earlier, we saw a strong right skew in the distribution of the variable lstat, which tracks the "lower status" of a neighborhood's population. Probably not the best phrasing.



What would a log transformation do to this distribution?

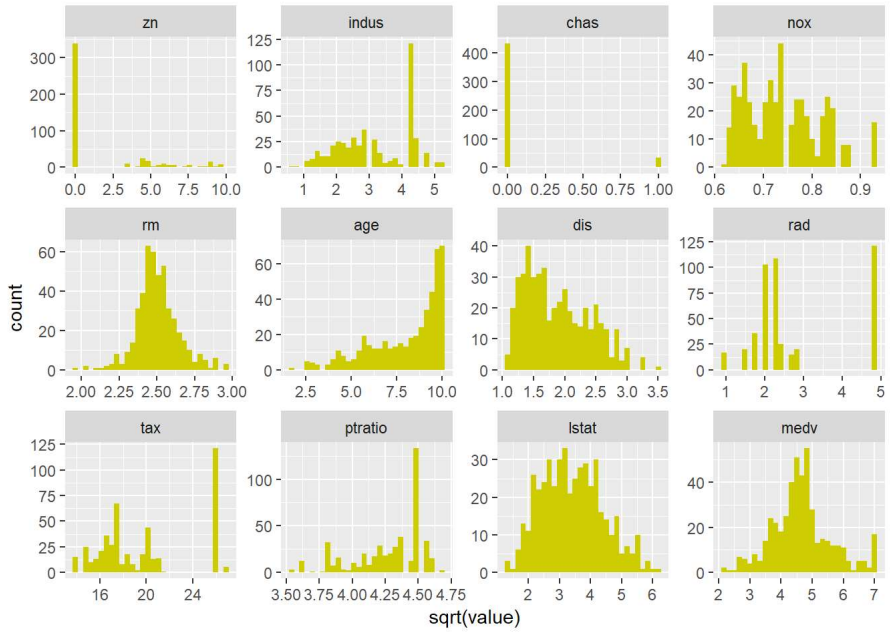


Looks slightly better. Let's generate log transformations for all variables in the dataset.



Medv looks slightly better. However, age remains strongly left skewed. Dis is now bimodal.

What about other transformations such as quadratic ones?



Nope. Not a lot of improvement.

In Part 1, we saw high covariances among variables such as rad and tax (.91). To build the best models, we'll likely want to examine combining some of these variables that are correlated to each other, which tends to increase standard errors. This can lead to overfitting and inefficient models. We will not combine variables here but instead revisit this concept in part 3 when evaluating our models.

Our textbooks have also discussed the possibility of creating bins for continuous variables. For example, dis, the weighted distance of means of distances from a neighborhood to five Boston job centers, might be better suited to fall into three categories than to remain a continuous variable for performance reasons.

Prior to building the model, we're going to split our training data into a true training set and a validation set. We'll go 80/20 training to validation.

3. Build Models

Following convention, we will start with a model consisting of all variables, none of which have been transformed. While we've moved on to Part 3, where we will construct the models, the boundary between data preparation and model building is grey. We will to explore transformations and collinearity.

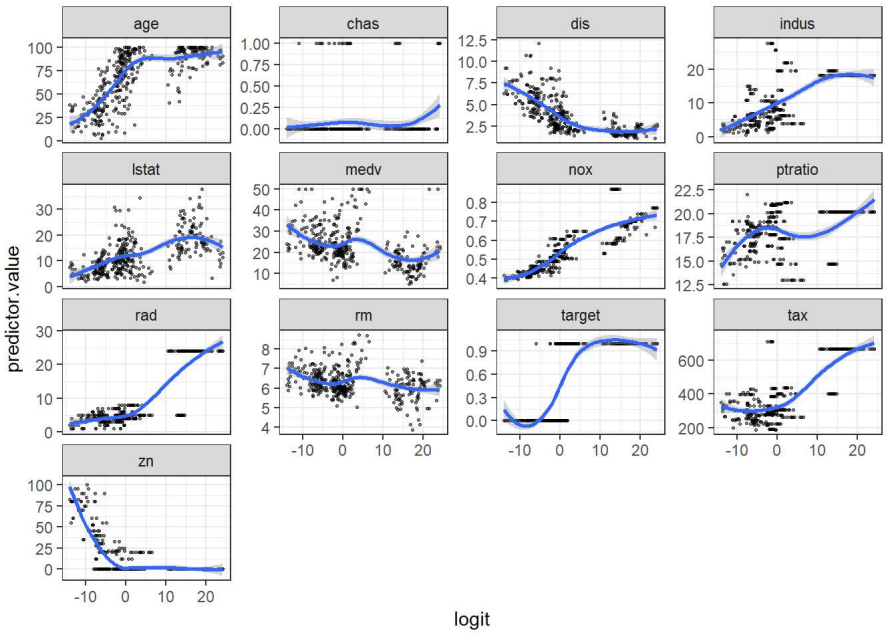
Model 1 - All Variables Untransformed

```
##
## Call:
## glm(formula = target ~ ., family = "binomial", data = crime_train_data_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9392  -0.1315  -0.0024   0.0015   3.4720
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -43.954712   8.008648  -5.488 4.06e-08 ***
## zn           -0.062257   0.039283  -1.585 0.113003
## indus        -0.053255   0.058450  -0.911 0.362232
## chas          0.675370   0.857697   0.787 0.431035
## nox          51.934147   9.821117   5.288 1.24e-07 ***
## rm           -0.519049   0.888956  -0.584 0.559297
## age           0.029116   0.015881   1.833 0.066735 .
## dis           0.766202   0.265472   2.886 0.003899 **
## rad           0.733765   0.194240   3.778 0.000158 ***
## tax          -0.006802   0.003386  -2.009 0.044539 *
## ptratio       0.400174   0.153418   2.608 0.009097 **
## lstat         0.092014   0.063229   1.455 0.145599
## medv          0.215374   0.087839   2.452 0.014210 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 514.63  on 371  degrees of freedom
## Residual deviance: 143.67  on 359  degrees of freedom
## AIC: 169.67
##
## Number of Fisher Scoring iterations: 9
```

Our most significant variables generally tie to the variables we saw have the highest correlations with the target value earlier. We have an AIC of 169.67 and a residual deviance of 143.67.

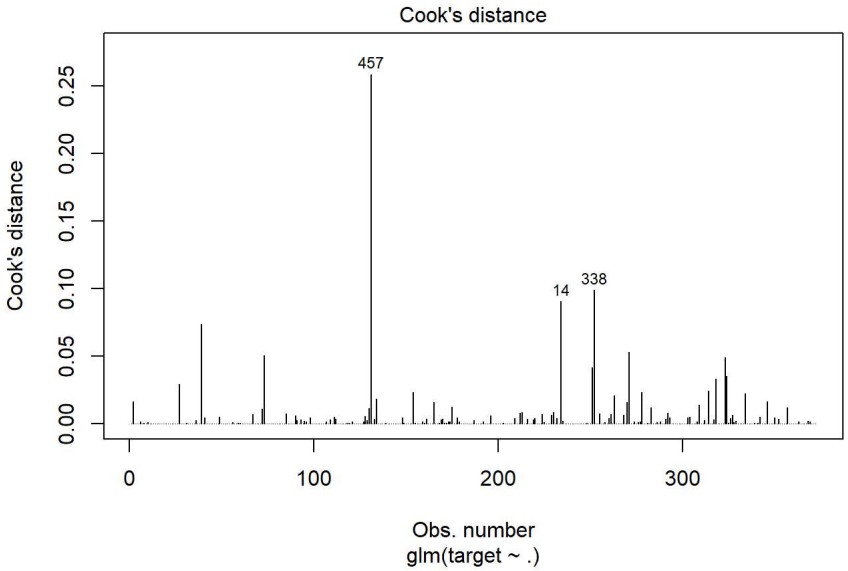
Let's run further diagnostics on the model. We will set a probability of .5 as being the cutoff for determining if a neighborhood will be high crime. Here, we check the relationship between the logit of the outcome and each predictive variable. (Target and the binary dummy variable chas should be ignored.) Again, these steps also could be labelled as data preparation.

```
##   135   367   190   409   435    22
## "neg" "pos" "pos" "pos" "neg" "pos"
```



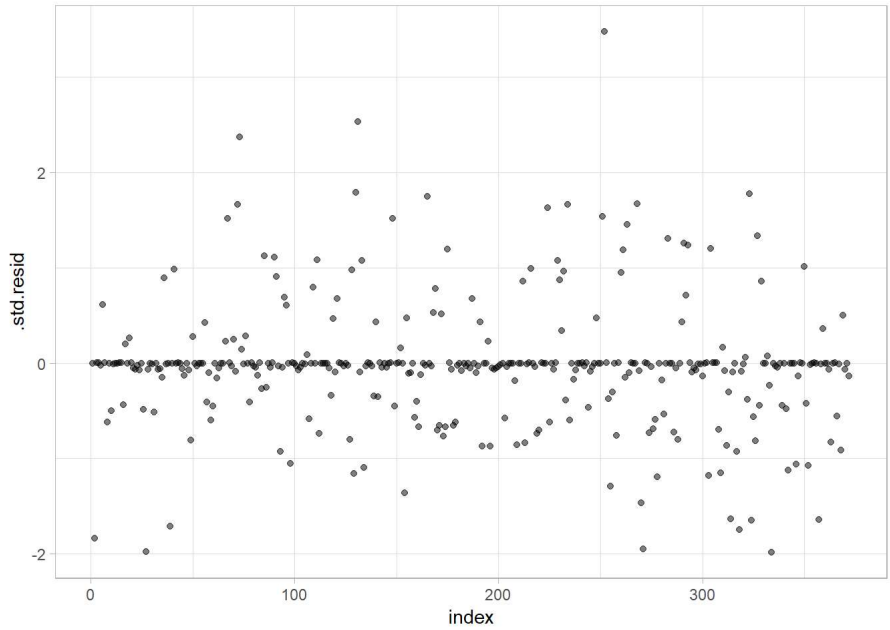
Tax and zn do not show linear associations with the outcomes in logit scale. Along with the previously discussed lstat, they might benefit from transformations.

Let's use Cook's Distance to check for outliers.



.rownames	target	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv	.fitted	.se.fit	.resid	.hat	.sigma	.coo
457	1	0	10.59	0	0.489	5.412	9.8	3.5875	4	277	18.6	29.55	23.7	-2.5803687	1.629657	2.303642	0.1738331	0.6191799	0.2586
14	1	22	5.86	0	0.431	8.259	8.4	8.9067	7	330	19.1	3.54	42.8	-0.3920093	1.194062	1.347768	0.3430954	0.6273771	0.0905
338	1	20	6.96	0	0.464	5.856	42.1	4.4290	3	223	18.6	13.00	21.1	-6.0250580	1.133342	3.472023	0.0030902	0.6062579	0.0989

An outlier is not necessarily influential. Let's check for that.



Let's pull that point that's above 3 standardized residuals from 0.

.rownames	target	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv	.fitted	.se.fit	.resid	.hat	.sigma	.cooksd
338	1	20	6.96	0	0.464	5.856	42.1	4.429	3	223	18.6	13	21.1	-6.025058	1.133342	3.472023	0.0030902	0.6062579	0.0989419

Observation 338 is an influential outlier.

Next, we check multicollinearity.

```
##       zn    indus     chas      nox       rm      age      dis      rad
## 1.881239 3.156811 1.229952 4.960830 6.065538 2.642072 4.089579 1.947943
##      tax  ptratio    lstat     medv
## 2.376876 2.419764 2.723224 8.696936
```

The rule of thumb is that vif scores above 5 should be judged as having a high amount of multicollinearity. So rm and medv have issues in this regard.

In summary, we have:

1. Multiple predictors that do not have linear relationships with the logit of the outcome variable.
2. One influential outlier - index 338.
3. Two predictors with potentially problematically high multicollinearity.

The above are among many methods to check assumptions and diagnostics of logistic regression models. We will not repeat these steps - other than the summary diagnostics - for our additional attempts at constructing a model to predict high-crime neighborhoods.

Model 2 - Collinearity

```
##
## Call:
## glm(formula = target ~ zn + indus + chas + nox + age + dis +
##     rad + tax + ptratio + lstat + medv, family = "binomial",
##     data = crime_train_data_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9586  -0.1457  -0.0028   0.0018   3.4402
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -44.415908   7.905817  -5.618 1.93e-08 ***
## zn           -0.063314   0.038909  -1.627 0.103689
## indus        -0.051556   0.058270  -0.885 0.376276
## chas          0.697782   0.857637   0.814 0.415869
## nox          50.739927   9.490760   5.346 8.98e-08 ***
## age           0.025030   0.014057   1.781 0.074979 .
## dis           0.724497   0.251411   2.882 0.003955 **
## rad           0.711955   0.188502   3.777 0.000159 ***
## tax          -0.007035   0.003359  -2.095 0.036203 *
## ptratio       0.359167   0.134298   2.674 0.007486 **
## lstat         0.105872   0.058173   1.820 0.068764 .
## medv          0.175190   0.052623   3.329 0.000871 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 514.63  on 371  degrees of freedom
## Residual deviance: 144.02  on 360  degrees of freedom
## AIC: 168.02
##
## Number of Fisher Scoring iterations: 9
```

Iniitally for model 2, we removed the variable with the highest collinearity (medv), which led to slightly higher residual deviance and AIC. Medv has a low p-value in the original model, so we instead removed rm, which led to a slight increase in residual deviance but a drop in AIC. Not a big difference.

Model 3 - Partial Log Transformations

Here, we try log transformations of a couple variables that did not show linear relations between them and the logit of the outcome.

```
##
## Call:
## glm(formula = target ~ zn + indus + chas + nox + rm + age + dis +
##     rad + tax + ptratio + log(lstat) + log(medv), family = "binomial",
##     data = crime_train_data_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9146  -0.1649  -0.0048   0.0029   3.3521
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -51.103411  11.370274  -4.494 6.97e-06 ***
## zn           -0.050353   0.034899  -1.443 0.149070
## indus        -0.052091   0.058319  -0.893 0.371739
## chas          0.751968   0.890887   0.844 0.398632
## nox          49.332175   9.664369   5.105 3.32e-07 ***
## rm            0.063858   0.756310   0.084 0.932712
## age           0.030108   0.015184   1.983 0.047387 *
## dis           0.712916   0.252333   2.825 0.004724 **
## rad           0.664907   0.180679   3.680 0.000233 ***
## tax          -0.006223   0.003470  -1.793 0.072946 .
## ptratio       0.346223   0.148787   2.327 0.019967 *
## log(lstat)    0.542533   0.796489   0.681 0.495773
## log(medv)     3.519152   2.010423   1.750 0.080040 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 514.63  on 371  degrees of freedom
## Residual deviance: 149.84  on 359  degrees of freedom
## AIC: 175.84
##
## Number of Fisher Scoring iterations: 9
```

Adding log transformations to lstat and medv has increased both our residual deviance and AIC, which is less than desirable.

Model 4 - Significant Variables

Here, we include only the variables that were signfiicant (p value < .05) fro m model 1.

```
##
## Call:
## glm(formula = target ~ nox + rad + dis + ptratio + medv + tax +
##     age, family = "binomial", data = crime_train_data_train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.97027  -0.17714  -0.01856   0.00296   3.13739
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -38.666602   6.897163  -5.606 2.07e-08 ***
## nox          44.611899   7.789706   5.727 1.02e-08 ***
## rad           0.739014   0.164010   4.506 6.61e-06 ***
## dis           0.487788   0.196070   2.488  0.01285 *
## ptratio       0.366746   0.127129   2.885  0.00392 **
## medv          0.110761   0.041069   2.697  0.00700 **
## tax          -0.008714   0.002875  -3.031  0.00244 **
## age           0.034673   0.013382   2.591  0.00957 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 514.63  on 371  degrees of freedom
## Residual deviance: 152.25  on 364  degrees of freedom
## AIC: 168.25
##
## Number of Fisher Scoring iterations: 9
```

The residual deviance is higher here than in model 1 without a significant drop in AIC.

Model 5 - Using LEAP for forward selection

We're not making good progress here on improving upon model 1, the initial try that included all untransformed variabes. Let's try the Leaps package, which will start with the most significant variable and then continue adding variables/features until many possible models have been evaluated. This is known as Forward selection. Note that while the Leaps pacage executed perfectly in R Studio and allowed us to evaluate numerous possible models with the variables, it caused an error when "knitting."

The LEAPS package provided adjusted R^2 scores for a variety of packages. It started with just the nox variable and then added variables in constructing 12 total models. The 10h model (zn + indus + nox + rm + age+ dis + rad + tax + lstat + medv) from LEAPS has the highest R^2 but the 6th model is only slightly lower in adjusted R^2 with only six variables. In the interest of simplicity, we'll take that 6th model rom LEAPS - i-n-a-rd-t-m, which translates to the below.

```
##
## Call:
## glm(formula = target ~ nox + rad + tax + medv + age + indus,
##     family = "binomial", data = crime_train_data_train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.84034  -0.26538  -0.04395   0.00700   2.73765
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -20.816101   3.518666  -5.916 3.30e-09 ***
## nox          34.004675   7.540052   4.510 6.49e-06 ***
## rad           0.647194   0.157451   4.110 3.95e-05 ***
## tax          -0.008066   0.003092  -2.608   0.0091 **
## medv          0.026737   0.029892   0.894   0.3711
## age           0.018464   0.011659   1.584   0.1133
## indus        -0.024421   0.055520  -0.440   0.6600
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 514.63  on 371  degrees of freedom
## Residual deviance: 165.20  on 365  degrees of freedom
## AIC: 179.2
##
## Number of Fisher Scoring iterations: 8
```

Note that we selected this model from LEAPS based on R^2 but are now comparing it to other models using residual deviance and AIC, where it is found wanting. Deciding whether to use residual deviance, AIC, or adjusted R^2 among other model diagnostics like AICc and BIC is always challenge and may be domain - or maybe even problem - dependent.

Model 6 - Using stepAIC for forward selection

We now looked at using the stepAIC function and forward selection. As with the leaps package, we had problems "knitting" the stepAIC function from the MASS package here. Processed fine in R Studio, and led to the four-variable described below in model 6.

```
##
## Call:
## glm(formula = target ~ nox + rad + medv + age, family = "binomial",
##     data = crime_train_data_train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.65002  -0.34124  -0.10477   0.01132   2.76068
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -18.39756    2.53421  -7.260 3.88e-13 ***
## nox          23.79415    4.69296   5.070 3.97e-07 ***
## rad           0.49666    0.13052   3.805 0.000142 ***
## medv          0.06182    0.02781   2.223 0.026196 *
## age           0.01839    0.01133   1.623 0.104634
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 514.63  on 371  degrees of freedom
## Residual deviance: 176.87  on 367  degrees of freedom
## AIC: 186.87
##
## Number of Fisher Scoring iterations: 8
```

In summary, for Part 3, Build Models, we attempted a number of methods to construct a good model. After setting a baseline with all variables, we attempted to address collinearity by removing a variable with a high vif score in Model 2. In Model 3, we applied logarithmic transformations to two of our variables the showd skewed distrubtions. In Model 4, we only included variables that showed significance in Model 1. Models 5 and 6 both used forward selection using R functions.

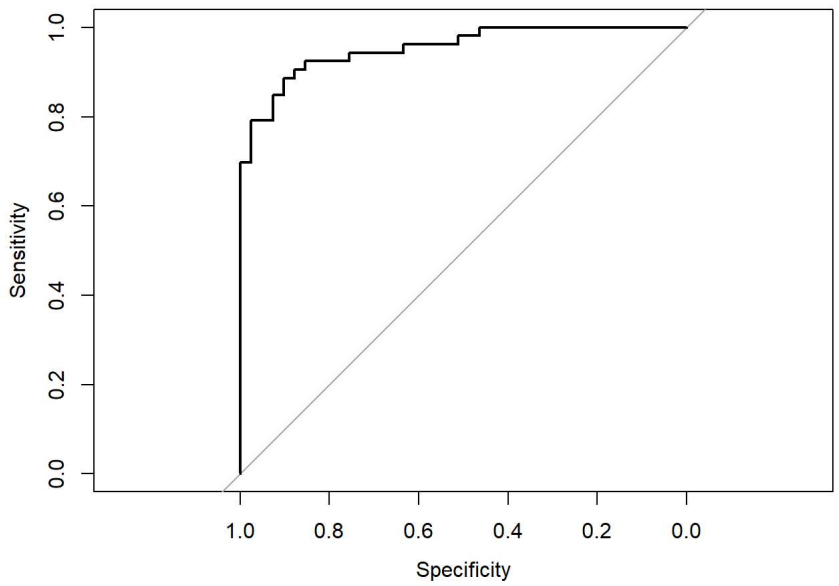
Some of our findings were counterintuitive in that models with more variables showed higher AICs. Also, transfomrations did not seem effective, at least in terms of AIC and residual deviances.

4. Select Models

We will now run that remaining 20% of our training data through the models created with the 80% training set. We'll evaluate models 1, 2, 5, and 6. Finally, we will run the evaluation file that does not include target data through our model to ouptut predictions.

Model 1 confusion matrix and AUC information:

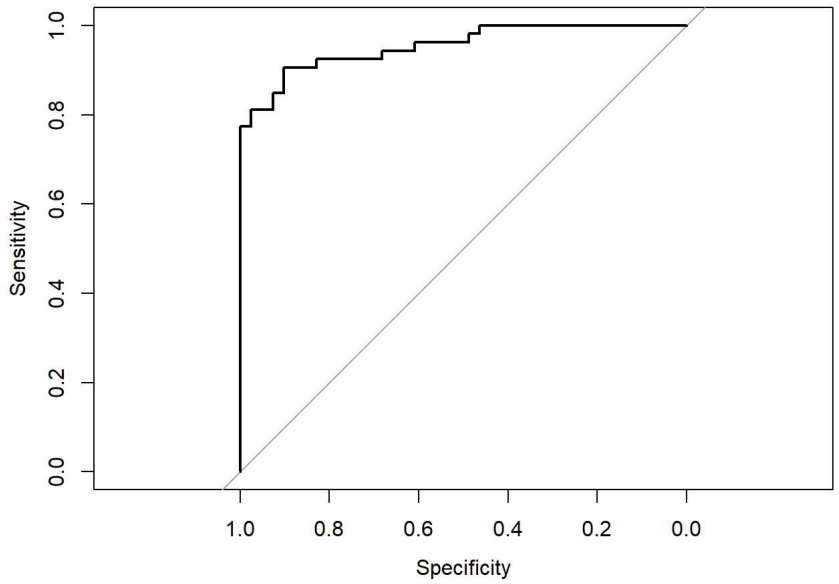
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 36  5
##          1  5 48
##
##                Accuracy : 0.8936
##                  95% CI : (0.813, 0.9478)
##     No Information Rate : 0.5638
##     P-Value [Acc > NIR] : 3.307e-12
##
##                   Kappa : 0.7837
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9057
##             Specificity : 0.8780
##          Pos Pred Value : 0.9057
##          Neg Pred Value : 0.8780
##              Prevalence : 0.5638
##          Detection Rate : 0.5106
##    Detection Prevalence : 0.5638
##       Balanced Accuracy : 0.8919
##
##        'Positive' Class : 1
##
```




```
##
## Call:
## roc.default(response = crime_train_data_valid$target, predictor = model1_predict)
##
## Data: model1_predict in 41 controls (crime_train_data_valid$target 0) < 53 cases (crime_train_data_valid$target 1).
## Area under the curve: 0.954
```

Now the same for model 2:

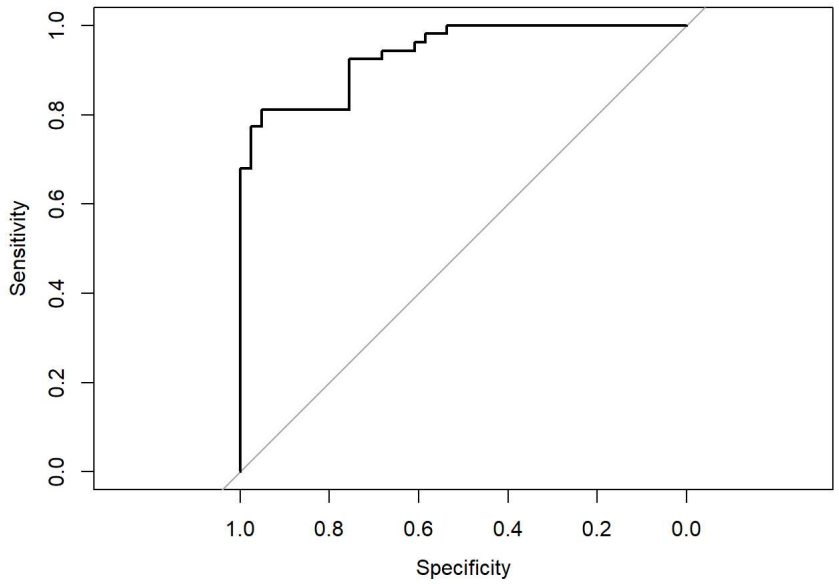
```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction  0  1
##          0 35  5
##          1  6 48
##
##                Accuracy : 0.883
##                  95% CI : (0.8003, 0.9401)
##     No Information Rate : 0.5638
##     P-Value [Acc > NIR] : 1.993e-11
##
##                   Kappa : 0.7614
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9057
##             Specificity : 0.8537
##          Pos Pred Value : 0.8889
##          Neg Pred Value : 0.8750
##              Prevalence : 0.5638
##          Detection Rate : 0.5106
##    Detection Prevalence : 0.5745
##       Balanced Accuracy : 0.8797
##
##        'Positive' Class : 1
##
```




```
##
## Call:
## roc.default(response = crime_train_data_valid$target, predictor = model2_predict)
##
## Data: model2_predict in 41 controls (crime_train_data_valid$target 0) < 53 cases (crime_train_data_valid$target 1).
## Area under the curve: 0.9544
```

And model 5:

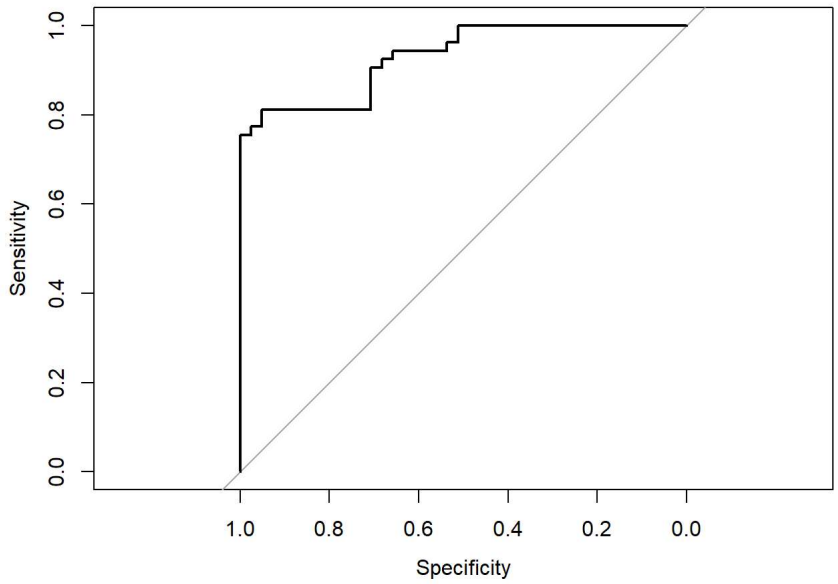
```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction  0  1
##          0 34 10
##          1  7 43
##
##                Accuracy : 0.8191
##                  95% CI : (0.7263, 0.891)
##     No Information Rate : 0.5638
##     P-Value [Acc > NIR] : 1.497e-07
##
##                   Kappa : 0.6353
##  Mcnemar's Test P-Value : 0.6276
##
##             Sensitivity : 0.8113
##             Specificity : 0.8293
##          Pos Pred Value : 0.8600
##          Neg Pred Value : 0.7727
##              Prevalence : 0.5638
##          Detection Rate : 0.4574
##    Detection Prevalence : 0.5319
##       Balanced Accuracy : 0.8203
##
##        'Positive' Class : 1
##
```




```
##
## Call:
## roc.default(response = crime_train_data_valid$target, predictor = model5_predict)
##
## Data: model5_predict in 41 controls (crime_train_data_valid$target 0) < 53 cases (crime_train_data_valid$target 1).
## Area under the curve: 0.9383
```

Finally, model 6:

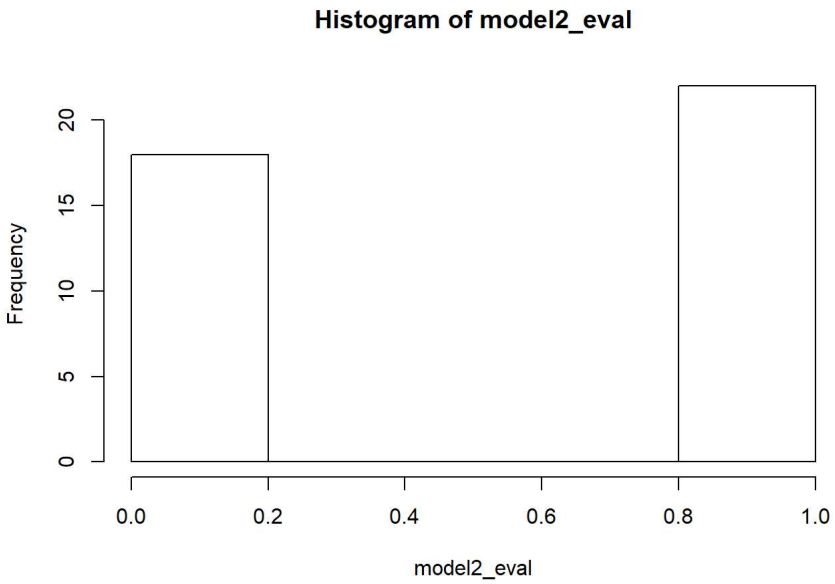
```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction  0  1
##          0 36 10
##          1  5 43
##
##                Accuracy : 0.8404
##                  95% CI : (0.7505, 0.9078)
##     No Information Rate : 0.5638
##     P-Value [Acc > NIR] : 1.05e-08
##
##                   Kappa : 0.68
##  Mcnemar's Test P-Value : 0.3017
##
##             Sensitivity : 0.8113
##             Specificity : 0.8780
##          Pos Pred Value : 0.8958
##          Neg Pred Value : 0.7826
##              Prevalence : 0.5638
##          Detection Rate : 0.4574
##    Detection Prevalence : 0.5106
##       Balanced Accuracy : 0.8447
##
##        'Positive' Class : 1
##
```



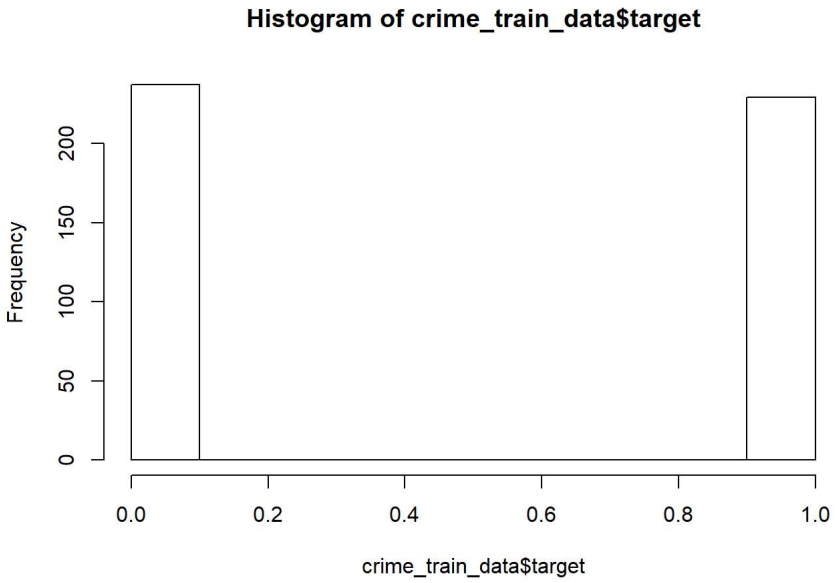

```
##
## Call:
## roc.default(response = crime_train_data_valid$target, predictor = model6_predict)
##
## Data: model6_predict in 41 controls (crime_train_data_valid$target 0) < 53 cases (crime_train_data_valid$target 1).
## Area under the curve: 0.9305
```

Models 1 and 2 provide better accuracy and AIC. Due to a slight edge in simplicity with having one fewer variable, we will go with model 2, which is just all untransformed variables other than the multicollinear rm. In reality, for a given business problem, there are often considerations at play that would lead one to value specificitiy over sensitivity, for example. Predicting that a neighborhood is high crime could be an expensive mistake, or it could be the safer assumption of the two, depending on context. Here, we don't know.

Finally, we run the evaluation dataset through model 2, which is submitted along with this report. A reflection of the predictions of the evaluation set:



The above looks like it predicts neighborhoods to be high-crime at a slightly lower rate than indicated by the frequency distribution of the target value in the original full training set. We would need to know more about the sampling techinque used to split training from evaluation data before determining if this is problematic.



References

Model diagnostics: http://www.sthda.com/english/articles/36-classification-methods-essentials/148-logistic-regression-assumptions-and-diagnostics-in-r/ (http://www.sthda.com/english/articles/36-classification-methods-essentials/148-logistic-regression-assumptions-and-diagnostics-in-r/)

Leaps package usage: https://rstudio-pubs-static.s3.amazonaws.com/2897_9220b21cfc0c43a396ff9abf122bb351.html (https://rstudio-pubs-static.s3.amazonaws.com/2897_9220b21cfc0c43a396ff9abf122bb351.html)

Appendix

load packages

library(knitr) library(dplyr) library(kableExtra)

load training data

url_crime_train <- 'https://raw.githubusercontent.com/littlejohnjeff/Data_621_Fall_2019/master/crime-training-data_modified.csv (https://raw.githubusercontent.com/littlejohnjeff/Data_621_Fall_2019/master/crime-training-data_modified.csv)' crime_train_data <- read.csv(url_crime_train, header = TRUE) kable(crime_train_data[1:15,]) %>% kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"))

summarize training data

library(psych) kable(psych::describe(crime_train_data)) %>% kable_styling(bootstrap_options = c("striped", "hover", "responsive"))

boxplots of each variable split by target value

library(reshape2) library(dplyr) library(ggplot2) crime_plot <- melt(crime_train_data, id.vars= 'target') %>% mutate(target = as.factor(target))

```
ggplot(data = crime_plot, aes(x = variable, y = value)) + geom_boxplot(aes(fill = target)) + facet_wrap( ~ variable, , dir = "h", scales = 'free') + scale_fill_manual(values=c("blue3", "red3"))
```

histograms of training set

```
ggplot(crime_plot,aes(value)) + geom_histogram(bin=25,fill="Red3") + facet_wrap( ~ variable, , dir = "h", scales = 'free')
```

histrogram with log transformations of all variables

```
ggplot(crime_plot,aes(log(value))) + geom_histogram(bin=25,fill="Blue3") + facet_wrap( ~ variable, , dir = "h", scales = 'free')
```

correlations

```
library(stats) library(corrplot) cor_train <- cor(crime_train_data, method="pearson") kable(cor_train, "html") %>% kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"))
```

plot correlations

```
corrplot(cor_train)
```

correlations just with target value

```
cor_train_target <- as.data.frame(cor_train) %>% dplyr::select(target) kable(cor_train_target, "html") %>% kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"))
```

lstat distribution

```
ggplot(crime_train_data,aes(x=lstat)) + geom_histogram(bin=25,fill="Red3")
```

lstat distribution after log transformation

```
ggplot(crime_train_data,aes(x=log(lstat))) + geom_histogram(bin=25,fill="Blue3")
```

split training data into true training and validation/tuning

```
train_size <- floor(0.8 * nrow(crime_train_data))
```

```
set.seed(123) train_ind <- sample(seq_len(nrow(crime_train_data)), size = train_size)
```

```
crime_train_data_train <- crime_train_data[train_ind, ] crime_train_data_valid <- crime_train_data[-train_ind, ]
```

build model 1 - all variables untransformed

```
model1_untransformed <- glm(formula = target ~ ., family = "binomial", data = crime_train_data_train) summary(model1_untransformed)
```

model 1 logit relationships

```
library(tidyr) probabilities <- predict(model1_untransformed, type = "response") predicted.classes <- ifelse(probabilities > 0.5, "pos", "neg") head(predicted.classes)
```

```
mydata <- crime_train_data_train %>% dplyr::select_if(is.numeric) predictors <- colnames(mydata)
```

```
mydata <- mydata %>% mutate(logit = log(probabilities/(1-probabilities))) %>% gather(key = "predictors", value = "predictor.value", -logit)
```

```
ggplot(mydata, aes(logit, predictor.value))+ geom_point(size = 0.5, alpha = 0.5) + geom_smooth(method = "loess") + theme_bw() + facet_wrap(~predictors, scales = "free_y")
```

Cook's distance

https://stat.ethz.ch/R-manual/R-devel/library/stats/html/plot.lm.html (https://stat.ethz.ch/R-manual/R-devel/library/stats/html/plot.lm.html)

```
plot(model1_untransformed, which = 4, id.n = 3)
```

influential outlier checks

```
library(broom) model1_untransformed.data <- augment(model1_untransformed) %>% mutate(index = 1:n())
```

```
kable(model1_untransformed.data %>% top_n(3, .cooksd)) %>% kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"))
```

graph standardized residuals

```
ggplot(model1_untransformed.data, aes(index, .std.resid)) + geom_point(alpha = .5) + theme_light()
```

model 1 vif

```
library(car) car::vif(model1_untransformed)
```

build model 2

```
model2_coll <- glm(formula = target ~ zn + indus + chas + nox + age + dis + rad + tax + ptratio + lstat + medv, family = "binomial", data = crime_train_data_train) summary(model2_coll)
```

build model 3

```
model3_some_logs <- glm(formula = target ~ zn + indus + chas + nox + rm + age + dis + rad + tax + ptratio + log(lstat) + log(medv), family = "binomial", data = crime_train_data_train) summary(model3_some_logs)
```

build model 4

```
model4_sig_var <- glm(formula = target ~ nox + rad + dis + ptratio + medv + tax + age, family = "binomial", data = crime_train_data_train) summary(model4_sig_var)
```

build model 5

```
model5_leap <- glm(formula = target ~ nox + rad + tax + medv + age + indus, family = "binomial", data = crime_train_data_train) summary(model5_leap)
```

build model 6

```
model6_step <- glm(formula = target ~ nox + rad + medv + age, family = "binomial", data = crime_train_data_train) summary(model6_step)
```

model 1 confusion matrix

library(caret) model1_predict <- predict(model1_untransformed, newdata=crime_train_data_valid, type="response") model1_predict_data <- ifelse(model1_predict > .5, 1, 0) confusionMatrix(data=model1_predict_data, crime_train_data_valid$target, positive='1')

model 1 ROC

library('pROC') plot(roc(crime_train_data_valid$target, model1_predict))

model 2 confusion matrix

model2_predict <- predict(model2_coll, newdata=crime_train_data_valid, type="response") model2_predict_data <- ifelse(model2_predict > .5, 1, 0) confusionMatrix(data=model2_predict_data, crime_train_data_valid$target, positive='1')

model 2 roc

plot(roc(crime_train_data_valid$target, model2_predict))

model 5 confusion matrix

model5_predict <- predict(model5_leap, newdata=crime_train_data_valid, type="response") model5_predict_data <- ifelse(model5_predict > .5, 1, 0) confusionMatrix(data=model5_predict_data, crime_train_data_valid$target, positive='1')

model 5 roc

plot(roc(crime_train_data_valid$target, model5_predict))

model 6 confusion matrix

model6_predict <- predict(model6_step, newdata=crime_train_data_valid, type="response") model6_predict_data <- ifelse(model6_predict > .5, 1, 0) confusionMatrix(data=model6_predict_data, crime_train_data_valid$target, positive='1')

model 6 roc

plot(roc(crime_train_data_valid$target, model6_predict))

model 2 eval data histogram

url_crime_eval <- 'https://raw.githubusercontent.com/littlejohnjeff/Data_621_Fall_2019/master/crime-evaluation-data_modified.csv (https://raw.githubusercontent.com/littlejohnjeff/Data_621_Fall_2019/master/crime-evaluation-data_modified.csv)' crime_eval_data <- read.csv(url_crime_eval, header = TRUE) model2_eval_data <- predict(model2_coll, newdata=crime_eval_data, type="response") model2_eval <- ifelse(model2_eval_data > .5, 1, 0) hist(model2_eval) crime_eval_data_out <- cbind(crime_eval_data,model2_eval)

training data histogram

hist(crime_train_data$target)

write eval result file from model 2

write.csv(crime_eval_data_out,"DATA 621Data_621_Hw_3_Evaluation_Output.csv",row.names = FALSE)

frontward steps using LEAPS - didn't run in rmarkdown but ran fine in Ru Studio

library(leaps) reg_subsets.out <- regsubsets(target ~ ., data = crime_train_data_train, nbest = 1, ##### 1 best model for each number of predictors nvmax = NULL, ##### NULL for no limit on number of variables force.in = NULL, force.out = NULL, method = "exhaustive")

library(car) layout(matrix(1:2, ncol = 2)) ########## Adjusted R2 res.legend <- subsets(reg_subsets.out, statistic="adjr2", legend = FALSE, min.size = 5, main = "Adjusted R^2")

summary.out$adjr2

summary.out$which[6,]

frontward setps using MASS package - didn't run in rmarkdown but ran fine in Ru Studio

library(MASS) mod6a1 <- lm(target ~ ., data=crime_train_data_train) mod6a2 <- lm(target ~ 1, data=crime_train_data_train) mod6a <- stepAIC(mod4a2, direction="forward", scope = list(upper=mod6a1, lower=mod6a2)) summary(mod6a)