DATA 621 Business Analytics & Data Mining

Homework #3 Binary Logistic Regression

Kyle Gilde

3/31/2018

Contents

	installed_and_loaded.packages.
prettydoc	TRUE
tidyverse	TRUE
caret	TRUE
pROC	TRUE
DT	TRUE
knitr	TRUE
ggthemes	TRUE
Hmisc	TRUE
psych	TRUE
corrplot	TRUE
reshape2	TRUE
car	TRUE
MASS	TRUE
ResourceSelection	TRUE
boot	TRUE
tinytex	TRUE
devtools	TRUE

Overview

In this homework assignment, you will explore, analyze and model a data set containing information on crime for various neighborhoods of a major city. Each record has a response variable indicating whether or not the crime rate is above the median crime rate (1) or not (0).

Your objective is to 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). Below is a short description of the variables of interest in the data set:

Deliverables

A write-up submitted in PDF format. Your write-up should have four sections. Each one is described below. You may assume you are addressing me as a fellow data scientist, so do not need to shy away from technical details. Assigned prediction (probabilities, classifications) for the evaluation data set. Use 0.5 threshold. Include your R statistical programming code in an Appendix.

1. DATA EXPLORATION

Examine the data

- This data set was first published in 1978, and it contains 13 variables related to housing, property, transportation, geography, environment, education & crime for the Boston metropolitan area.
- For this binary logistic regression, the response variable `target` is either a 1 or 0, where 1 indicates that the crime rate is above the median.
- Of the 12 explanatory variables, 11 are numeric, and only `chas` is categorical, which is a dummy variable indicating whether the suburb borders the Charles River
- The training data contains 466 complete cases while the evaluation data contains 40 complete case.

vars	class_type	n_rows	complete_cases	NA_ct	NA_pct	unique_value_ct	most_common_values
zn	numeric	466	466	0	0	26	0; 20; 80; 12.5; 22
indus	numeric	466	466	0	0	73	18.1; 19.58; 8.14; 6.2; 21.89
chas	integer	466	466	0	0	2	0; 1; NA; NA; NA
nox	numeric	466	466	0	0	79	0.538; 0.437; 0.713; 0.871; 0.489
rm	numeric	466	466	0	0	419	5.713; 6.127; 6.167; 6.229
age	numeric	466	466	0	0	333	100; 95.4; 96; 97.9
dis	numeric	466	466	0	0	380	3.4952; 5.2873; 5.4007; 5.7209
rad	integer	466	466	0	0	9	24; 5; 4; 3; 6
tax	integer	466	466	0	0	63	666; 307; 403; 437; 304
ptratio	numeric	466	466	0	0	46	20.2; 14.7; 21; 17.8; 19.2
lstat	numeric	466	466	0	0	424	6.36; 7.79; 8.05; 3.11
medv	numeric	466	466	0	0	218	50; 22; 23.1; 19.4
target	integer	466	466	0	0	2	0; 1; NA; NA; NA

Data Dictionary

The variable definitions are listed below.

If we use the heuristic that crime-prone areas are more likely to have less desirable characteristics, we would expect

- 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)
- 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)**

Figure 1:

that the crime rate might be **positively correlated** with the follow variables:

- having industry (`indus`)
- the amouunt of pollution (`nox`)
- pupil-teacher ratios (`ptratio`)
- lower social status (`lstat`)

Conversely, we would expect that the crime rate would be **inversely related** to the following variables:

- rate of large residential lots (`zn`)
- the average rooms per dwelling (`rm`)
- access to radial highways (`rad`)
- median owner-occupied home values (`medv`)

Without knowing more about 1970s Boston, it's difficult to hypothesize on the relationship of the crime rate to following variables:

- bordering the Charles River (`chas`)
- having owner-occupied homedbuilt before 1940 (`age`)
- the distance to Boston's employment centers (`dis`)
- the property tax rate per $10,000. (`tax`)

Statistical Summary

- In the summary statistics below, the skewness and kurtosis of the varaibles are not large enough to suggest any variable transformations.
- Besides the dummy variable for bordering the Charles River `chas`, only `zn` has a heavier tail than a normal distribution, and it also has a decent amount of right skewness.

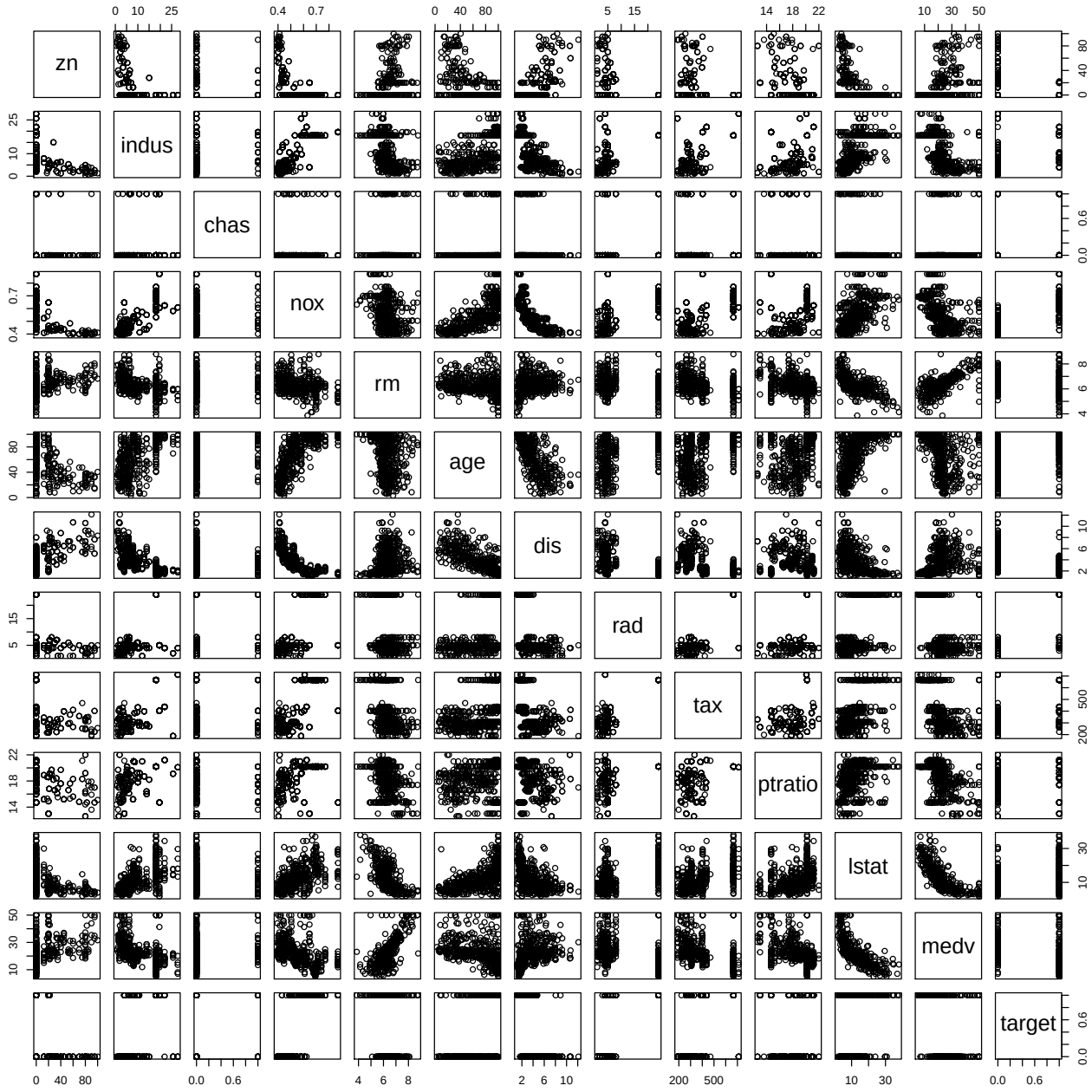
n	mi	n Q__1	st med	ian me	an Q__3	rd ma	x ran	ge	sd ske	w kur	tosis
zn	466	0	0	0	11.6	16.2	100	100	23.4	2.2	3.8
indus	466	0.5	5.1	9.7	11.1	18.1	27.7	27.3	6.8	0.3	-1.2
chas	466	0	0	0	0.1	0	1	1	0.3	3.3	9.1
nox	466	0.4	0.4	0.5	0.6	0.6	0.9	0.5	0.1	0.7	0
rm	466	3.9	5.9	6.2	6.3	6.6	8.8	4.9	0.7	0.5	1.5
age	466	2.9	43.9	77.2	68.4	94.1	100	97.1	28.3	-0.6	-1

n	mi	n Q_1	st med	ian me	an Q_3	rd ma	x ran	ge	sd ske	w kur	tosis
dis	466	1.1	2.1	3.2	3.8	5.2	12.1	11	2.1	1	0.5
rad	466	1	4	5	9.5	24	24	23	8.7	1	-0.9
tax	466	187	281	334.5	409.5	666	711	524	167.9	0.7	-1.1
ptratio	466	12.6	16.9	18.9	18.4	20.2	22	9.4	2.2	-0.8	-0.4
lstat	466	1.7	7	11.4	12.6	16.9	38	36.2	7.1	0.9	0.5
medv	466	5	17	21.2	22.6	25	50	45	9.2	1.1	1.4
target	466	0	0	0	0.5	1	1	1	0.5	0	-2

Visual Exploration

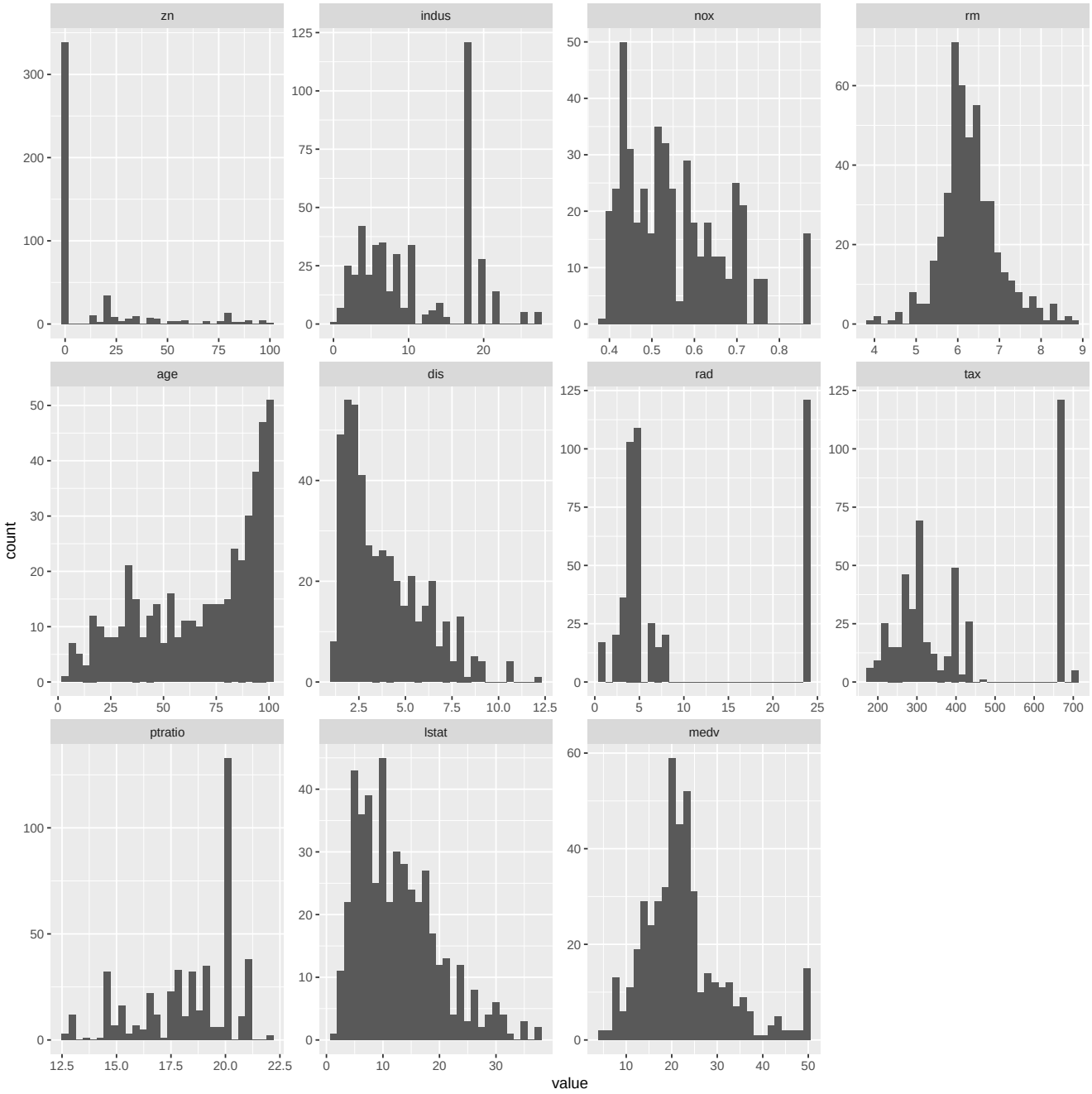
Pairwise scatterplots

- From the scatterplot, several of the predictor variables may have nonlinear relationships with each other. We will use side-by-side boxplots to see how the predictor variables are distributed across the categorical response variable.



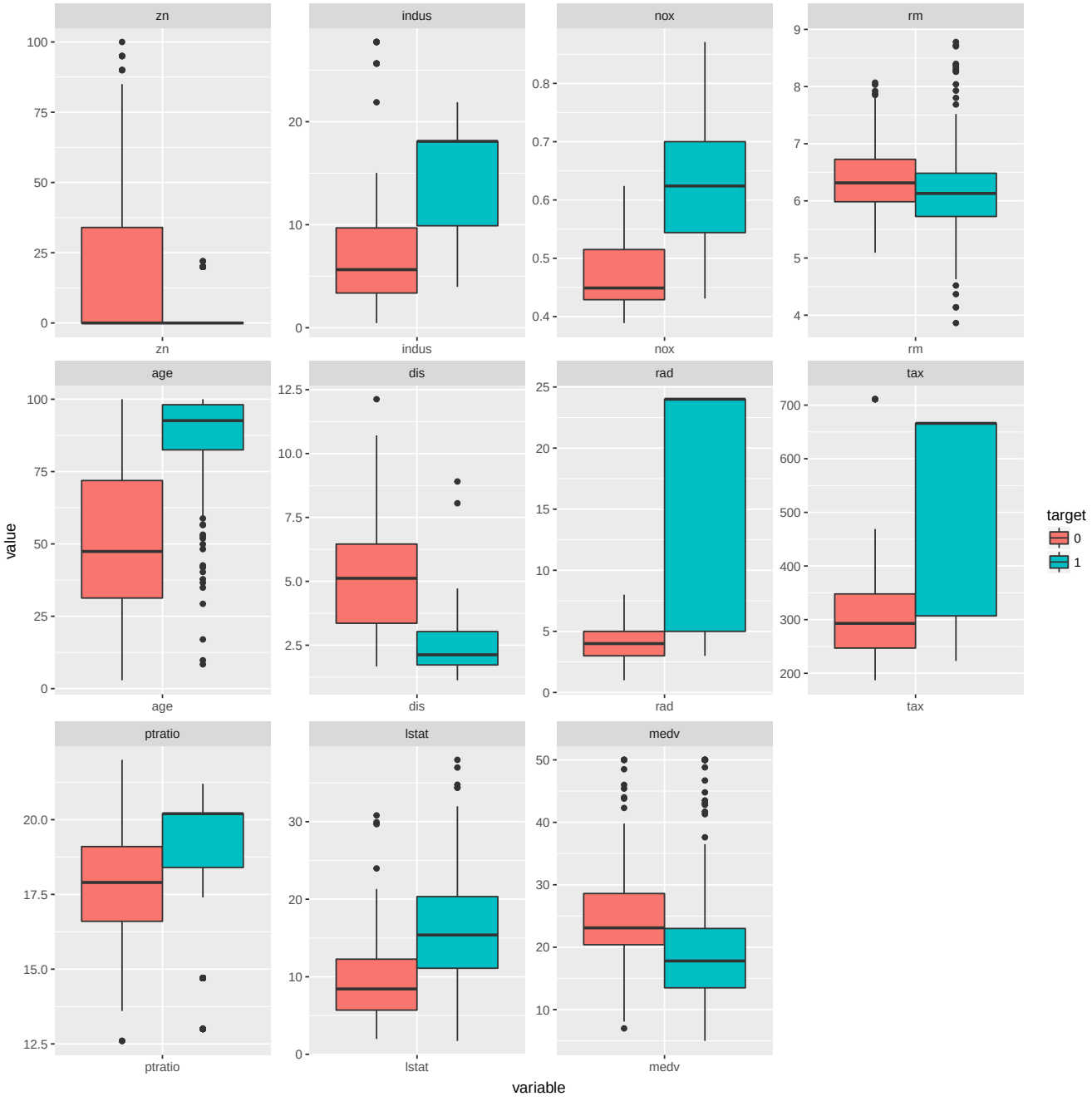
Histograms

- The proportion of large residential lots `zn` is very skewed or is bimodal, and it may benefit from a transformation.
- To lesser degrees, `dis`, `age` & `lstat` are skewed. We will consider transforms for them as well.
- `rad` & `tax`are bimodal.



Side-by-Side Boxplots

- The variance between the 2 values of target differs for `zn`, `nox`, `age`, `dis`, `rad` & `tax`, which indicates that we will want to consider adding quadratic terms for them.



Correlations

- In the correlation plot and the table below it, we see that only 2 of the variables are highly correlated with each other. Having access to radial highways (`rad`) and the property tax rate per $10,000 (`tax`) have a positive correlation coefficent of .91. In our final model, we will want to make sure that we have eliminated any collinearity by checking the variance inflation factor.

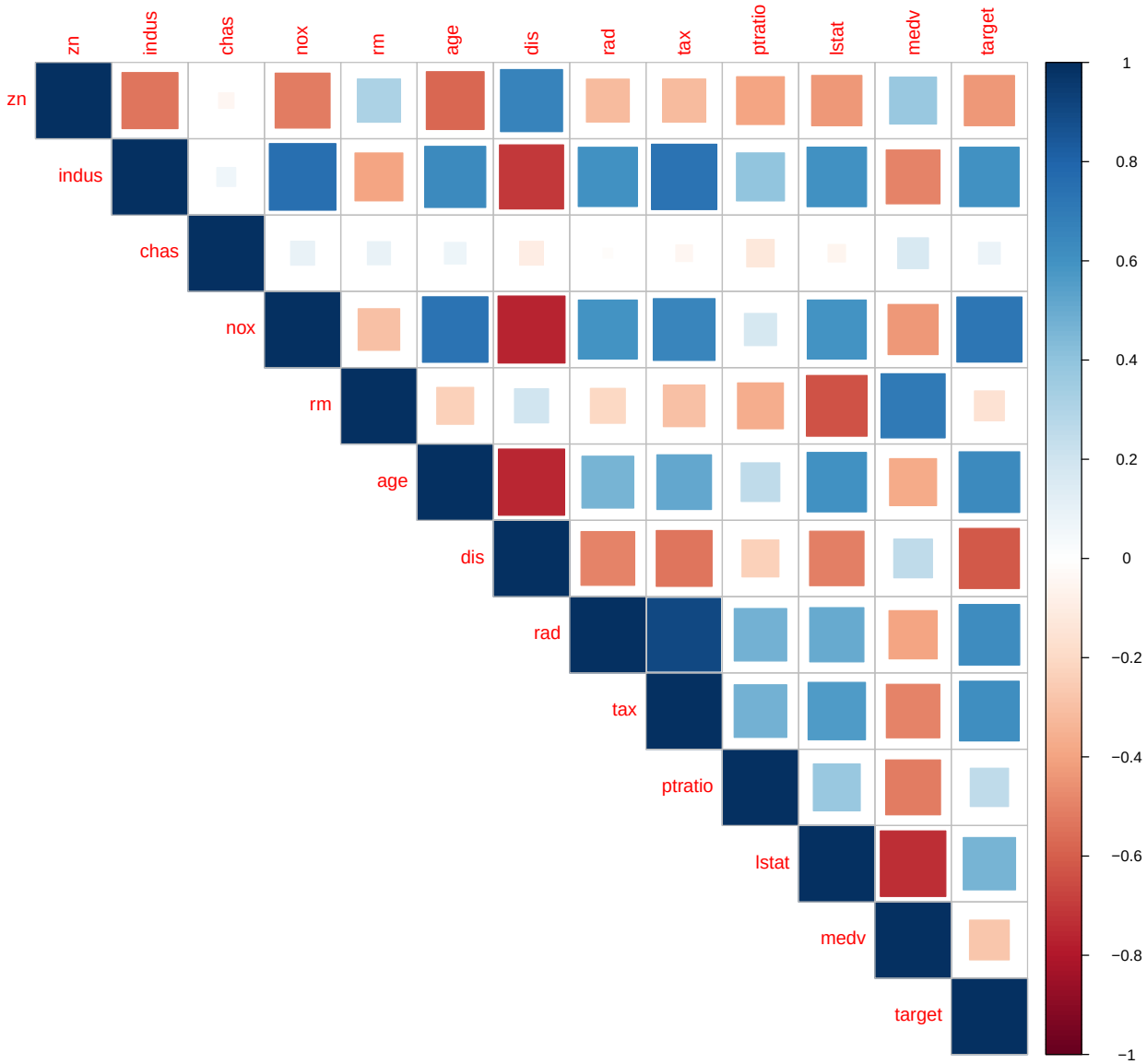


Table 4: Top Correlated Variable Pairs

	Var1	Var2	Correlation	Rsquared
1	rad	tax	0.91	0.82
2	nox	dis	-0.77	0.59
3	indus	nox	0.76	0.58
4	age	dis	-0.75	0.56
5	lstat	medv	-0.74	0.54
6	nox	age	0.74	0.54
7	indus	tax	0.73	0.54
8	nox	target	0.73	0.53
9	rm	medv	0.71	0.50
10	indus	dis	-0.70	0.50

- The next table shows the correlation coefficents with the response variable `target`.
- The concentration of nitrogen oxide (`nox`) has the highest correlation with the response variable with a positive correlation of 0.73. Let's see if `nox` plays a prominent role in our modeling.

Table 5: Corrrelations with the Response Variable

	Var1	Var2	Correlation	Rsquared
1	nox	target	0.73	0.53
2	age	target	0.63	0.40
3	rad	target	0.63	0.39
4	dis	target	-0.62	0.38
5	tax	target	0.61	0.37
6	indus	target	0.60	0.37
7	lstat	target	0.47	0.22
8	zn	target	-0.43	0.19
9	medv	target	-0.27	0.07
10	ptratio	target	0.25	0.06
11	rm	target	-0.15	0.02
12	chas	target	0.08	0.01

2. DATA PREPARATION

- Because of the **skewed distributions** for `age` & `lstat`, we will follow Sheather's quote of Cook & Weisberg on page 284 of MARR and add **log terms** to the model.
- Because of the **variance** between the 2 values of `target` differs for `zn`, `nox` & `rad`, we will follow Sheather's advice on page 289 and add **quadratic terms** to the model.
- For `zn`specifically, after examining the extreme difference in variances in the boxplot, adding a quadratic terms seems most appropriate.

```
## target ~ zn + indus + chas + nox + rm + age + dis + rad + tax +
##     ptratio + lstat + medv + log_age + log_lstat + zn2 + rad2 +
##     nox2
```

3. BUILD MODELS

Base model: All original variables

First, let's take a look at the model that contains all the original variables.

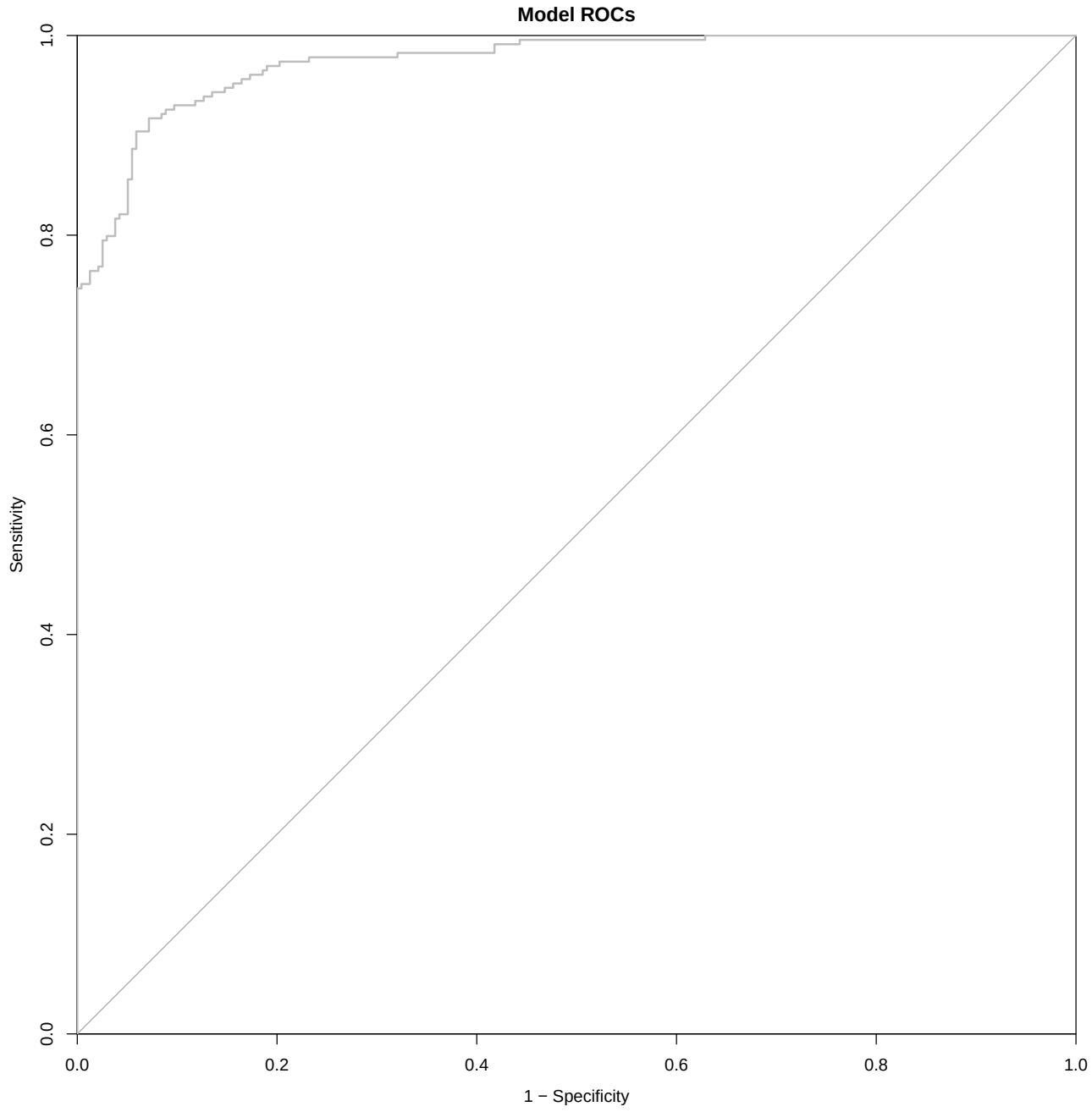
- In the model summary below, 7 of the 12 variables have statistically significant (stat-sig) p-values at a significance level of .05.
- We notice that the variable `nox`, which had the greatest correlation with the response variable has by far the largest coefficient.
- Contrary to what we expected, the proportion of non-retail business `indus` has a negative coefficient. Having access to radial highways `rad` and median owner-occupied home values `medv` have actually have positive coefficients instead of the expected negative ones. This suggests that we may have some multicollinearity to deal with.

```
##
## Call:
## glm(formula = target ~ ., family = "binomial", data = train_data)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8464  -0.1445  -0.0017   0.0029   3.4665
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -40.822934   6.632913  -6.155 7.53e-10 ***
## zn           -0.065946   0.034656  -1.903  0.05706 .
## indus        -0.064614   0.047622  -1.357  0.17485
## chas          0.910765   0.755546   1.205  0.22803
## nox          49.122297   7.931706   6.193 5.90e-10 ***
## rm           -0.587488   0.722847  -0.813  0.41637
## age           0.034189   0.013814   2.475  0.01333 *
## dis           0.738660   0.230275   3.208  0.00134 **
## rad           0.666366   0.163152   4.084 4.42e-05 ***
## tax          -0.006171   0.002955  -2.089  0.03674 *
## ptratio       0.402566   0.126627   3.179  0.00148 **
## lstat         0.045869   0.054049   0.849  0.39608
## medv          0.180824   0.068294   2.648  0.00810 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 192.05  on 453  degrees of freedom
## AIC: 218.05
##
## Number of Fisher Scoring iterations: 9
```

- In the model's summary statistics below, we see that the model's p-value is near zero, so we would reject that the null hypothesis that the coefficient values are equal to zero and not related to the response variable.
- However, the model fails the Hosmer-Lemeshow goodness-of-fit test. With a p-value at .023, we have to reject the test's null hypothesis that the model has a good fit.
- With 3 variables having a variance inflation factor of greater than 4 (`VIF_gt_4`), we do see evidence of multicollinearity.
- The receiver operating characteristic curve plots the true positive rate versus the false positive rate. A value of .5 would incidate that the model is no better than randomly selecting the response variable, and a value of 1 would indicate that the model predicts the correct outcome for all values in the data set. In this first model, the area under the ROC curve (`AUC`) is relatively high at .974. This model is already relatively effective at predicting the response variable.

model_name	n_vars	model_pvalue	residual_deviance	H_L_pvalue	VIF_gt_4	LOOCV_accuracy	AUC
base_model	12	1.06e-100	192.047	0.023	3	0.91	0.974



Base model plus variable transoformations

Next, let's take a look at the model that includes the original variables plus the 5 transformed variables.

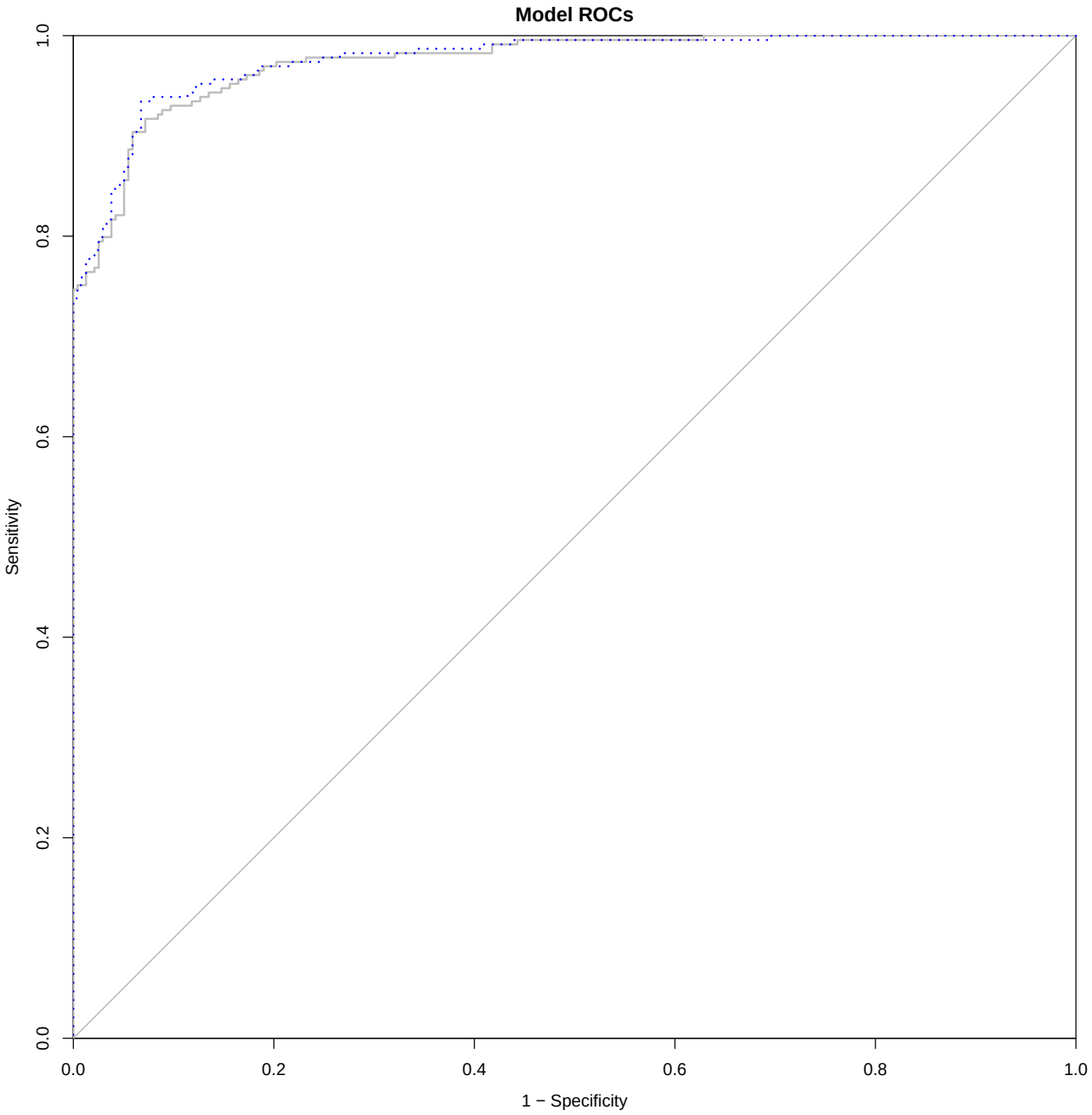
- In this model, only 6 of the 17 variables have stat-sig p-values, and only one of the transformed variables is among them.
- The coefficients of `indus`, `rad`, `medv`, `log_lstat` & `zn2` have the opposite of the signs we expected.
- In this model, the `nox` or `nox2` variables do not stand out from the rest of the coefficients.

```
##
## Call:
## glm(formula = target ~ ., family = "binomial", data = train_data_plus)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0698  -0.1434  -0.0012   0.0022   3.6665
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.396e+01  2.644e+01  -0.528  0.59762
## zn          -7.132e-02  7.728e-02  -0.923  0.35604
## indus       -1.008e-01  5.885e-02  -1.713  0.08680 .
## chas         9.628e-01  7.936e-01   1.213  0.22508
## nox          1.431e+00  9.971e+01   0.014  0.98855
## rm          -1.171e+00  8.043e-01  -1.456  0.14537
## age          9.632e-02  2.925e-02   3.293  0.00099 ***
## dis          6.516e-01  2.550e-01   2.555  0.01062 *
## rad          8.048e-01  5.602e-01   1.437  0.15083
## tax         -8.226e-03  3.568e-03  -2.305  0.02114 *
## ptratio      4.325e-01  1.363e-01   3.174  0.00150 **
## lstat        1.522e-01  1.262e-01   1.205  0.22805
## medv         1.712e-01  7.510e-02   2.279  0.02266 *
## log_age     -2.914e+00  1.119e+00  -2.605  0.00920 **
## log_lstat   -1.958e+00  1.620e+00  -1.209  0.22677
## zn2          3.715e-05  2.240e-03   0.017  0.98677
## rad2        -4.508e-03  4.856e-02  -0.093  0.92603
## nox2         4.806e+01  9.289e+01   0.517  0.60490
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 182.44  on 448  degrees of freedom
## AIC: 218.44
##
## Number of Fisher Scoring iterations: 12
```

- In the model's summary statistics below, this **model's p-value & deviance** have both decreased from the previous 12-variable model.
- However, **Hosmer-Lemeshow goodness-of-fit** p-value has declined as well, which indicates that we have to reject the Hosmer-Lemeshow null hypothesis that the model does have a good fit.
- Since 13 of the variables have a **variance inflation factor** of greater than 4 (`VIF_gt_4`), the number of collinear issues have increased.
- The **leave-one-out cross-validation accuracy** (`LOOCV_accuracy`) is .903 and has declined from the previous model.
- While the area under the **blue-dotted line** of the ROC curve ticked upward to .976, this model has an abundance of problems.

model_name	n_vars	model_pvalue	residual_deviance	H_L_pvalue	VIF_gt_4	LOOCV_accuracy	AUC
base_model	12	1.06e-100	192.047	0.023	3	0.910	0.974
base_model_plus	17	8.60e-103	182.441	0.003	13	0.903	0.976



Backward Elimination

For our 3rd model, let's take the 17 original and transformed variables and perform a backward elimination selection process. We will remove the variable with the highest p-value until only stat-sig variables remain.

- The results below show that our selection process removed 8 of the variables, including 3 of the transformations, which didn't prove to be statistically significant to the model.
- Of the 9 selected variables, the squared transformation of nitrogen oxides concentration `nox2` has by far the largest coefficent at 39.99. We can interpret the effect of of the variable as a unit increase in `nox2` with the other variables held constant increases the log-odds of being above the median crime rate by 39.99.
- `log_age` has the 2nd largest practically significant coefficient at -2.40. We can interpret the effect of of the variable as a unit increase in `log_age` with the other variables held constant decreases the log-odds of being

above the median crime rate by 2.4.

- The coefficients of `rad` & `medv`have the opposite of the signs we expected.

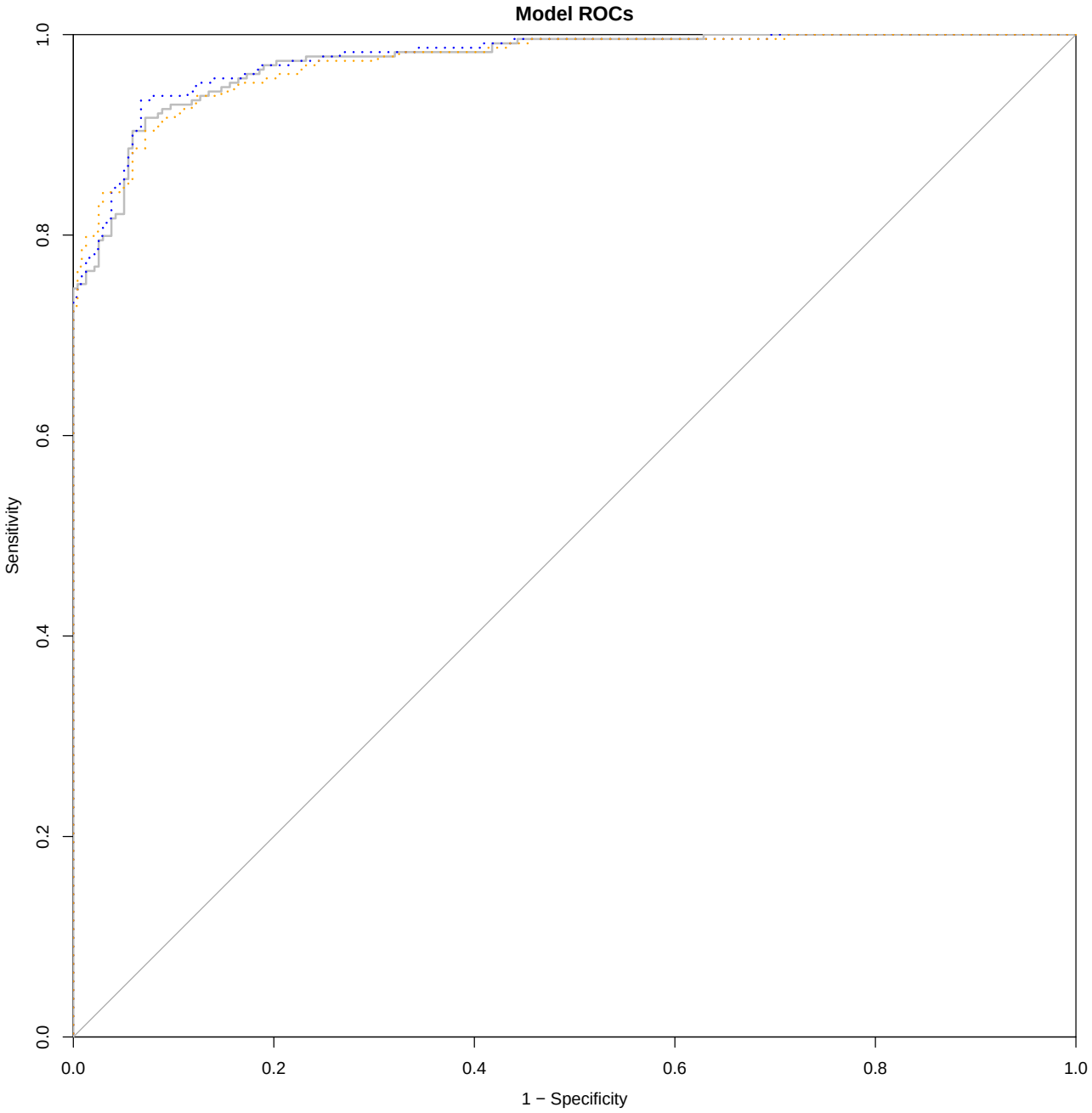
```
## [1] "Removed variables:"
##      removed_vars
## [1,] "nox"        "0.989"
## [2,] "zn2"        "0.987"
## [3,] "rad2"       "0.924"
## [4,] "lstat"      "0.229"
## [5,] "log_lstat"  "0.76"
## [6,] "chas"       "0.204"
## [7,] "indus"      "0.085"
## [8,] "rm"         "0.106"
```

```
##
## Call:
## glm(formula = target ~ zn + age + dis + rad + tax + ptratio +
##     medv + log_age + nox2, family = "binomial", data = train_data_plus)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0309  -0.2137  -0.0015   0.0010   3.5178
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -18.660646   4.766954  -3.915 9.06e-05 ***
## zn           -0.078832   0.033839  -2.330 0.019825 *
## age           0.079466   0.022145   3.588 0.000333 ***
## dis           0.598451   0.217707   2.749 0.005980 **
## rad           0.794790   0.157451   5.048 4.47e-07 ***
## tax          -0.009582   0.002894  -3.311 0.000929 ***
## ptratio       0.325480   0.114203   2.850 0.004372 **
## medv          0.101055   0.036014   2.806 0.005016 **
## log_age      -2.403291   0.910212  -2.640 0.008282 **
## nox2         39.994667   6.239480   6.410 1.46e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 191.54  on 456  degrees of freedom
## AIC: 211.54
##
## Number of Fisher Scoring iterations: 9
```

- In the model's summary statistics below, the **p-value & deviance** have both increased from the previous 17-variable model.
- Additionally, the area under the **orange-dotted line** of the ROC curve is smaller than the previous 2 models, indicating that the model's response probabilities do not correspond as much with the binary outcome of being above or below the median crime rate.
- However, **Hosmer-Lemeshow goodness-of-fit** p-value is not extreme, which indicates a good fit for the model.
- Addtionally, only 2 variables in the model have a **variance inflation factor** of greater than 4 (`VIF_gt_4`), which indicates that there are not as many of collinear issues as the previous models.

- Finally, the **leave-one-out cross-validation accuracy** (`LOOCV_accuracy`) is .91, which means that this model was as good at predicting each of its values as the `base_model`, but used fewer variables and had less multicollinearity.
- Overall, this is a decent model, but let's see if we can do better by finding a model without multicollinearity.

model_name	n_vars	model_pvalue	residual_deviance	H_L_pvalue	VIF_gt_4	LOOCV_accuracy	AUC
base_model	12	1.06e-100	192.047	0.023	3	0.910	0.974
base_model_plus	17	8.60e-103	182.441	0.003	13	0.903	0.976
bk_elim_mod	9	8.23e-101	191.545	0.260	2	0.910	0.973



AIC

For our 4th model, let's take the 17 original and transformed variables and perform a forward & backward Akaike information criterion process, where we will select the model that minimizes the AIC value.

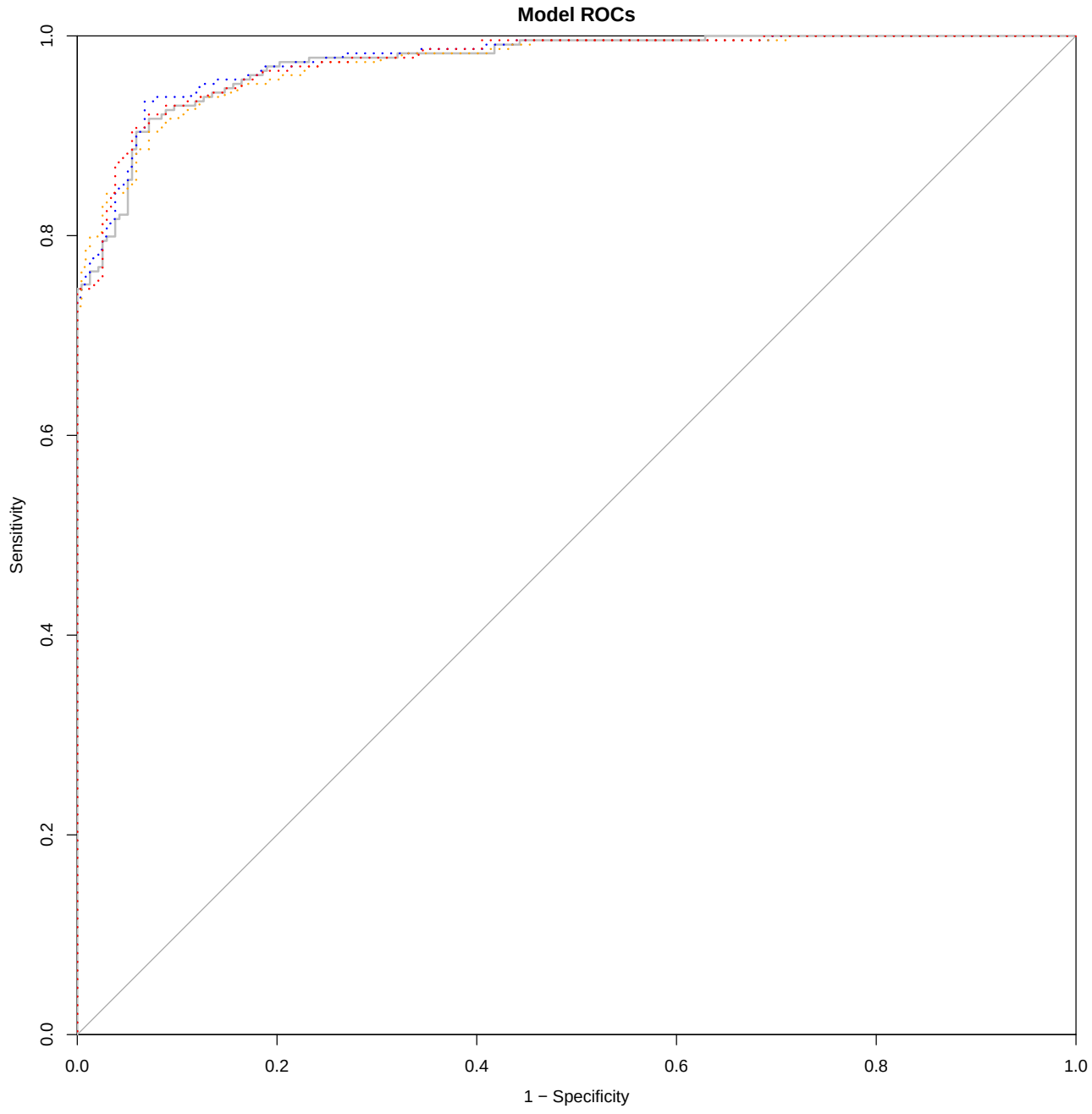
- Our selection processed removed 6 of the variables, including 3 of the 5 variable transformations.
- Of the 11 selected variables, `nox2` and `log_age` still have the 2 largest practically significant coefficients at 47.90 and -3.25, respectively.

```
## [1] "removed variable(s): 6"
## [1] "chas"      "nox"       "lstat"     "log_lstat" "zn2"       "rad2"
```

```
##
## Call:
## glm(formula = target ~ zn + indus + rm + age + dis + rad + tax +
##     ptratio + medv + log_age + nox2, family = "binomial", data = train_data_plus)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9307  -0.1577  -0.0010   0.0008   3.6555
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -15.567472   4.950468  -3.145  0.00166 **
## zn           -0.079234   0.037028  -2.140  0.03237 *
## indus        -0.087685   0.050920  -1.722  0.08506 .
## rm           -1.197138   0.686009  -1.745  0.08097 .
## age           0.105508   0.025753   4.097 4.19e-05 ***
## dis           0.701089   0.234673   2.988  0.00281 **
## rad           0.795293   0.169319   4.697 2.64e-06 ***
## tax          -0.008428   0.003235  -2.606  0.00917 **
## ptratio       0.414148   0.128168   3.231  0.00123 **
## medv          0.197444   0.069794   2.829  0.00467 **
## log_age      -3.246706   0.996112  -3.259  0.00112 **
## nox2         47.901273   7.855502   6.098 1.08e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 185.66  on 454  degrees of freedom
## AIC: 209.66
##
## Number of Fisher Scoring iterations: 9
```

- In the AIC model's summary statistics below, the **p-value & deviance** have both decreased from the backward elimination model.
- Additionally, the area under the **red-dotted line** of the ROC curve is larger than the previous model.
- However, Hosmer-Lemeshow goodness-of-fit p-value **is extreme**, which indicates that the model is not a good fit.
- Addtionally, **more than half** of the variables have a variance inflation factor of greater than 4, which indicates that there is more multicollinearity than the previous model.
- Finally, the leave-one-out cross-validation accuracy (`LOOCV_accuracy`) is the highest so far at .912. We would expect a higher `LOOCV_accuracy` since AIC selection generally optimizes for prediction.

model_name	n_vars	model_pvalue	residual_deviance	H_L_pvalue	VIF_gt_4	LOOCV_accuracy	AUC
base_model	12	1.06e-100	192.047	0.023	3	0.910	0.974
base_model_plus	17	8.60e-103	182.441	0.003	13	0.903	0.976
bk_elim_mod	9	8.23e-101	191.545	0.260	2	0.910	0.973
AIC_mod	11	4.31e-102	185.659	0.008	6	0.912	0.975



BIC

For our final model, let's take the 17 original and transformed variables and perform a forward & backward Baysean information criterion process, where we will select the model that minimizes the AIC value.

- As we would expect, the BIC process with its larger predictor penalty selected fewer varaiables than AIC.

- It removed 9 of the variables, including 4 of the 5 variable transformations.
- Of the 11 selected variables, `nox2` has the largest practical significance with a coefficient of 39.5. Its 95% confidence interval is between 28.24 and 52.47.

```
## [1] "removed variable(s): 9"
## [1] "indus"     "chas"      "nox"       "rm"        "lstat"     "log_age"
## [7] "log_lstat" "zn2"       "rad2"
```

```
##
## Call:
## glm(formula = target ~ zn + age + dis + rad + tax + ptratio +
##     medv + nox2, family = "binomial", data = train_data_plus)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9135  -0.1972  -0.0018   0.0016   3.4249
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -25.687340   4.542669  -5.655 1.56e-08 ***
## zn           -0.078820   0.032991  -2.389  0.01689 *
## age           0.033277   0.010928   3.045  0.00233 **
## dis           0.647667   0.213828   3.029  0.00245 **
## rad           0.739843   0.149302   4.955 7.22e-07 ***
## tax          -0.008184   0.002674  -3.060  0.00221 **
## ptratio       0.321032   0.112292   2.859  0.00425 **
## medv          0.107232   0.035092   3.056  0.00225 **
## nox2         39.498279   6.148288   6.424 1.33e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 196.77  on 457  degrees of freedom
## AIC: 214.77
##
## Number of Fisher Scoring iterations: 9
```

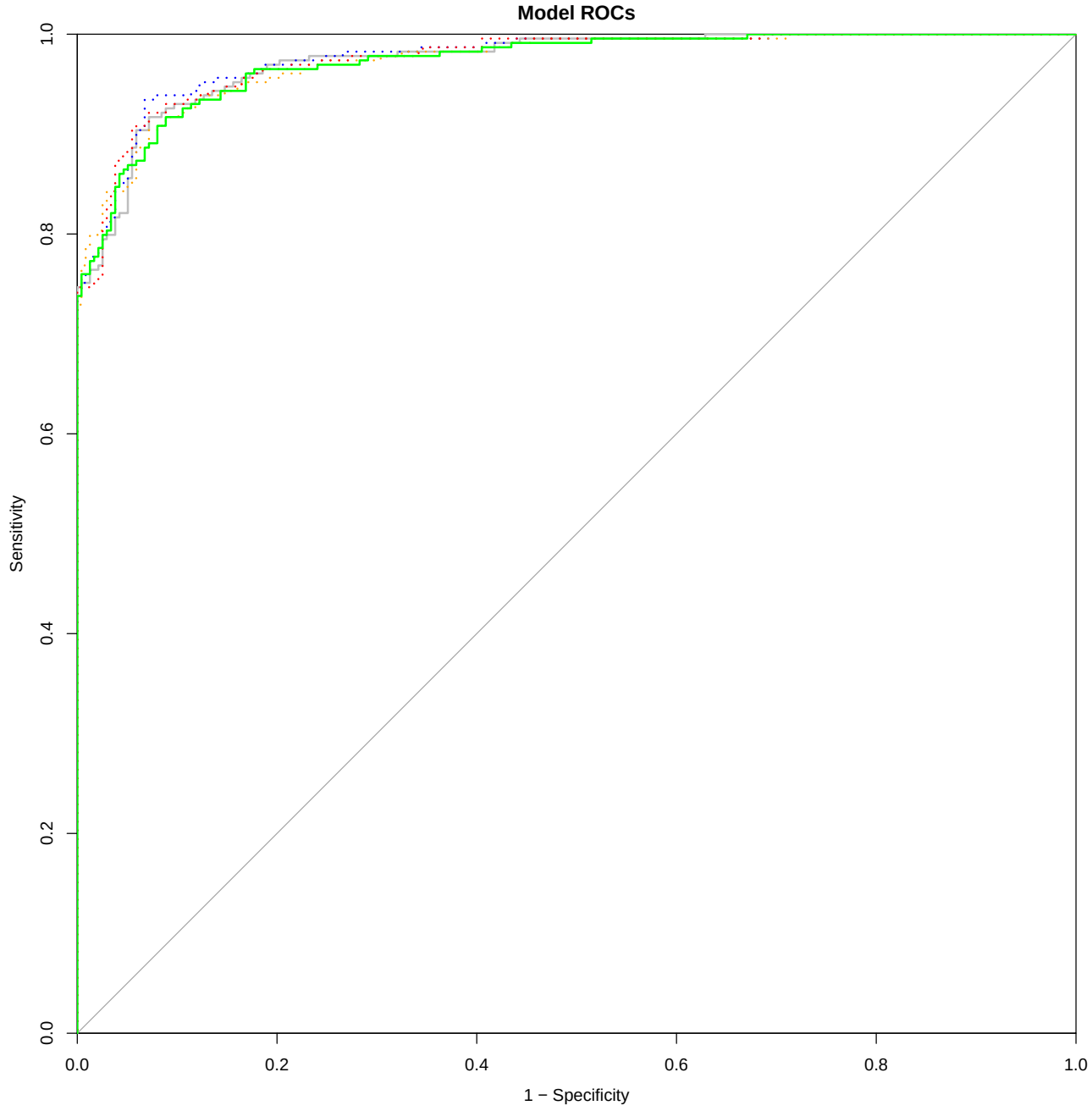
```
## Waiting for profiling to be done...
```

```
##                    2.5 %        97.5 %
## (Intercept) -35.30459011 -17.384769045
## zn           -0.14966926  -0.020919518
## age           0.01247599   0.055539294
## dis           0.24334735   1.087101102
## rad           0.46535040   1.054467045
## tax          -0.01395333  -0.003296981
## ptratio       0.10588331   0.548732141
## medv          0.04131377   0.179190905
## nox2         28.24031852  52.473874894
```

- In the BIC model's summary statistics below, the p-value & deviance are the largest of all the models.
- Additionally, the area under the **green line** of the ROC curve is the smallest.
- Hosmer-Lemeshow goodness-of-fit p-value **is not extreme**, which indicates that the model is a good fit.
- Moreover, these model **does not** have any of the collinearity of the previous models.

- Finally, the leave-one-out cross-validation accuracy (`LOOCV_accuracy`) is the 2nd lowest of all the models.

model_name	n_vars	model_pvalue	residual_deviance	H_L_pvalue	VIF_gt_4	LOOCV_accuracy	AUC
base_model	12	1.06e-100	192.047	0.023	3	0.910	0.974
base_model_plus	17	8.60e-103	182.441	0.003	13	0.903	0.976
bk_elim_mod	9	8.23e-101	191.545	0.260	2	0.910	0.973
AIC_mod	11	4.31e-102	185.659	0.008	6	0.912	0.975
BIC_mod	8	1.13e-99	196.771	0.257	0	0.908	0.972



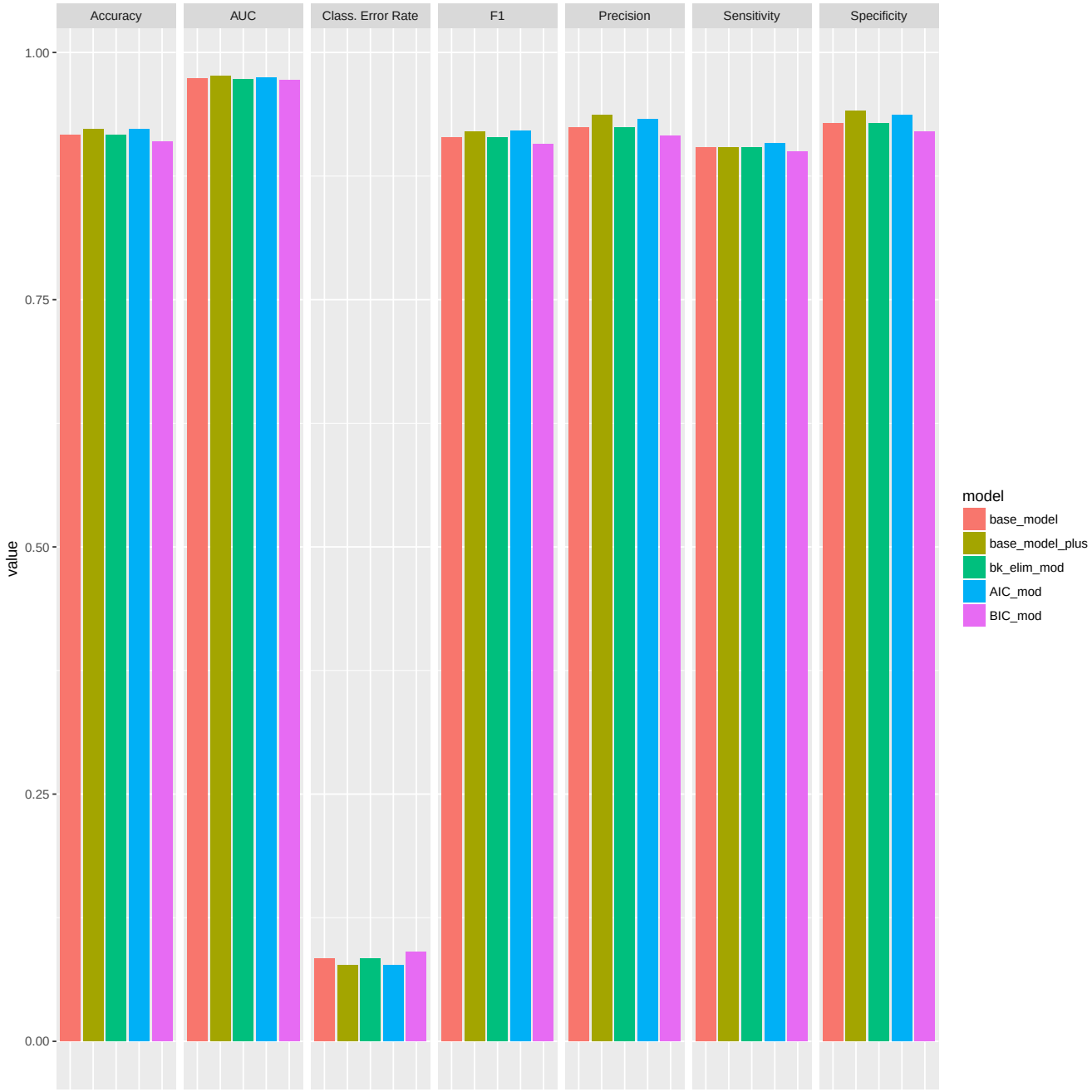
4. SELECT MODELS

Confusion Matrix Metrics

While the BIC model performs the worst on several of confusion matrix metrics, it is by only **neglible** amounts.

The **strengths of the BIC model** include the following:

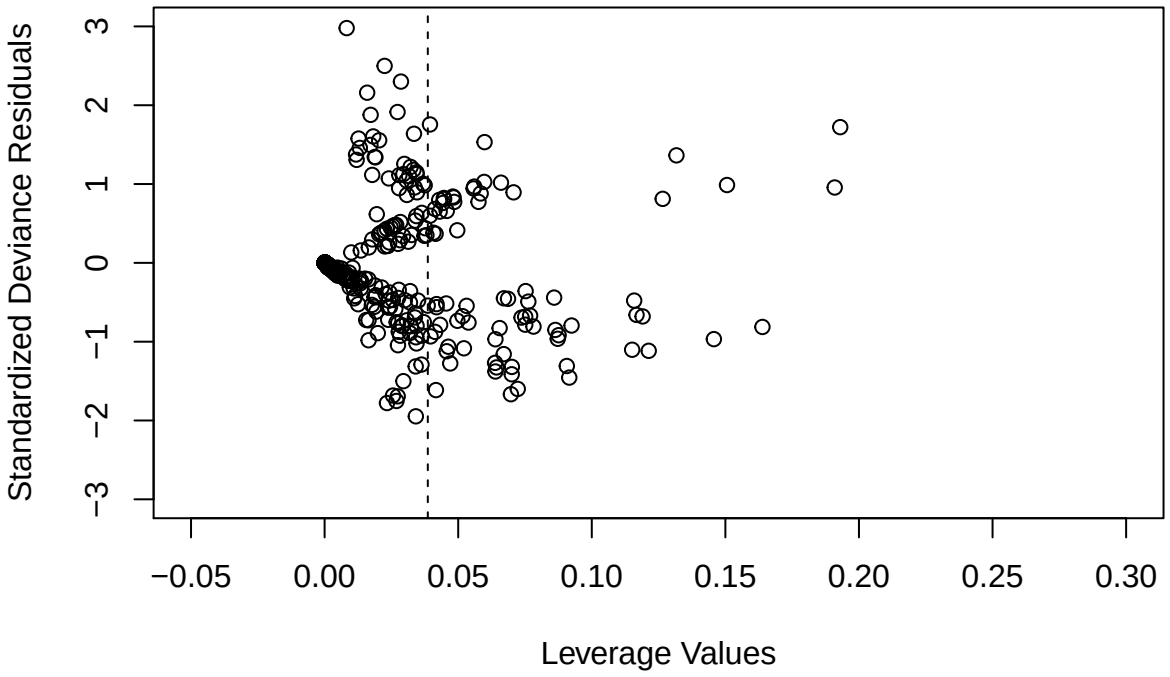
- With only 8 variables, it is the most parsimonious.
- It passes the Hosmer-Lemeshow goodness-of-fit test.
- It doesn't have multicollinearity issues.



Diagnostics

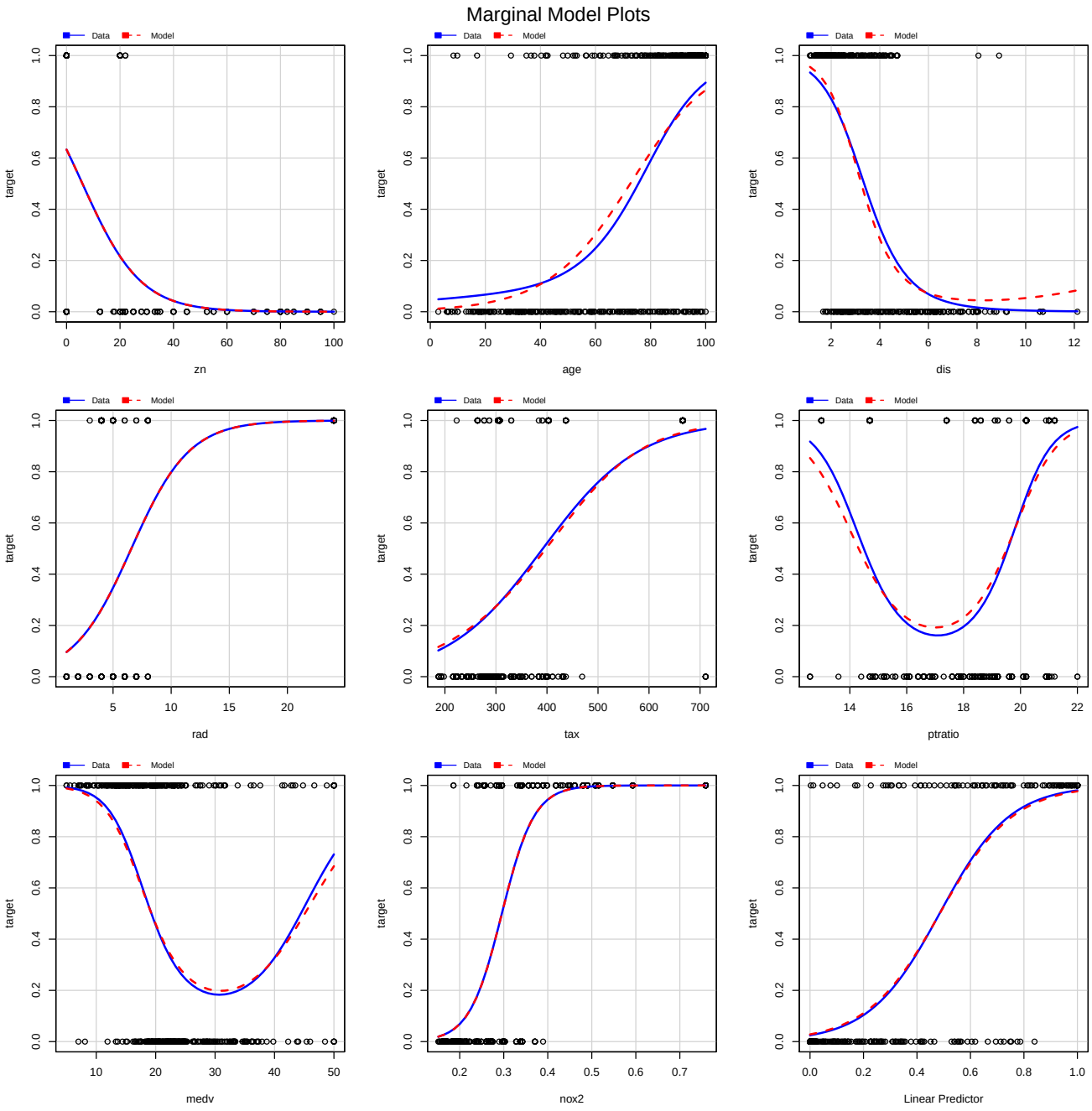
Influence Leverage Values

However, a plot of the Standardized Deviance Values against the leverage values shows that we have several observations greater than twice the average leverage value. This indicates that there may be other possible variable transformations that we have not considered.



Marginal Model Plots

The marginal model plots of the response variable versus the predictors and the fitted response values show that the model closely aligns with the smooth fit function.



Evalution data set

Finally, when we apply the BIC model to the evalution data, it predicts that there are 21 observations below the median crime rate and 19 above the median crime rate.

```
##
##  0  1
## 21 19
```

Code Appendix

```
knitr::opts_chunk$set(
                      error = F
                      , message = T
                      #,tidy = T
                      , cache = T
                      , warning = F
                      , echo = F
                      )
  # prettydoc::html_pretty:
  #   theme: cayman
    # highlight: github

installed_and_loaded <- function(pkg){
  # Load packages. Install them if needed.
  # CODE SOURCE: https://gist.github.com/stevenworthington/3178163
  new.pkg <- pkg[!(pkg %in% installed.packages()[, "Package"])]
  if (length(new.pkg)) install.packages(new.pkg, dependencies = TRUE)
  sapply(pkg, require, character.only = TRUE, quietly = TRUE, warn.conflicts = FALSE)
}



# requi/red packages
packages <- c("prettydoc","tidyverse", "caret", "pROC", "DT", "knitr", "ggthemes", "Hmisc", "psych", "corr

#install_version("rmarkdown",version=1.8)

#excute function and display the loaded packages
data.frame(installed_and_loaded(packages))
#import data

train_data <- read.csv("https://raw.githubusercontent.com/kylegilde/D621-Data-Mining/master/HW3%20Binary%2

eval_data <- read.csv("https://raw.githubusercontent.com/kylegilde/D621-Data-Mining/master/HW3%20Binary%20

# train_data <- train_raw
# train_data$chas <- as.factor(train_data$chas)
# train_data$target <- as.factor(train_data$target)

metadata <- function(df){
  #Takes a data frame & Checks NAs, class types, inspects the unique values
  df_len <- nrow(df)
  NA_ct = as.vector(rapply(df, function(x) sum(is.na(x))))

  #create dataframe
  df_metadata <- data.frame(
    vars = names(df),
    class_type = rapply(lapply(df, class), function(x) x[[1]]),
    n_rows = rapply(df, length),
    complete_cases = sum(complete.cases(df)),
    NA_ct = NA_ct,
    NA_pct = NA_ct / df_len * 100,
    unique_value_ct = rapply(df, function(x) length(unique(x))),
    most_common_values = rapply(df, function(x) str_replace(paste(names(sort(summary(as.factor(x)), decreas
```

```
  )
 rownames(df_metadata) <- NULL
 return(df_metadata)
}

meta_df <- metadata(train_data)

#datatable(meta_df, options = list(searching = F, paging = F))
kable(meta_df, digits = 1, format.args = list(big.mark = ',', scientific = F, drop0trailing = T))
#dimensions
nrow(train_data)
nrow(eval_data)
sum(complete.cases(train_data))
sum(complete.cases(eval_data))

metrics <- function(df){
  ###Creates summary metrics table
  metrics_only <- df[, which(rapply(lapply(df, class), function(x) x[[1]]) %in% c("numeric", "integer"))]

  df_metrics <- psych::describe(metrics_only, quant = c(.25,.75))

  df_metrics <-
    dplyr::select(df_metrics, n, min, Q_1st = Q0.25, median, mean, Q_3rd = Q0.75,
    max, range, sd, skew, kurtosis
  )

  return(df_metrics)
}

metrics_df <- metrics(train_data)
#(dt_metrics <- datatable(round(metrics_df, 2), options = list(searching = F, paging = F)))
kable(metrics_df, digits = 1, format.args = list(big.mark = ',', scientific = F, drop0trailing = T))

pairs(train_data)

#Predictor histograms
train_melted <- reshape::melt(train_data, id.vars = "target") %>%
  dplyr::filter(variable != "chas") %>%
  mutate(target = as.factor(target))

ggplot(data = train_melted, aes(x = value)) +
  geom_histogram(bins = 30) +
  facet_wrap(~variable, scales = "free")

#https://www3.nd.edu/~steve/computing_with_data/13_Facets/facets.html


### Side-by-Side Boxplots
ggplot(data = train_melted, aes(x = variable, y = value)) +
  geom_boxplot(aes(fill = target)) +
  facet_wrap( ~ variable, scales = "free")

#Reference: https://stackoverflow.com/questions/14604439/plot-multiple-boxplot-in-one-graph?utm_medium=org
##CORRELATIONS
cormatrix <- cor(train_data)
```

```r
#plot
corrplot(cormatrix, method = "square", type = "upper")

#find the top correlations
correlations <- c(cormatrix[upper.tri(cormatrix)])
cor_df <- data.frame(Var1 = rownames(cormatrix)[row(cormatrix)[upper.tri(cormatrix)]],
                     Var2 = colnames(cormatrix)[col(cormatrix)[upper.tri(cormatrix)]],
                     Correlation = correlations,
                     Rsquared = correlations^2) %>%
  arrange(-Rsquared)
#Reference: https://stackoverflow.com/questions/28035001/transform-correlation-matrix-into-dataframe-with-

kable(head(cor_df, 10), digits = 2, row.names = T, caption = "Top Correlated Variable Pairs")
#Corrrelations with Target
target_corr <- subset(cor_df, Var2 == "target" | Var1 == "target")
rownames(target_corr) <- 1:nrow(target_corr)

kable(target_corr, digits = 2, row.names = T, caption = "Corrrelations with the Response Variable")
# 2. DATA PREPARATION

train_data_plus <-
  train_data %>%
  mutate(
    log_age = log(age),
    log_lstat = log(lstat),
    zn2 = zn^2,
    rad2 = rad^2,
    nox2 = I(nox^2)
  )


base_model_plus <- glm(target ~ . , family = "binomial", data = train_data_plus)

formula(base_model_plus)

#+  log(dis) + I(tax^2)
## Base model: All original variables
summary(base_model <- glm(target ~ . , family = "binomial", data = train_data))


model_summary <- function(model) {
  ### Summarizes the model's key statistics
  ### References: https://www.r-bloggers.com/predicting-creditability-using-logistic-regression-in-r-cross
  ### https://www.rdocumentation.org/packages/boot/versions/1.3-20/topics/cv.glm
  ### https://www.rdocumentation.org/packages/ResourceSelection/versions/0.3-1/topics/hoslem.test
  cost <- function(r, pi = 0) mean(abs(r - pi) > 0.5)

  df_summary <- data.frame(
    model_name = deparse(substitute(model)),
    n_vars = length(coef(model)) - 1,
    model_pvalue = formatC(pchisq(model$null.deviance - model$deviance, 1, lower=FALSE), format = "e", dig
    residual_deviance = model$deviance,
    H_L_pvalue = hoslem.test(model$y, fitted(model))$p.value,
    VIF_gt_4 = sum(car::vif(model) > 4),
    LOOCV_accuracy = 1 - cv.glm(model$model, model, cost = cost)$delta[1],
    AUC = as.numeric(pROC::roc(model$y, fitted(model))$auc)
```

```
  )
}


mod_sum <- model_summary(base_model)
kable(all_results <- mod_sum, digits = 3)


base_model_roc <- roc(base_model$y, fitted(base_model))
plot(base_model_roc, legacy.axes = T, main = "Model ROCs", col = "gray", xaxs = "i", yaxs = "i")

#https://web.archive.org/web/20160407221300/http://metaoptimize.com:80/qa/questions/988/simple-explanation
## Base model plus variable transoformations
summary(base_model_plus)


mod_sum <- model_summary(base_model_plus)
kable(all_results <- rbind(all_results, mod_sum), digits = 3)


base_model_plus_roc <- roc(base_model_plus$y, fitted(base_model_plus))

plot(base_model_roc, legacy.axes = T, main = "Model ROCs", col = "gray", xaxs = "i", yaxs = "i")
plot(base_model_plus_roc, add = T, col = "blue", lty = 3)

#https://stats.stackexchange.com/questions/29039/plotting-overlaid-roc-curves?utm_medium=organic&utm_sourc
## Backward Elimination

backward_elim_glm <- function(glmod){
  #performs backward elimination model selection
  #removes variables until all remaining ones are stat-sig
  removed_vars <- c()
  removed_pvalues <- c()

  while (max(summary(glmod)$coefficients[, 4]) > .05){
    # find insignificant pvalue
    pvalues <- summary(glmod)$coefficients[, 4]
    max_pvalue <- max(pvalues)
    remove <- names(which.max(pvalues))
    removed_vars <- c(removed_vars, remove)
    removed_pvalues <- c(removed_pvalues, max_pvalue)
    # update model
    glmod <- update(glmod, as.formula(paste(".~.-", remove)))
  }

  print("Removed variables:")
  print(cbind(removed_vars, round(removed_pvalues,3)))
  return(glmod)
}

summary(bk_elim_mod <- backward_elim_glm(base_model_plus))


#format(exp(max(coef(bk_elim_mod))), scientific = F)
#exp(sort(coef(bk_elim_mod))[2])
# http://www.stat.tamu.edu/~sheather/book/docs/rcode/Chapter8.R
```

```r
mod_sum <- model_summary(bk_elim_mod)
kable(all_results <- rbind(all_results, mod_sum), digits = 3)


bk_elim_mod_roc <- roc(bk_elim_mod$y, fitted(bk_elim_mod))

plot(base_model_roc, legacy.axes = T, main = "Model ROCs", col = "gray", xaxs = "i", yaxs = "i")
plot(base_model_plus_roc, add = T, col = "blue", lty = 3)
plot(bk_elim_mod_roc, add = T, col = "orange", lty = 3)


AIC_mod <- MASS::stepAIC(base_model_plus, trace = 0)

removed_variables <- function(larger_mod, smaller_mod){
  removed <- names(coef(larger_mod))[!names(coef(larger_mod)) %in%
names(coef(smaller_mod))]
    print(paste("removed variable(s):", length(removed)))
    print(removed)
}

removed_variables(base_model_plus, AIC_mod)

summary(AIC_mod)
mod_sum <- model_summary(AIC_mod)
kable(all_results <- rbind(all_results, mod_sum), digits = 3)

AIC_mod_roc <- roc(AIC_mod$y, fitted(AIC_mod))

plot(base_model_roc, legacy.axes = T, main = "Model ROCs", col = "gray", xaxs = "i", yaxs = "i")
plot(base_model_plus_roc, add = T, col = "blue", lty = 3)
plot(bk_elim_mod_roc, add = T, col = "orange", lty = 3)
plot(AIC_mod_roc, add = T, col = "red", lty = 3)


n <- nrow(base_model_plus$model)

BIC_mod <- step(base_model_plus, k = log(n), trace = 0)
removed_variables(base_model_plus, BIC_mod)
summary(BIC_mod)

confint(BIC_mod)

#Reference: https://stackoverflow.com/questions/19400494/running-a-stepwise-linear-model-with-bic-criterio
mod_sum <- model_summary(BIC_mod)
kable(all_results <- rbind(all_results, mod_sum), digits = 3)

BIC_mod_roc <- roc(BIC_mod$y, fitted(BIC_mod))

plot(base_model_roc, legacy.axes = T, main = "Model ROCs", col = "gray", xaxs = "i", yaxs = "i")
plot(base_model_plus_roc, add = T, col = "blue", lty = 3)
plot(bk_elim_mod_roc, add = T, col = "orange", lty = 3)
plot(AIC_mod_roc, add = T, col = "red", lty = 3)
plot(BIC_mod_roc, add = T, col = "green")
# 4. SELECT MODELS
all_models <- as.character(all_results$model_name)
```

```r
confusion_metrics <- data.frame(metric =  c("Accuracy", "Class. Error Rate", "Sensitivity", "Specificity",

for (i in 1:length(all_models)){
  model <- get(all_models[i])
  model_name <- all_models[i]
  predicted_values <- as.factor(as.integer(fitted(model) > .5))
  CM <- confusionMatrix(predicted_values, as.factor(model$y), positive = "1")
  caret_metrics <- c(CM$overall[1],
                     1 - as.numeric(CM$overall[1]),
                     CM$byClass[c(1, 2, 5, 7)],
                     get(paste0(model_name, "_roc"))$auc)
  confusion_metrics[, model_name] <- caret_metrics
}

confusion_metrics_melted <- confusion_metrics %>%
  reshape::melt(id.vars = "metric") %>%
  dplyr::rename(model = variable)

ggplot(data = confusion_metrics_melted, aes(x = model, y = value)) +
  geom_bar(aes(fill = model), stat='identity') +
  theme(axis.ticks.x=element_blank(),
        axis.text.x=element_blank(),
        axis.title.x=element_blank()) +
  facet_grid(~metric)

#https://stackoverflow.com/questions/18624394/ggplot-bar-plot-with-facet-dependent-order-of-categories/186
#https://stackoverflow.com/questions/18158461/grouped-bar-plot-in-ggplot?utm_medium=organic&utm_source=goo

#kable(confusion_metrics, digits = 3)

# influential leverage values
# MARR p291
hvalues <- influence(BIC_mod)$hat
stanresDeviance <- residuals(BIC_mod) / sqrt(1 - hvalues)
n_predictors <- length(names(BIC_mod$model)) - 1
average_leverage <- (n_predictors + 1) / nrow(BIC_mod$model)
plot(hvalues, stanresDeviance,
     ylab = "Standardized Deviance Residuals",
     xlab = "Leverage Values",
     ylim = c(-3, 3),
     xlim = c(-0.05, 0.3))
abline(v = 2 * average_leverage, lty = 2)

#Reference: http://www.stat.tamu.edu/~sheather/book/docs/rcode/Chapter8.R

car::mmps(BIC_mod)

#http://support.sas.com/documentation/cdl/en/statug/68162/HTML/default/viewer.htm#statug_templt_sect027.ht
## Evalution data set

eval_data_plus <-
  eval_data %>%
  mutate(
    log_age = log(age),
    log_lstat = log(lstat),
```

```
    zn2 = zn^2,
    rad2 = rad^2,
    nox2 = I(nox^2)
  )

eval_results <- predict(BIC_mod, newdata = eval_data_plus)

table(as.integer(eval_results > .5))
```