# Data 621 Homework 3

Critical Thinking Group 3: Vyannna Hill, Jose Rodriguez, and Christian Uriostegui

2023-12-18

## /// Data Exploration

### //// Introduction to the data set

For this assignment, the team will review the provided crime data set. The team will create a regression model that will predict the risk assessment of a particular neighborhood.

Let’s review the data below:

| Variable Name | Definition | Value |
|---|---|---|
| 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 |

| Variable Name | Definition | Value |
|---|---|---|
| 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 |

Reviewing the training set, the data set has 466 observations and 12 predictor variables. There are no missing values in the training set, therefore there is no need for imputation. However, the mean and median of several variables look strange. Let's review a visualization of the distribution of those values below.

```
## [1] "Number of observations: 466"
```

```
##        zn             indus            chas              nox
##  Min.   :  0.00   Min.   : 0.460   Min.   :0.00000   Min.   :0.3890
##  1st Qu.:  0.00   1st Qu.: 5.145   1st Qu.:0.00000   1st Qu.:0.4480
##  Median :  0.00   Median : 9.690   Median :0.00000   Median :0.5380
##  Mean   : 11.58   Mean   :11.105   Mean   :0.07082   Mean   :0.5543
##  3rd Qu.: 16.25   3rd Qu.:18.100   3rd Qu.:0.00000   3rd Qu.:0.6240
##  Max.   :100.00   Max.   :27.740   Max.   :1.00000   Max.   :0.8710
##        rm             age              dis              rad
##  Min.   :3.863   Min.   :  2.90   Min.   : 1.130   Min.   : 1.00
##  1st Qu.:5.887   1st Qu.: 43.88   1st Qu.: 2.101   1st Qu.: 4.00
##  Median :6.210   Median : 77.15   Median : 3.191   Median : 5.00
##  Mean   :6.291   Mean   : 68.37   Mean   : 3.796   Mean   : 9.53
##  3rd Qu.:6.630   3rd Qu.: 94.10   3rd Qu.: 5.215   3rd Qu.:24.00
##  Max.   :8.780   Max.   :100.00   Max.   :12.127   Max.   :24.00
##       tax           ptratio         lstat             medv
##  Min.   :187.0   Min.   :12.6   Min.   : 1.730   Min.   : 5.00
##  1st Qu.:281.0   1st Qu.:16.9   1st Qu.: 7.043   1st Qu.:17.02
##  Median :334.5   Median :18.9   Median :11.350   Median :21.20
##  Mean   :409.5   Mean   :18.4   Mean   :12.631   Mean   :22.59
##  3rd Qu.:666.0   3rd Qu.:20.2   3rd Qu.:16.930   3rd Qu.:25.00
##  Max.   :711.0   Max.   :22.0   Max.   :37.970   Max.   :50.00
##      target
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.4914
##  3rd Qu.:1.0000
##  Max.   :1.0000
```

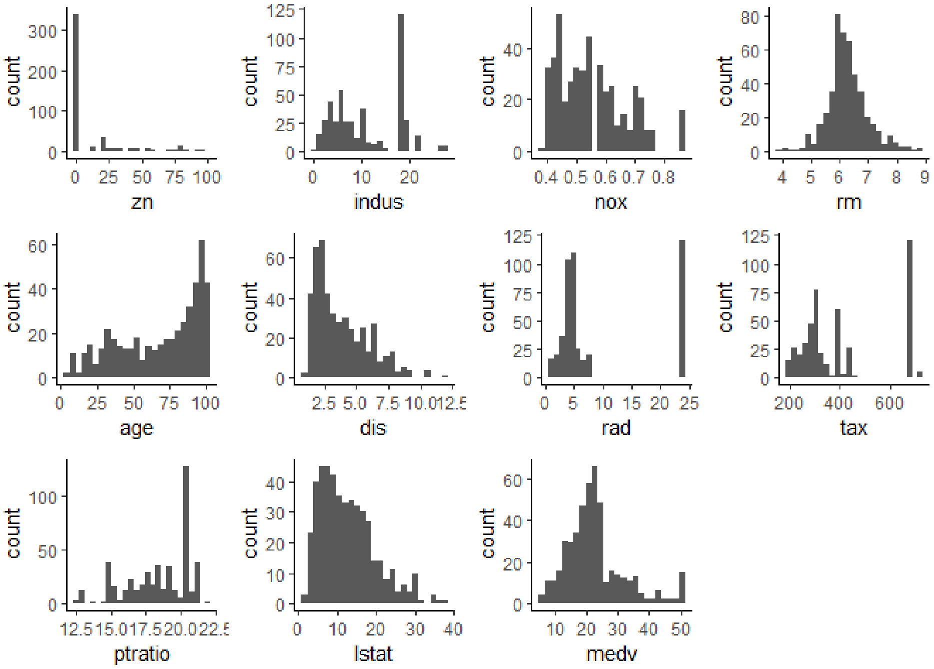
```
##      zn   indus    chas     nox      rm     age     dis     rad     tax ptratio
##       0       0       0       0       0       0       0       0       0       0
##   lstat    medv  target
##       0       0       0
```

## //// Running into heavily skewed data

Looking at the density plots of each predictor variable, only the "*average number of rooms per dwelling (rm)*" variable exhibits a normal distribution. For the variable "*zn*", the data is highly right skewed. This can represent this location does not have a lot of large residential plots; which could mean the area does not see larger apartments or luxury houses.

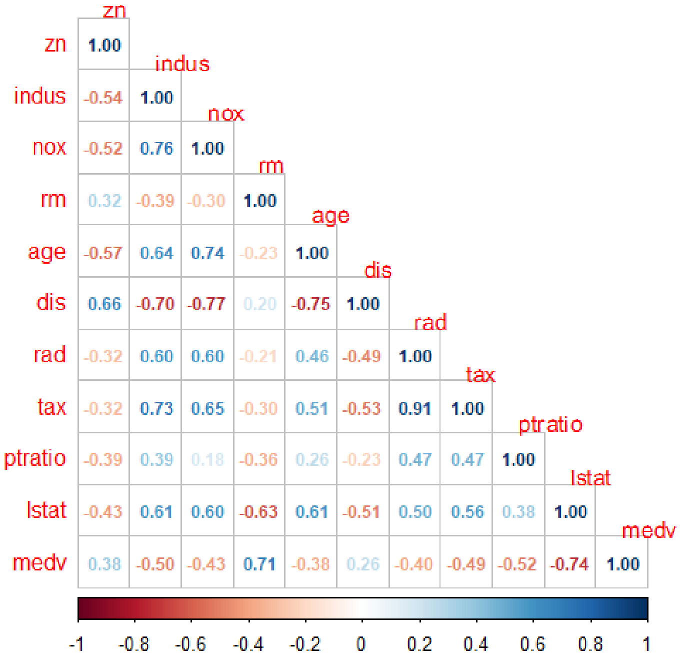
Another questionable variable is "*rad*", which is the index of accessibility to radial highway. This definition measures the dependence of a car in a location^(https://www.sciencedirect.com/science/article/pii/S0966692323000388). The scoring of *rad* in this data set ranks this location to be poorly accessible as majority of the observations ranks accessibility at 24.

Possible paths for the heavily skewed may need a transformation, but let's see if there are predictors that are multi-collinear that can be removed pre-transformation!



## //// Reviewing for multi-collinearity

In review of our predictor variables, rad and tax have a high correlation of 0.91 compared to other variables. The team can remove rad variable from the preliminary regression model as tax variable requires less transformations. There are some moderately correlated variables in the graph below (i.e *nox*&*dis*,*indus*&*nox*), but the team can investigate in model building if variable removal is necessary.



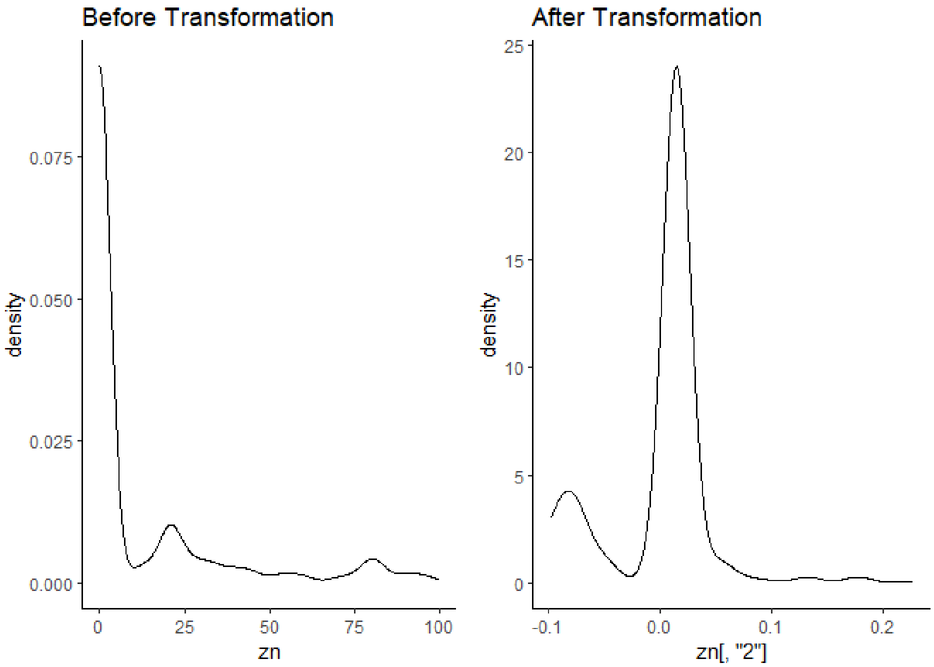
## /// Data Preparation

From the data exploration, it was found a few features are heavily skewed. However, this is a logistic model and not a linear model. There is no assumption of a normal distribution like a linear regression model will assume.

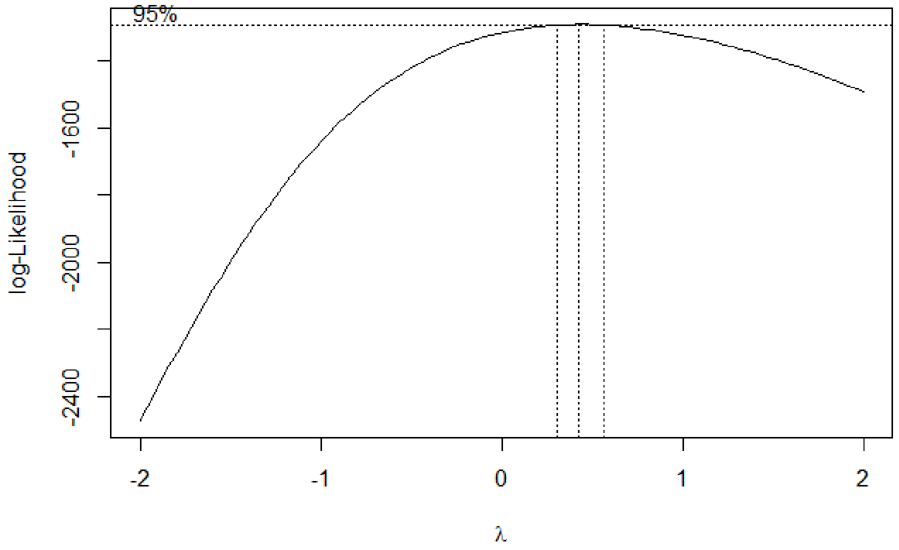
- The assumptions for a logistic regression are:
  - No Multi-collinearity
  - Residuals are independent
  - Large sample size
  - Linear relationship between predictors and logit of Y

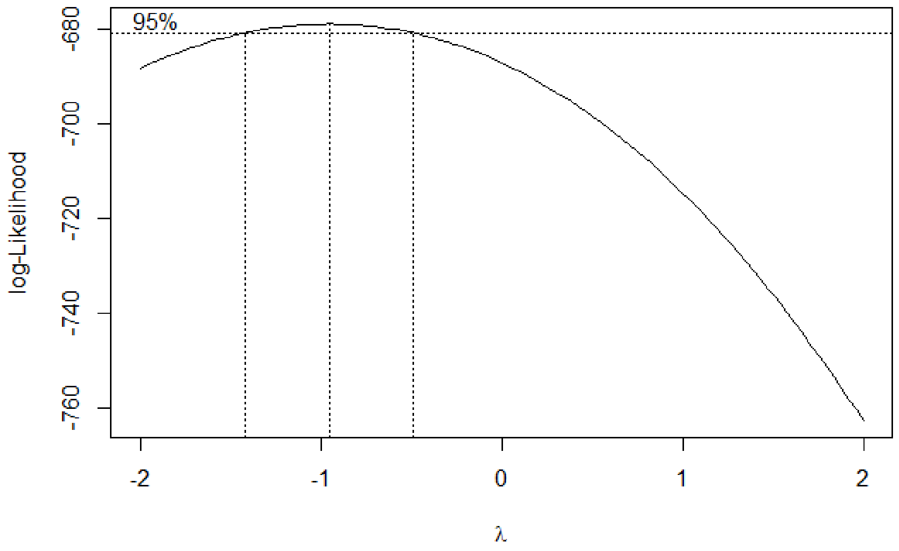
The most useful transformation is handling the highly skewed features for the model.

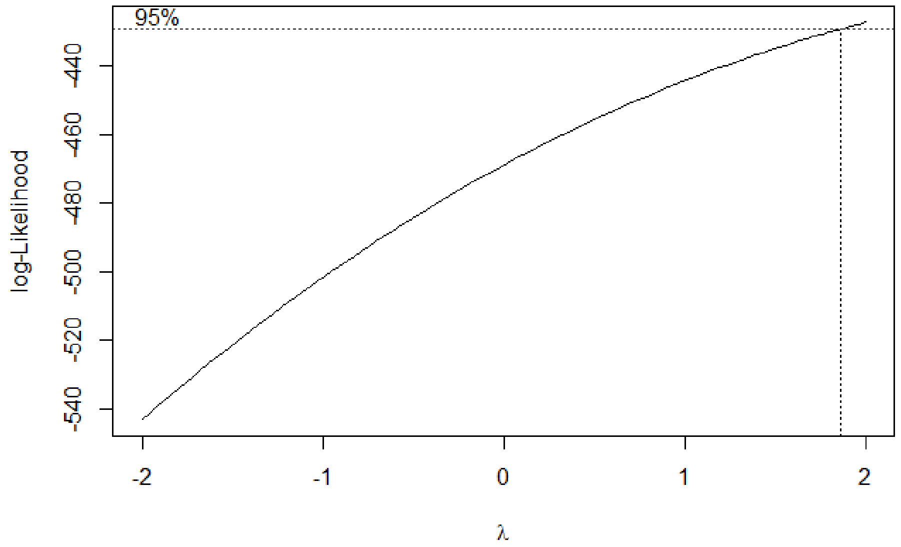
*zn*, *tax*, *pratio*, *nox*, and *indus* will need transformations before the feature selection. The variable *zn* is very fragment in its distribution, so it might benefit from a polynomial transformation. It is a quicker transformation than creating dummy variables for all quartiles in *zn*.

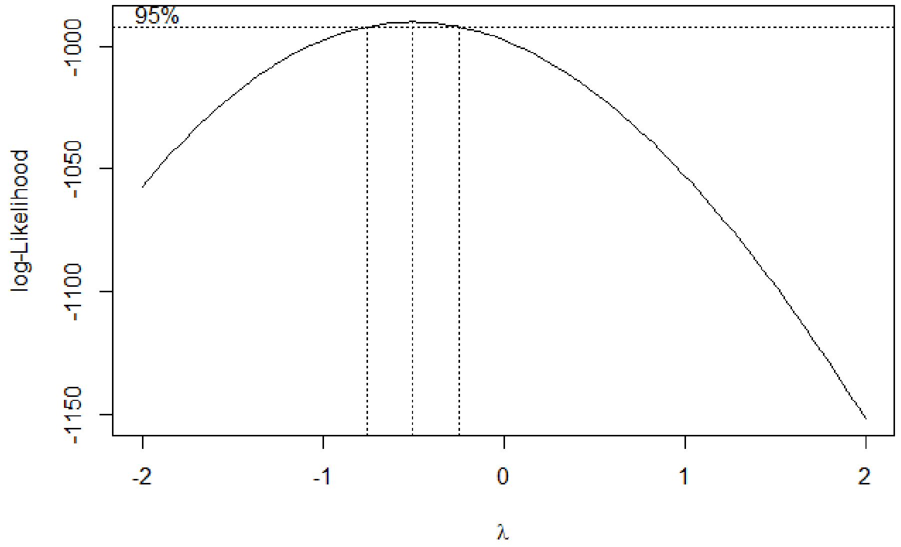


We can determine the other feature’s transformation through box cox!


95%
log-Likelihood
-1600
-2000
-2400
-2
-1
0
1
2
λ


95%
log-Likelihood
-680
-700
-720
-740
-760
-2
-1
0
1
2
λ





Now, the team can use the newly transform variables in the variable selection process!

## /// Build Models - Methodology

In all three models, we’re going to assigning the variable target as our dependent value, and every other variable as our predictors. We want to see what variables, such as *zn*, *indus*, and *chas*, have on target. Target tells us whether the crime rate in the area is high or low. If a coefficient is positive, this tell us that the higher the value of the variable, the higher the odds of Target to be 1 - which tell us the crime rate in the area is high. The opposite occurs when coefficient is negative.

## /// Model Fit 1: Full Baseline Model

Our residuals in the baseline model, which contains the original dataset without modifications, ranges from -1.8464 to 3.4665.

### Positive variables

The variables *chas*, and *lstat* have positive coefficients. The higher these values, the higher the chances that the crime rate in the area is high. In other words if the area is near the Charles River, or if there is a higher concentration of population that is on the lower end of socio economic status, they will likely be in a high crime rate area. These variables are not statistically significant.

The coefficients align with what we assume to be key identifiers of the crime rate of the area. While we are unsure why the Charles River has a positive coefficient, one can assume the it's likely going to be a high crime area. If there is a high concentration of poverty, then there will most likely be crime and so the positive coefficient for *lstat* also makes sense.

### Negative variables

The variables *zn*, *indus*, and *rm* have negative coefficients. The higher these values, the lower the chances that the crime rate in the area is high. If the proportion of residential land zoned for large lots is large, if the proportion of non-retail business acres per suburb is large, or if the average number of rooms per dwelling is large then the more likely they will be in a low crime rate area, These variables are not statistically significant.

These variables aren't as intuitive as when looking at the variables with positive coefficients. This tells us that areas with larger residential zoning (more open space), areas with non-retail businesses (local businesses) and average rooms, the more likely there will be a low crime area. These amenities likely offer more economic opportunities (jobs), more space and are most probable of being occupant by affluent families.

### Significant values

The variables nox, age, dis, rad, ptratio, medv and tax are all statistically significant.

If there is high population (*nox*), if the building is older (*age*), if there isn't close proximity to employment centers (*dis*), if there is close proximity to highways (*rad*) and high pupil to teacher ration (*ptratio*) the higher the chances that the crime rate in the area is high. The positive values of these coefficients is fitting because these values indicate negative environmental status, and weak economic opportunity, all which typically exist in high crime areas.

Areas with higher property taxes are less likely to be an area with high crime rate.

### AIC and BIC

The AIC for this model is 218.0469 and the BIC is 271.9213

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

```
## # A tibble: 1 × 8
##   null.deviance df.null logLik   AIC   BIC deviance df.residual  nobs
##           <dbl>   <int>  <dbl> <dbl> <dbl>    <dbl>       <int> <int>
## 1          646.     465  -96.0  218.  272.     192.         453   466
```

### /// Model Fit 2: Full Transformed Model

When comparing the transformed model to the baseline model, we can see a wider range when looking at the residuals. Model 2 ranged from -1.7292 to 3.7084.

Similarly we can see most of the same significant variables that we see in model 1: *nox*, *dis*, *rad*, *ptratio2*, and *medv* *tax* is not statistically significant in this model.

Interestingly, the *nox* variable in this model is negative coefficient value unlike the first model which was a positive coefficient value.

The AIC is 220.1605 and BIC is 282.3232 which is slightly larger than the first model. This indicates the first model is a better fit.

```
##
## Call:
## glm(formula = target ~ ., family = "binomial", data = train.set)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.15029    6.77902   0.465 0.642138
## zn1         -37.27656   26.43593  -1.410 0.158519
## zn2          11.02723   13.78141   0.800 0.423622
## indus         0.08989    0.36254   0.248 0.804181
## chas          0.87964    0.74507   1.181 0.237752
## nox          -8.11295    2.33219  -3.479 0.000504 ***
## rm           -0.65207    0.74510  -0.875 0.381494
## age           0.02546    0.01378   1.848 0.064546 .
## dis           0.72152    0.24611   2.932 0.003371 **
## rad           0.72166    0.18621   3.875 0.000106 ***
## tax          34.07820   40.43657   0.843 0.399364
## ptratio1      6.38668    6.81561   0.937 0.348725
## ptratio2     19.08677    5.91581   3.226 0.001254 **
## lstat         0.04778    0.05559   0.860 0.390060
## medv          0.18571    0.07218   2.573 0.010088 *
## ---
## 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: 190.16  on 451  degrees of freedom
## AIC: 220.16
##
## Number of Fisher Scoring iterations: 9
```

```
## # A tibble: 1 × 8
##   null.deviance df.null logLik   AIC   BIC deviance df.residual  nobs
##           <dbl>   <int>  <dbl> <dbl> <dbl>    <dbl>       <int> <int>
## 1          646.     465  -95.1  220.  282.     190.         451   466
```

### /// Model Fit 3: Stepwise Regression Model

For the final model, we utilize stepwise regression. This is a technique that seeks a parsimonious model, rather than one with a large selection of predictor values. It contains both forward selection and backward elimination. Forward selection starts with no variables and then adds them one by one. It only adds variables that give the most improvement based on the F-statistic. Backward elimination starts with all the predictors in the model and then constantly iterates to remove the least significant variables until only the most significant predictors remain.

All the significant variables from model 1 are in model 3: *zn*, *nox*, *age*, *dis*, *rad*, *tax*, *ptratio*, *medv*

We see the absence of variables such as indus, chas and lstat.

We can also see the AIC of 215.3229 and BIC of 252.6205 is lowest amongst all models, making this the most ideal model.

```
##
## Call:
## glm(formula = target ~ zn + nox + age + dis + rad + tax + ptratio +
##     medv, family = "binomial", data = training_data)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -37.415922   6.035013  -6.200 5.65e-10 ***
## zn           -0.068648   0.032019  -2.144  0.03203 *
## nox          42.807768   6.678692   6.410 1.46e-10 ***
## age           0.032950   0.010951   3.009  0.00262 **
## dis           0.654896   0.214050   3.060  0.00222 **
## rad           0.725109   0.149788   4.841 1.29e-06 ***
## tax          -0.007756   0.002653  -2.924  0.00346 **
## ptratio       0.323628   0.111390   2.905  0.00367 **
## medv          0.110472   0.035445   3.117  0.00183 **
## ---
## 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: 197.32  on 457  degrees of freedom
## AIC: 215.32
```

```
##
## Number of Fisher Scoring iterations: 9
```

```
## # A tibble: 1 × 8
##   null.deviance df.null logLik   AIC   BIC deviance df.residual  nobs
##           <dbl>   <int>  <dbl> <dbl> <dbl>    <dbl>       <int> <int>
## 1          646.     465  -98.7  215.  253.     197.         457   466
```

## /// Select Model

### Model Metrics Table

```
## fitting null model for pseudo-r2
```

```
## fitting null model for pseudo-r2
```

```
## fitting null model for pseudo-r2
```

| model.build | null.deviance | df.null | logLik | AIC | BIC | deviance | df.residual | nobs | McF |
|---|---|---|---|---|---|---|---|---|---|
| model 1 | 645.8758 | 465 | -96.02346 | 218.0469 | 271.9213 | 192.0469 | 453 | 466 | 0.70 |
| model 2 | 645.8758 | 465 | -95.08023 | 220.1605 | 282.3232 | 190.1605 | 451 | 466 | 0.70 |
| model 3 | 645.8758 | 465 | -98.66143 | 215.3229 | 252.6205 | 197.3229 | 457 | 466 | 0.69 |

### Deviance

The deviance is a measure of how well the model fits the data.The values are large for all three models, we cannot directly conclude a relationship.

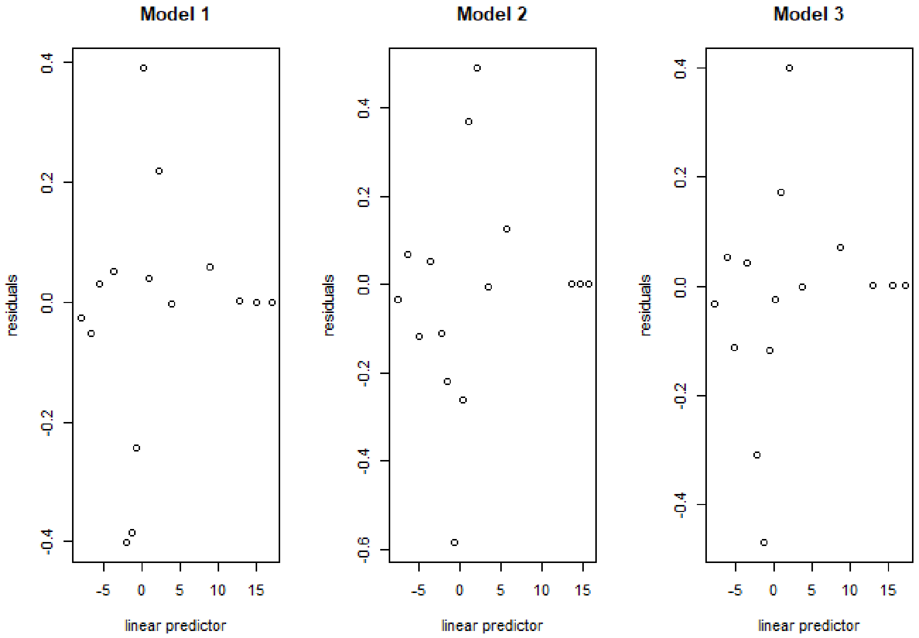
```
## [1] "Model 1 p-val is: 0.4911638%"
```

```
## [1] "Model 2 p-val is: 0.4911442%"
```

```
## [1] "Model 3 p-val is: 0.4912026%"
```

### Residuals for Logistics Regression

Similar residual variance is seen for all three models. Since the deviance residuals seem to have mostly even variation across all models we will accept the models as adequate for our purposes. If our goal was to explain which predictors contribute to the response variable, we would like to much more evenly distributed variance of residuals.



### AIC/BIC

Model 3 has the lowest values for AIC and BIC. The lower the value, the better fit the model has. The lower score for model 3 is a good sign which explains that goodness of fit is better than in Model 1 and Model 2. Both AIC and BIC rewards goodness of fit, but penalizes a model as the number of predictors increase. Model 3 has the least amount of predictors which checks out with having a lower complexity criterion. Given that the goal for this model is prediction, the preferred metric will be AIC. If the goal for this model had been to explain the response, BIC metric would be the preferred choice.

### Goodness of Fit

The Homer-Lemeshow goodness of fit test is used to assess how a binary logistic regression model fits the observed data. Our results show that model 1 and model 2 are not a good fit. This is observed by the low p-value. The null hypothesis states that there is no difference between the observed and expected frequencies across the groups. In other words, the logistic regression model fits the data well.

When observing the plot correlation between probability and the proportion of said event occurring, model 3 displays that all results align. This further confirms model 3 as a model of choice.

```
#model1
performance_hosmer(fit1, n_bins = 15)
```

```
## # Hosmer-Lemeshow Goodness-of-Fit Test
##
##   Chi-squared: 23.707
##            df: 13
##       p-value:  0.034
```

```
## Summary: model does not fit well.
```

```
#model2
performance_hosmer(fit2, n_bins = 15)
```

```
## # Hosmer-Lemeshow Goodness-of-Fit Test
## 
##   Chi-squared: 66.391
##            df: 13
##       p-value:  0.000
```

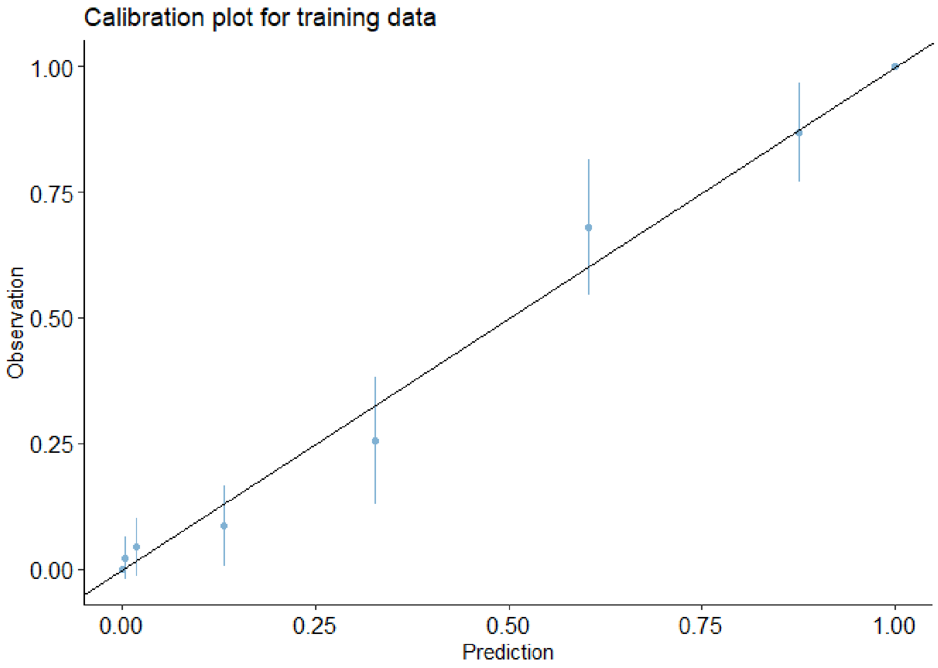
```
## Summary: model does not fit well.
```

```
#model3
performance_hosmer(fit3, n_bins = 15)
```

```
## # Hosmer-Lemeshow Goodness-of-Fit Test
## 
##   Chi-squared: 16.335
##            df: 13
##       p-value:  0.232
```

```
## Summary: model seems to fit well.
```

```
## $calibration_plot
```



**McFadden’s Pseudo-R2** All three models scored around 70% with model 3 having a slightly lower score. Nothing of concern is revealed by this metric. Although the score is lower, We will continue to accept model 3 as the top candidate. Again, evaluation metrics such as AIC are considered better for predictive purposes.

### Model 3 Evaluation

The binary logistic regression model achieved a score of 97.19% in AUC. The closer to 100% the better predictive power a model has. Additionally, the model scored a 91.20% in accuracy. Given that the training_set was well balanced between both categorical responses, we can disregard F1-score as a measure of performance. Had the model been imbalanced, the F1-score provides a harmonic mean score between precision and sensitivity, therefore providing a more realistic evaluation.

Generally speaking, the goal is to get the highest AUC possible as this indicates a high true positive rate (sensitivity) and a high true negative rate (specificity). However, McFadden's Pseudo-R-square being <70% and the deviance being high, this could indicate a case of overfitting. Overfitting occurs when the model learns the training data too well. To get a better sense if overfitting is occurring it could be sensible to employ cross-validation to evaluate multiple subsets of the training data.

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
## [1] "Model 3 Classification Accuracy is: 91.20%"
```

```
## [1] "Model 3 Classification Error Rate is: 8.80%"
```

```
## [1] "Model 3 Precision is: 90.83%"
```

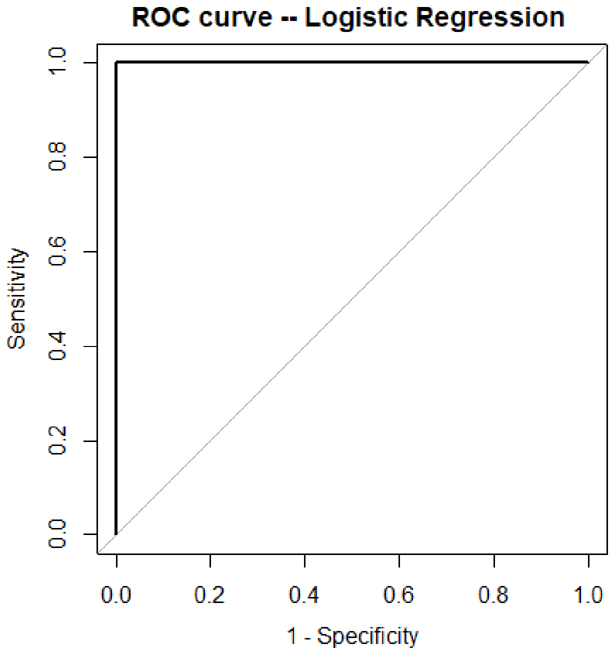
```
## [1] "Model 3 Sensitivity/Recall is: 91.98%"
```

```
## [1] "Model 3 Specificity is: 90.39%"
```

```
## [1] "Model 3 F1-score is: 91.40%"
```

```
## [1] "Model 3 AUC is: 97.19%"
```

**ROC Curve**



**Making Predictions** The model yielded a 20/20 split. If the test data has a distribution reflecting that of the testing data, it is sensible to assume the model has significant predictive capabilities.

| zn | indus | chas | nox | rm | age | dis | rad | tax | ptratio | lstat | medv | pred_prob | predout |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 7.07 | 0 | 0.469 | 7.185 | 61.1 | 4.9671 | 2 | 242 | 17.8 | 4.03 | 34.7 | 0.0518733 | 0 |
| 0 | 8.14 | 0 | 0.538 | 6.096 | 84.5 | 4.4619 | 4 | 307 | 21.0 | 10.26 | 18.2 | 0.6563639 | 1 |
| 0 | 8.14 | 0 | 0.538 | 6.495 | 94.4 | 4.4547 | 4 | 307 | 21.0 | 12.80 | 18.4 | 0.7292303 | 1 |
| 0 | 8.14 | 0 | 0.538 | 5.950 | 82.0 | 3.9900 | 4 | 307 | 21.0 | 27.71 | 13.2 | 0.4263767 | 0 |
| 0 | 5.96 | 0 | 0.499 | 5.850 | 41.5 | 3.9342 | 5 | 279 | 19.2 | 8.77 | 21.0 | 0.1075746 | 0 |
| 25 | 5.13 | 0 | 0.453 | 5.741 | 66.2 | 7.2254 | 8 | 284 | 19.7 | 13.15 | 18.7 | 0.3126897 | 0 |
| 25 | 5.13 | 0 | 0.453 | 5.966 | 93.4 | 6.8185 | 8 | 284 | 19.7 | 14.44 | 16.0 | 0.3879178 | 0 |
| 0 | 4.49 | 0 | 0.449 | 6.630 | 56.1 | 4.4377 | 3 | 247 | 18.5 | 6.53 | 26.6 | 0.0139910 | 0 |
| 0 | 4.49 | 0 | 0.449 | 6.121 | 56.8 | 3.7476 | 3 | 247 | 18.5 | 8.44 | 22.2 | 0.0056513 | 0 |
| 0 | 2.89 | 0 | 0.445 | 6.163 | 69.6 | 3.4952 | 2 | 276 | 18.0 | 11.34 | 21.4 | 0.0018602 | 0 |

| Var1 | Freq |
|---|---|
| 0 | 20 |
| 1 | 20 |

## // Appendix

```
#importing data sets
library(MASS)
library(tidyverse)
library(ggpubr)
library(corrplot)
library(broom)
library(pscl)
library(lmtest)
library(performance)
library(PredictABEL)
library(predtools)
library(caret)
library(pROC)

training_data<-
         read.csv("https://raw.githubusercontent.com/Vy4thewin/criticalthinking3/main/crime-
         training-data_modified.csv")
testing_data<-
         read.csv("https://raw.githubusercontent.com/Vy4thewin/criticalthinking3/main/crime-
         evaluation-data_modified.csv")
#count of training set
sprintf("Number of observations: %1.f",count(training_data))

#convert chas and target cols to factor
#training_data$chas <-as.factor(training_data$chas)
#training_data$target <-as.factor(training_data$target)

#summary of the data set
summary(training_data)

#double checking there's no NAs
colSums(is.na(training_data))
#Look at the distribution of all predictor variables
g1<-training_data%>%ggplot(aes(x=zn))+geom_histogram(bins=25)+theme_classic()
g2<-training_data%>%ggplot(aes(x=indus))+geom_histogram(bins=25)+theme_classic()
#g3<-training_data%>%ggplot(aes(x=chas))+geom_histogram(bins=25)+theme_classic()
g4<-training_data%>%ggplot(aes(x=nox))+geom_histogram(bins=25)+theme_classic()
g5<-training_data%>%ggplot(aes(x=rm))+geom_histogram(bins=25)+theme_classic()
g6<-training_data%>%ggplot(aes(x=age))+geom_histogram(bins=25)+theme_classic()
g7<-training_data%>%ggplot(aes(x=dis))+geom_histogram(bins=25)+theme_classic()
g8<-training_data%>%ggplot(aes(x=rad))+geom_histogram(bins=25)+theme_classic()
g9<-training_data%>%ggplot(aes(x=tax))+geom_histogram(bins=25)+theme_classic()
g10<-training_data%>%ggplot(aes(x=ptratio))+geom_histogram(bins=25)+theme_classic()
g11<-training_data%>%ggplot(aes(x=lstat))+geom_histogram(bins=25)+theme_classic()
g12<-training_data%>%ggplot(aes(x=medv))+geom_histogram(bins=25)+theme_classic()
ggarrange(g1,g2,g4,g5,g6,g7,g8,g9,g10,g11,g12,nrow = 3,ncol = 4)
#checking for multi-collinearity
drop <- c("chas","target")
corrplot(cor(training_data[,!(names(training_data) %in% drop)]),method =
         "number",type="lower", tl.srt = .71,number.cex=0.75)
#Creating a polynomial term from zn for a more stable predictor variable
train.set<-training_data%>%mutate_at(c(1),~poly(.,2))

#seeing the difference in distribution
g1<-training_data%>%ggplot(aes(x=zn))+geom_density()+theme_classic()+ ggtitle("Before
         Transformation")
g2<-train.set%>%ggplot(aes(x=zn[,"2"]))+geom_density()+theme_classic()+ ggtitle("After
         Transformation")
ggarrange(g1,g2)
```

```r
#Performing box cox on the predictors and retrieving their lambdas
lamb.indus<-boxcox(training_data$indus~1)
lamb.nox<-boxcox(training_data$nox~1)
lamb.pratio<-boxcox(training_data$ptratio~1)
lamb.tax<-boxcox(training_data$tax~1)

#Extracting highest value of lambda from the box-cox result lists
lamb.indus<-lamb.indus$x[which.max(lamb.indus$y)]
lamb.nox<-lamb.nox$x[which.max(lamb.nox$y)]
lamb.pratio<-lamb.pratio$x[which.max(lamb.pratio$y)]
lamb.tax<-lamb.tax$x[which.max(lamb.tax$y)]

#indus is near .5 so it will need sqrt(x)
train.set<-train.set%>%mutate(indus=sqrt(indus))

#nox is near -1 so it will get 1/x
train.set<-train.set%>%mutate(nox=1/nox)

#pratio will be ploy like zn
train.set<-train.set%>%mutate(ptratio=poly(ptratio,2))

#tax will get 1/ sqrt(x)
train.set<-train.set%>%mutate(tax=1/sqrt(tax))
# Baseline model
fit1 <- glm(target ~ ., data = training_data, family = "binomial")

summary(fit1)
glance(fit1)
# Our second logistic regression model with transformed variables
fit2 <- glm(target ~ ., data = train.set, family = "binomial")

# Summary of the second model
summary(fit2)
glance(fit2)
# The third model with stepwise regression
fit3 <- stepAIC(fit1, direction = "both", trace = FALSE)

# Summary of the third model
summary(fit3)
glance(fit3)
#Calc McFaddens pseudo r^2 for each model
pseudo_r2.m1 <- pR2(fit1, method = "mcfadden")
pseudo_r2.m2 <- pR2(fit2, method = "mcfadden")
pseudo_r2.m3 <- pR2(fit3, method = "mcfadden")
mcfads_vals <- c(pseudo_r2.m1[4], pseudo_r2.m2[4], pseudo_r2.m3[4])

model_res <- bind_rows(glance(fit1), glance(fit2), glance(fit3))
model_names <- c("model 1","model 2","model 3")
model_res <- cbind(model.build = model_names, model_res)
model_res <- cbind(model_res,McFaddens.R2 = mcfads_vals)

knitr::kable(model_res, "pipe")

# # Predict the probability (p) of crime
# probabilities <- predict(fit3, type = "response")
# predicted.classes <- ifelse(probabilities < 0.5, 0, 1)
#
```

```
# # Select only numeric predictors
# num_predictors <- training_data %>%
#   dplyr::select_if(is.numeric) |>
#   select(-c(chas, target))
# predictors <- colnames(num_predictors)
#
# # Bind the logit and tidying the data for plot
# num_predictors <- num_predictors %>%
#   mutate(logit = log(probabilities/(1-probabilities))) %>%
#   gather(key = "predictors", value = "predictor.value", -logit)
#
# #Create Scatter plots
# ggplot(num_predictors, aes(logit, predictor.value))+
#   geom_point(size = 0.5, alpha = 0.5) +
#   geom_smooth(method = "loess") +
#   theme_bw() +
#   facet_wrap(~predictors, scales = "free_y")
#the p-value for the test of the hypothesis that at least one of the predictors is related
         to the response.
#Values are large for all models, we cannot directly conclude a relationship.
sprintf("Model 1 p-val is: %.7f%%",(1-pchisq(model_res[["df.residual"]]
         [1],fit1$df.residual)))  #model1
sprintf("Model 2 p-val is: %.7f%%",(1-pchisq(model_res[["df.residual"]]
         [2],fit2$df.residual)))  #model2
sprintf("Model 3 p-val is: %.7f%%",(1-pchisq(model_res[["df.residual"]]
         [3],fit3$df.residual)))  #model3

 #model3
par(mfrow = c(1, 3))

#model 1
resid.df1 <- mutate(training_data, residuals=residuals(fit1), linpred=predict(fit1))
gdf1 <- group_by(resid.df1, cut(linpred, breaks=unique(quantile(linpred,(1:15)/16))))
diagdf1 <- summarise(gdf1, residuals=mean(residuals), linpred=mean(linpred))
plot(residuals ~ linpred, diagdf1, xlab="linear predictor", main="Model 1")

#model 2
resid.df2 <- mutate(train.set, residuals=residuals(fit2), linpred=predict(fit2))
gdf2 <- group_by(resid.df2, cut(linpred, breaks=unique(quantile(linpred,(1:15)/16))))
diagdf2 <- summarise(gdf2, residuals=mean(residuals), linpred=mean(linpred))
plot(residuals ~ linpred, diagdf2, xlab="linear predictor", main="Model 2")

#model 3
resid.df3 <- mutate(train.set, residuals=residuals(fit3), linpred=predict(fit3))
gdf3 <- group_by(resid.df3, cut(linpred, breaks=unique(quantile(linpred,(1:15)/16))))
diagdf3 <- summarise(gdf3, residuals=mean(residuals), linpred=mean(linpred))
plot(residuals ~ linpred, diagdf3, xlab="linear predictor", main="Model 3")

#model1
performance_hosmer(fit1, n_bins = 15)

#model2
performance_hosmer(fit2, n_bins = 15)

#model3
performance_hosmer(fit3, n_bins = 15)
#plotCalibration(data = training_data, cOutcome = 13, predRisk = fitted(fit3), groups= 15)
         #target found in column 13
```

```r
training_data$pred <- predict.glm(fit3, type = 'response')
calibration_plot(data = training_data, obs = "target", pred = "pred", title = "Calibration
         plot for training data")
training_data <- mutate(training_data, predout=ifelse(pred < 0.5, 0, 1))

#Create confusion matrix
cm <- confusionMatrix(as.factor(training_data$predout), as.factor(training_data$target))

#Calculate AUC
auc_res <- auc(training_data$target, training_data$pred)

sprintf("Model 3 Classification Accuracy is: %.2f%%",(cm$overall[1])*100)
sprintf("Model 3 Classification Error Rate is: %.2f%%",(1-cm$overall[1])*100)
sprintf("Model 3 Precision is: %.2f%%",(cm$byClass['Pos Pred Value']*100))
sprintf("Model 3 Sensitivity/Recall is: %.2f%%",(cm$byClass['Sensitivity']*100))
sprintf("Model 3 Specificity is: %.2f%%",(cm$byClass['Specificity']*100))
sprintf("Model 3 F1-score is: %.2f%%",(cm$byClass['F1']*100))
sprintf("Model 3 AUC is: %.2f%%",(auc_res[1]*100))
par(pty="s")
roc_score= roc(training_data$predout, fit3$fitted.values) #AUC score
plot(roc_score ,main ="ROC curve -- Logistic Regression ", legacy.axes=TRUE)



# true_labels <- training_data$predout
# roc_curve <- roc(true_labels, predicted_probabilities)
# plot(roc_curve, main = "ROC Curve", col = "blue", lwd = 2)
# abline(a = 0, b = 1, col = "gray", lty = 2)
# legend("bottomright", legend = paste("AUC =", round(auc(roc_curve), 2)), col = "blue",
         lwd = 2)
testing_data$pred_prob <- predict(fit3, testing_data, type="response")
testing_data <- mutate(testing_data, predout=ifelse(pred_prob < 0.5, 0, 1))

knitr::kable(head(testing_data,10) , "pipe")

knitr::kable(table(testing_data$predout), "pipe")
```