Homework #3: Logistic Regression

CUNY SPS DATA 621 Spring 2019

Group # 4

April 10, 2019

Contents

1 Objective

The the goal of this report is to develop a binary logistic regression model which can predict whether a neighborhood will be at risk for high crime levels.

2 Data exploration

The major city is not explicitly stated in the data description does have a variables `chas` for whether a neighborhood borders the Charles River. This suggests that the data comes from the Boston metropolitan area, even though we cannot use data outside of what was provided it does provide additional perspective when evaluating relationships across variables.

2.1 Data dictionary

The table below table below describes the variables in the dataset.

variable	description
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)

2.2 Summary statistics

The training data set contains 466 observations and 13 variables including the response variable, `target`. Reviewing the summary statistics show that there are no missing values, of the 13 variables in the data set 12 are numeric and both `chas` and the response variable `target` are categorical. The mean of `target` shows that there are slightly more neighborhoods with crime rates above the median which suggests that the underlying data may have had a slighly negative skew, but since the mean is relatively close to 0.5 there no special treatment such as sampling is required before classification can be implemented.

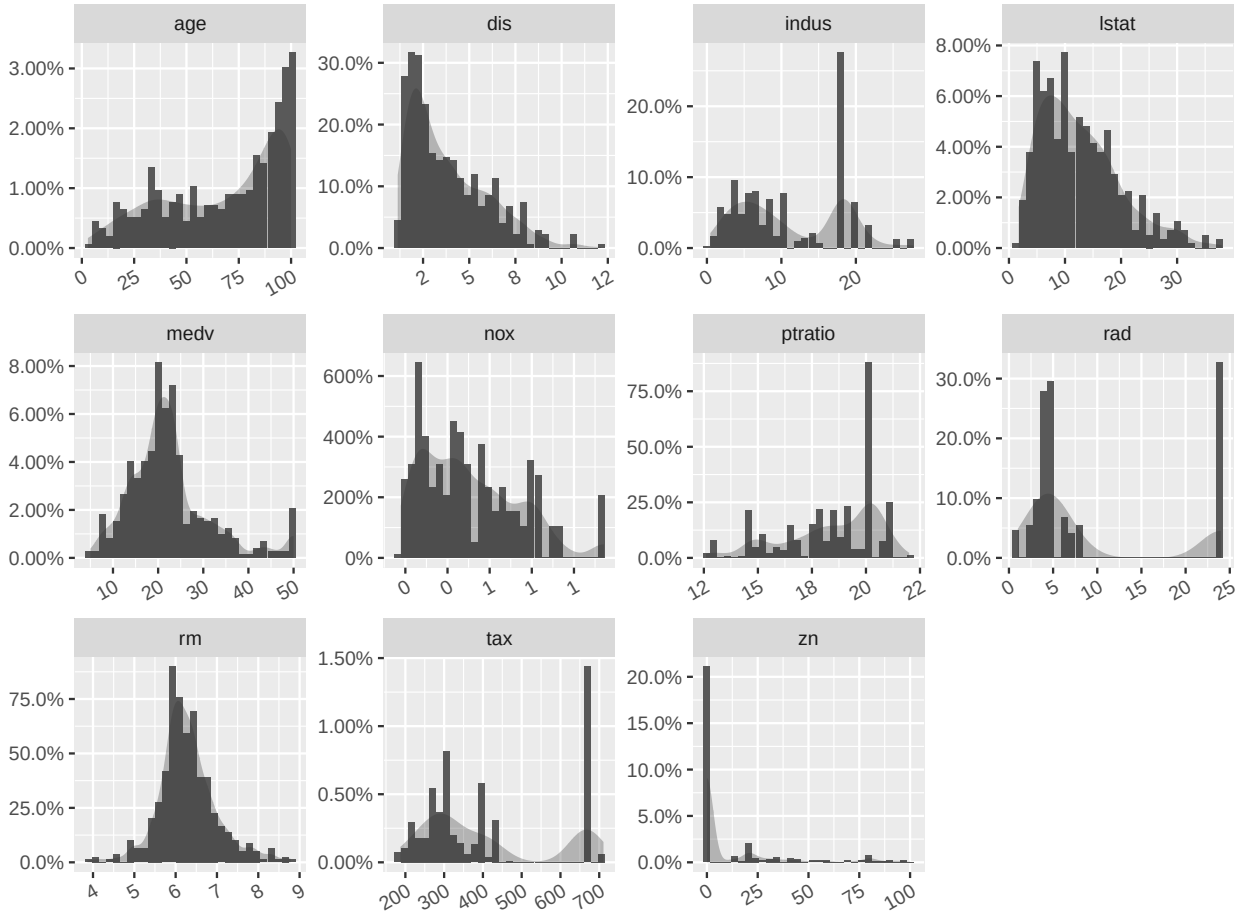
metric	min	Q1	mean	median	Q3	max
age	2.9	43.88	68.37	77.15	94.1	100
chas	0	0	0.07082	0	0	1
dis	1.13	2.101	3.796	3.191	5.215	12.13
indus	0.46	5.145	11.11	9.69	18.1	27.74
lstat	1.73	7.043	12.63	11.35	16.93	37.97
medv	5	17.02	22.59	21.2	25	50
nox	0.389	0.448	0.5543	0.538	0.624	0.871
ptratio	12.6	16.9	18.4	18.9	20.2	22
rad	1	4	9.53	5	24	24
rm	3.863	5.887	6.291	6.21	6.63	8.78
target	0	0	0.4914	0	1	1
tax	187	281	409.5	334.5	666	711
zn	0	0	11.58	0	16.25	100

metric	missing	sd	mad	skewness	kurtosis
age	0	28.32	30.02	-0.5777	-1.01
chas	0	0.2568	0	3.335	9.145
dis	0	2.107	1.914	0.9989	0.472
indus	0	6.846	9.34	0.2885	-1.243
lstat	0	7.102	7.072	0.9056	0.5034
medv	0	9.24	6.005	1.077	1.374
nox	0	0.1167	0.1334	0.7463	-0.03577
ptratio	0	2.197	1.927	-0.7543	-0.4004
rad	0	8.686	1.483	1.01	-0.8619
rm	0	0.7049	0.5167	0.4793	1.542
target	0	0.5005	0	0.03423	-2.003
tax	0	167.9	104.5	0.6593	-1.148
zn	0	23.36	0	2.177	3.814

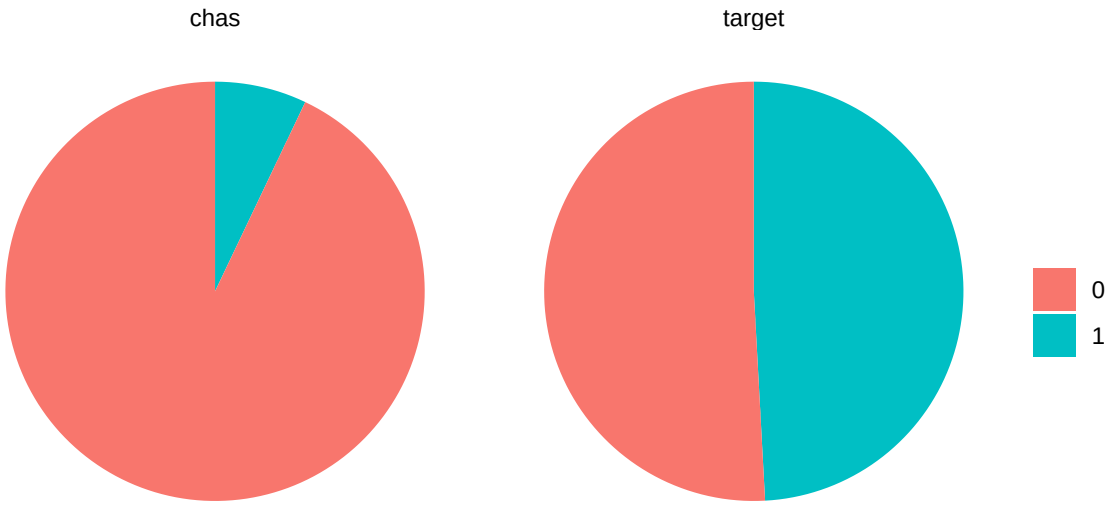
2.3 Visualizations

2.3.1 Univariate distributions

The plot below shows histograms and density plots for each numeric variable in the data. The visualizations show that some variables such as `age`, `dis`, and `lstat` are skewed and may benefit from transformation. It also shows that several variables have a mode which far far exceeds other values in the data. Also `rad` appears to be multimodal. For these variables it may be beneficial to create dichotomous categorical variables instead of using the continuous values.

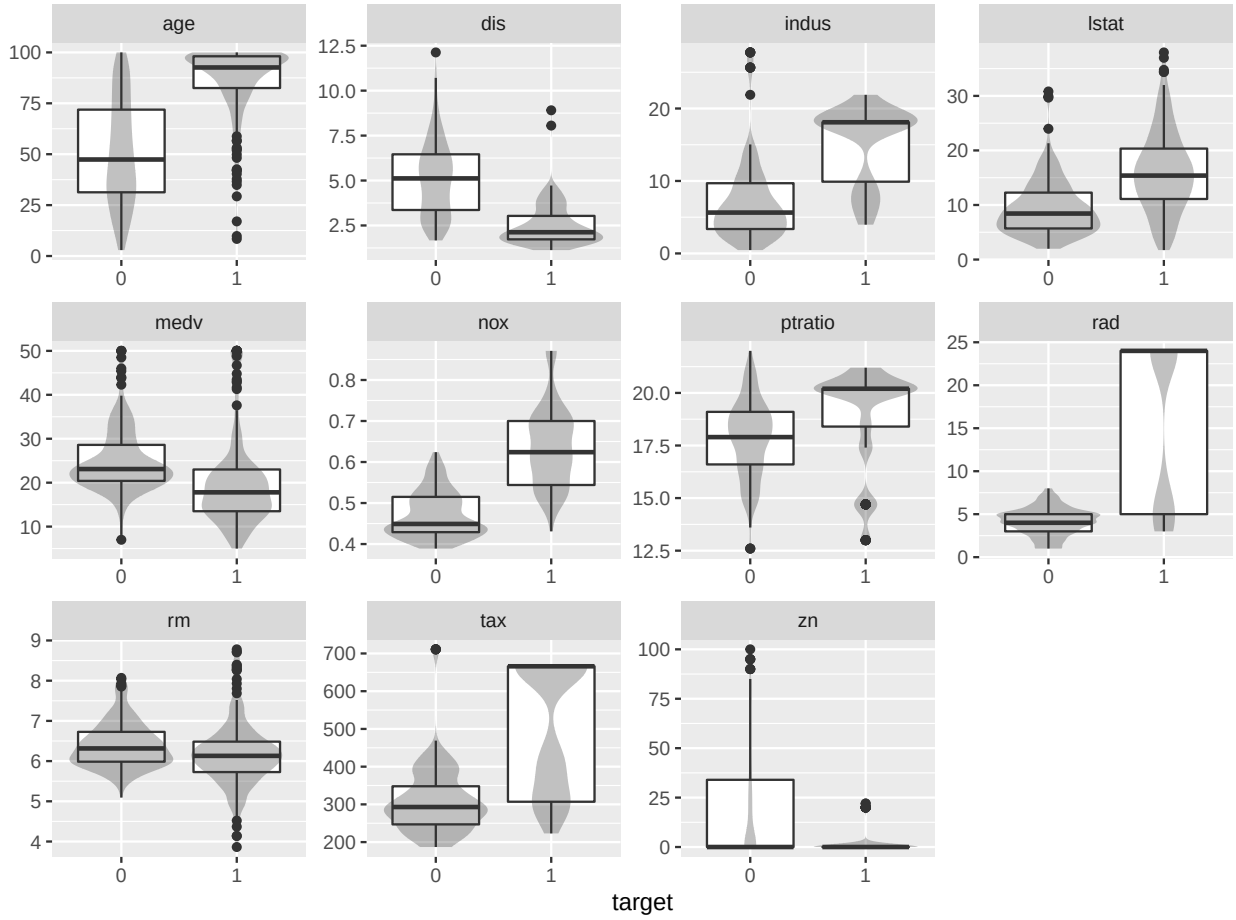


The pie charts below reiterate the findings from the summary statistics for the two dichotomous variables; `target` is relatively evenly split while `chas` shows that that the majority of neighborhoods do not border Charles River.

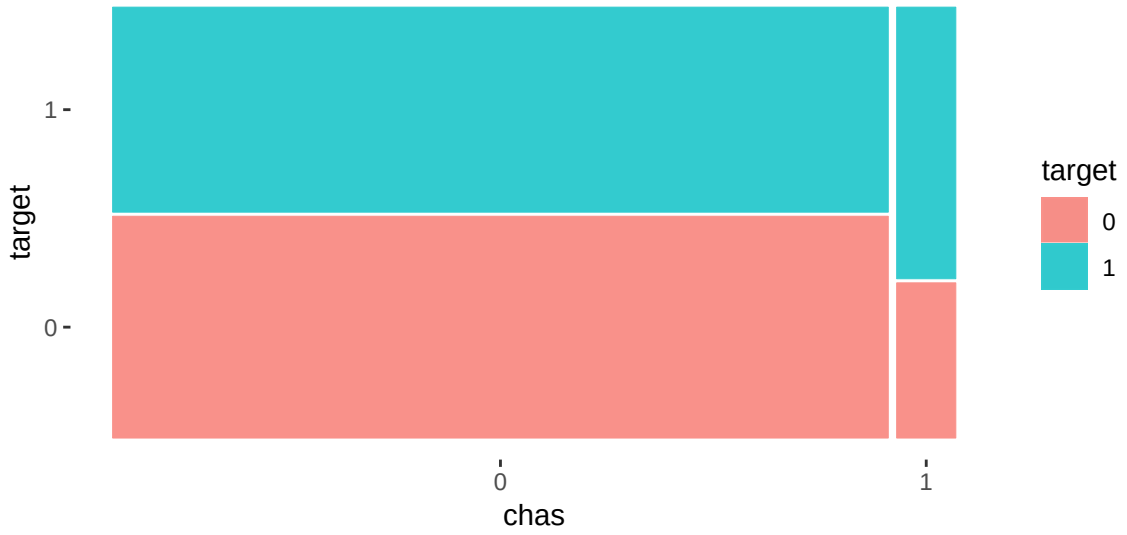


2.3.2 Bivariate relationships

In the plot below the data has been split by the response variable; each plot contains two boxplots with a violin plot overlayed to show the distribution of each predictor for both classifications of `target`. The plot shows that areas with older housing are more likely to have crime rates above the median and that for `rad` and `tax` appear to be multimodal in the higher than median crime rate group.

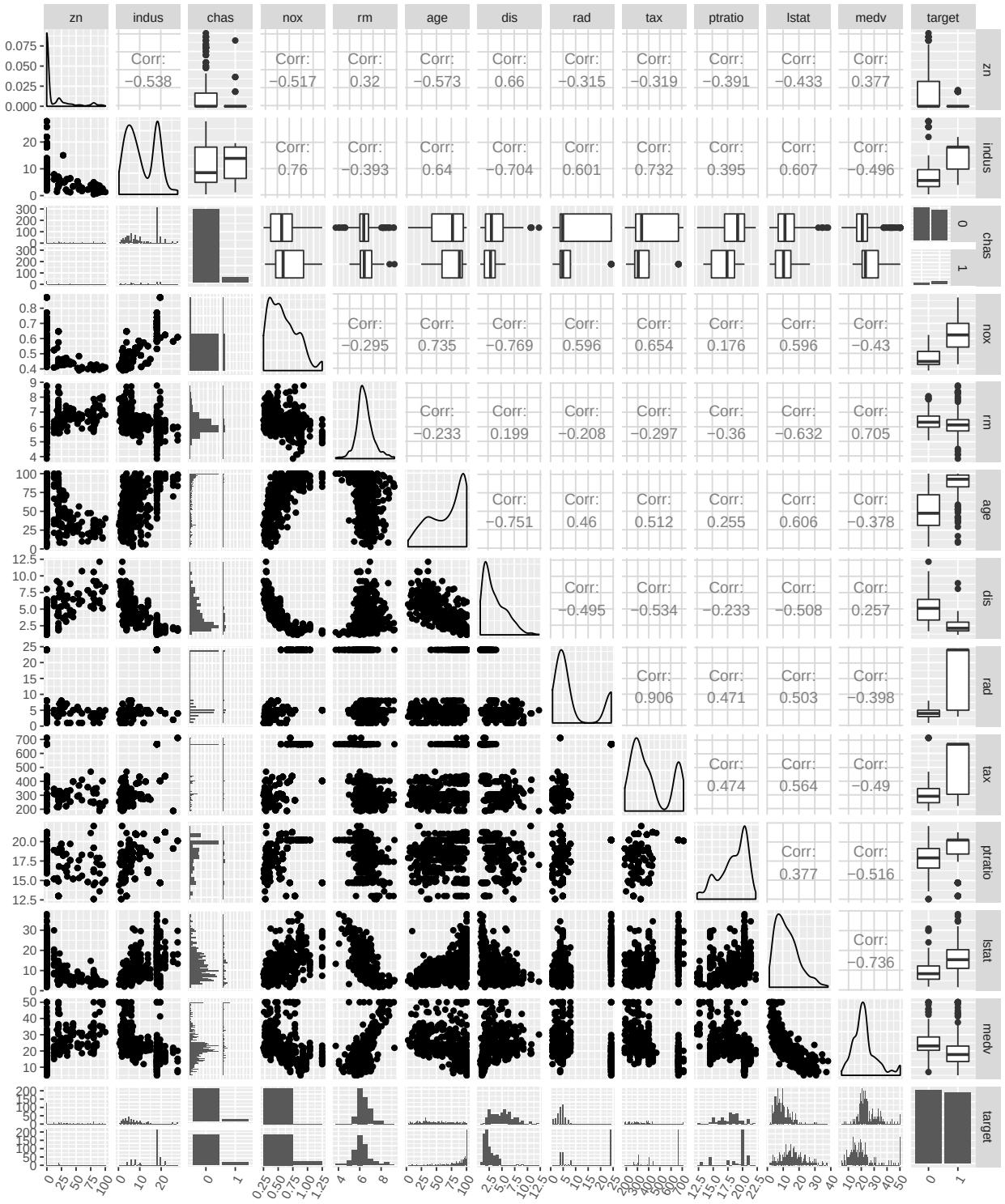


The mosaic plot below shows the relationship between `chas` and `target`. The width of the bars shows the proportion of each categorization of `chas` while the height and color show the proportion of `target`. As seen previously neighborhoods which do not border Charles River are the majority, but it does show that those that do border Charles River are more likely to be in a neighborhood with a higher than median crime rate.



2.3.2.1 Pairs plot

Previously the visualizations have focused on the relationship with `target`. The pairs plot provides and idea of the relationships between all variables in the data. A Few notable relationships are between `nox` and `dis` and `medv` and `lstat`. In the prior higher levels of nitrogen oxide show a polynomial relationship with distance to a Boston Employment center. It is unlikely that these two variables have a direct relationship, but a higher level of `nox` and shorter `dis` could be indicative of higher population density compared to other regions. Similarly `medv` and `lstat` show a polynomial relationship. Unlike `nox` and `dis` this could have a more direct relationship as areas with fewer lower status people can directly impact the median house prices. It is also shown that neighborhoods near Charles River have shorter distances to Empolyment centers, `dis`, and more people of lower status.



2.3.2.2 Multicollinearity

The pairs plot showed that a number of predictors had very strong Pearson correlation coefficients, $|r| > 0.75$, which could suggest multicollinearity in the data. As a preliminary test a logit link logistic regression was produced using all the predictors originally provided in the data and the VIF was calculated on the resulting model. No VIF is greater than 10 that there isn't an inherent need to address multicollinearity, but a few are

above 3 which could mean that these variables may benefit from modification.

variable	vif
zn	1.823
indus	2.682
chas	1.241
nox	4.16
rm	5.814
age	2.57
dis	3.888
rad	1.943
tax	2.144
ptratio	2.276
lstat	2.643
medv	8.122

3 Data preparation

3.1 Missing values

Both the training and evaluation data set are complete, no missing observations.

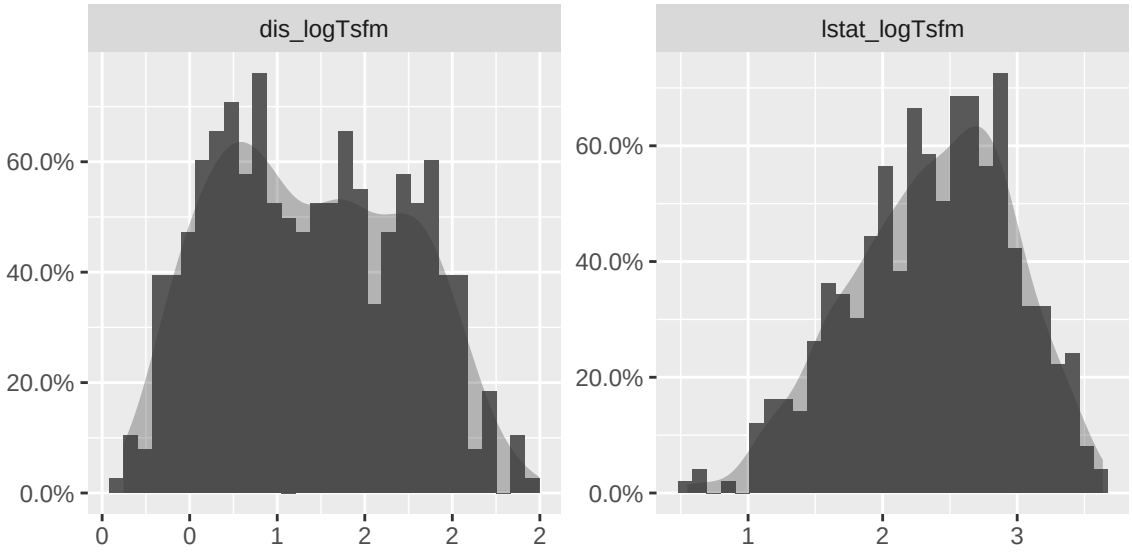
3.2 Variable creation

Both `rad` and `tax` appear to have bimodal distributions and the box plots grouped by `target` suggest that the data around the lower mode is typically associated with areas that have lower than median crime rates. `indus` and `age` do not show as clear a divide as the other variables, but do have a similar spike at higher values. For all three variables a binary variable will be created by splitting based on the visualization.

3.3 Transformations

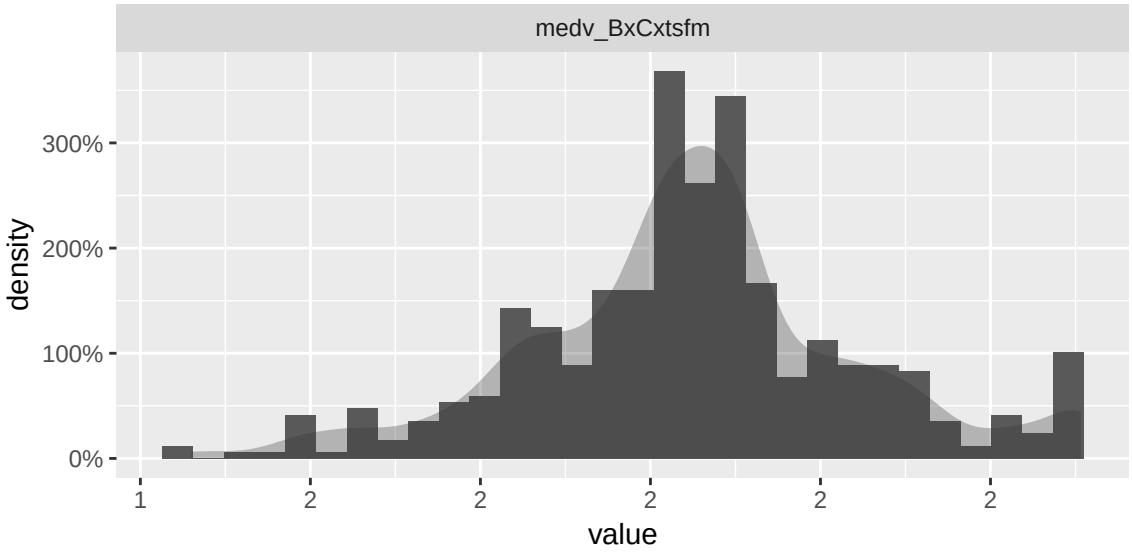
3.3.1 Log

The summary statistics and visualization showed that the variables `dis` and `lstat` have a positive skew. In both the skew is in the range of ± 1, given the number of observations in the data a transformation is not necessary for modeling, but would simplify the interperetation of the β for these predictors.



3.3.2 Box-Cox

The summary statistics and historgram for `medv` show a bit of skew and kurtosis. It is small enough that it may not negatively impact its use as a predictor, however for purposes of experiementation in model development it has been transformed. Initially a log transformation was attempted, but in reviewing the transformed distribution a Box-Cox transformation showed better results.



3.4 Outliers

The histogram and box-plots suggest show that there are a few data points which could be considered outliers. Some of these outliers may be explainable with additional context from external data sources, however the project description prohibts this method. In order to handle potential outliers winsorizing will be employed. Any values outside outside of the [5%, 95%] range will be substitiuted witht the value for 5% or 95% respectively.

4 Build models

For purposes of model development and evaluation the training data is split 70/30 with 70% of the data being used for model development, and 30% being used to evaluate the model.

4.1 Model 1: logit original predictors

This classification model makes use of a logit link and makes use of all the predictors originally provided from the data set. Reviewing the model we see that the β for `nox` is fairly large,as the parts per million of nitrogen oxide the more likely a neighborhood is to have an above median crime rate. This type of pollution can come from vehicles and could be suggestive of a more densly populated area, while not always the case that could, logically, result in a higher crime rate. Similarly we see that neighborhoods with more plots zoned for larger housing, houses with more rooms, higher taxes, more expensive houses, and less industry are less likely to have an above median crime rate. Using the McFadden R^2 as mechanism to evaluate how well the model fits the training data we see that the full model fits reasonably well. However the full model contains a number of predictors which are not statistically significant so there is the potential for improvement.

```
##
## Call:
## glm(formula = target ~ . - index, family = binomial(link = "logit"),
##     data = dfMDLTrain[, -grep("_", colnames(dfMDLTrain))])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7447  -0.0635   0.0000   0.0007   3.4827
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -58.508219  11.367216  -5.147 2.65e-07 ***
## zn           -0.186382   0.070029  -2.662 0.007779 **
## indus        -0.035947   0.083480  -0.431 0.666759
## chas1         0.272140   1.080790   0.252 0.801198
## nox          64.128256  13.360827   4.800 1.59e-06 ***
## rm           -1.523001   1.043120  -1.460 0.144278
## age           0.081346   0.023169   3.511 0.000447 ***
## dis           1.437761   0.386888   3.716 0.000202 ***
## rad           0.965266   0.276850   3.487 0.000489 ***
## tax          -0.008182   0.004051  -2.020 0.043381 *
## ptratio       0.583500   0.193787   3.011 0.002604 **
## lstat         0.043518   0.077331   0.563 0.573601
## medv          0.412227   0.115662   3.564 0.000365 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 451.63  on 325  degrees of freedom
## Residual deviance: 110.61  on 313  degrees of freedom
## AIC: 136.61
##
## Number of Fisher Scoring iterations: 9
```

```
##  McFadden
## 0.7550798
```

4.2 Model 2: logit original predictor stepAIC feature selection

This model builds on the original predictor logit by using backward and forward stepwise selection to choose predictors. The `stepAIC` function from the `MASS` package makes use of the Akaike Information Criterion (AIC) for model selection. Compared to the all predictor logit the AIC is slightly lower suggesting an increase in model perfomance; this is further supported by a higher McFadden R^2. While the coefficients are different the relationships between whether an increase in predictor mean a neighborhood is more/less likely to have a higher than median crime rate are the same.

```
##
## Call:
## glm(formula = target ~ zn + nox + rm + age + dis + rad + tax +
##     ptratio + medv, family = binomial(link = "logit"), data = dfMDLTrain[,
##     -grep("_", colnames(dfMDLTrain))])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7837  -0.0640   0.0000   0.0004   3.5085
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -56.694275  11.048691  -5.131 2.88e-07 ***
## zn           -0.183999   0.062326  -2.952 0.003155 **
## nox          62.005312  11.854277   5.231 1.69e-07 ***
## rm           -1.756181   0.976298  -1.799 0.072048 .
## age           0.087330   0.021722   4.020 5.81e-05 ***
## dis           1.455132   0.387932   3.751 0.000176 ***
## rad           1.025354   0.249700   4.106 4.02e-05 ***
## tax          -0.008802   0.003727  -2.361 0.018202 *
## ptratio       0.602326   0.193550   3.112 0.001858 **
## medv          0.413034   0.114009   3.623 0.000291 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 451.63  on 325  degrees of freedom
## Residual deviance: 111.13  on 316  degrees of freedom
## AIC: 131.13
##
## Number of Fisher Scoring iterations: 9
```

```
##  McFadden
## 0.7539345
```

4.3 Model 3: logit original and derived predictor stepAIC feature selection

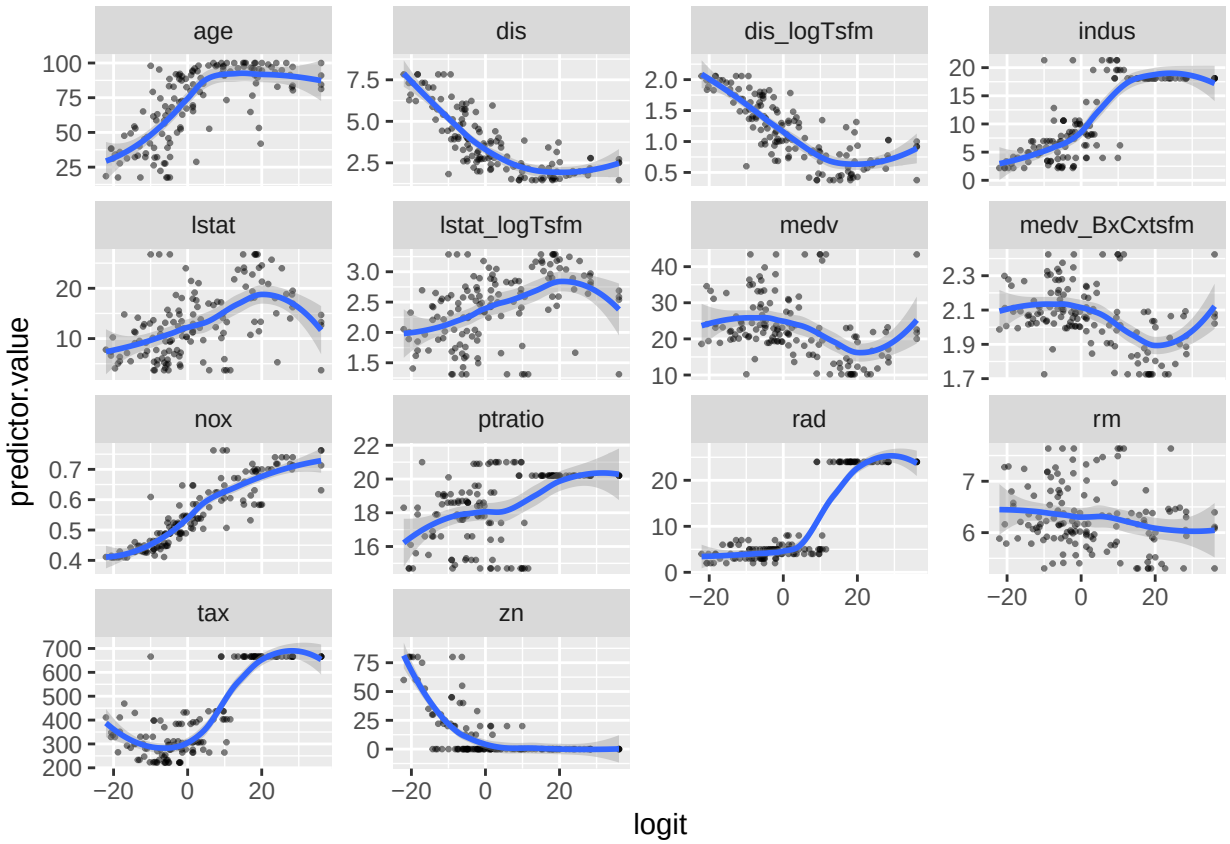
In the exploratory data analysis several predictors could benefit from transformation and the pairs plot and vif suggested that some variables may have some multicollinearity. As described in the data preparation section several variables were transformed and dichotomous predictors were created. This model makes use of `stepAIC` building off a full model including both the original, transformed, and derived variables. Comparing the AIC and McFadden R^2 suggests a slightly more performant model. One confound is that stepAIC included both the original and log-transformed version of `lstat`. This could be indicative of interaction or multicollineartiy between with another predictor, however only one version of `lstat` should be included.

```
##
## Call:
## glm(formula = target ~ indus + nox + rm + age + dis + rad + ptratio +
##     lstat + medv + hghTax_drv + hghIndus_drv + dis_logTsfm +
##     lstat_logTsfm, family = binomial(link = "logit"), data = dfMDLTrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2887  -0.0274   0.0000   0.0031   3.9991
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -74.29765   16.37766  -4.537 5.72e-06 ***
## indus           0.25253    0.11986   2.107 0.035131 *
## nox            90.17328   16.80734   5.365 8.09e-08 ***
## rm             -2.91454    1.34347  -2.169 0.030052 *
## age             0.09782    0.02862   3.417 0.000632 ***
## dis            -6.71722    1.94839  -3.448 0.000566 ***
## rad             1.30549    0.32622   4.002 6.29e-05 ***
## ptratio         0.98381    0.26914   3.655 0.000257 ***
## lstat           0.51957    0.30375   1.711 0.087167 .
## medv            0.49898    0.14898   3.349 0.000810 ***
## hghTax_drv1   -11.37170   29.11596  -0.391 0.696118
## hghIndus_drv1  -6.38527    2.98820  -2.137 0.032612 *
## dis_logTsfm    32.64050    7.75660   4.208 2.58e-05 ***
## lstat_logTsfm  -7.20110    3.91159  -1.841 0.065627 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 451.625  on 325  degrees of freedom
## Residual deviance:  78.883  on 312  degrees of freedom
## AIC: 106.88
##
## Number of Fisher Scoring iterations: 13

##  McFadden
## 0.8253354
```

4.4 Model 4: polynomials and interaction

The vif showed that there was some potential for multicollinearity between predictors and the pairs plot shows that some of the relationships between variables may be more complex than a simple linear relationship. Additionally the plot below of the predicted probability of model 3 against all numeric predictors shows that `rad` and `tax` have a clear break in values which could suggest the absence of a predictor or relationship.



Exploring inclusion of polynomials and interaction terms produced a model marginally more performant than model 3, but with a much more complicated interpretation.

```
##
## Call:
## glm(formula = target ~ zn + nox + rm + age + rad + tax + ptratio +
##     hghRad_drv + dis_logTsfm + poly(medv, 2) + tax:poly(medv,
##     2) + nox:rad + rad:dis_logTsfm, family = binomial(link = "logit"),
##     data = dfMDLTrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9405  -0.0059   0.0000   0.0000   3.6328
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -1.962e+02  5.608e+01  -3.499 0.000467 ***
## zn                 -2.999e-01  1.376e-01  -2.180 0.029277 *
## nox                 2.942e+02  8.781e+01   3.351 0.000807 ***
## rm                 -3.875e+00  1.287e+00  -3.010 0.002614 **
## age                 1.065e-01  2.761e-02   3.857 0.000115 ***
## rad                 3.067e+01  1.122e+01   2.734 0.006260 **
## tax                 6.042e-03  1.115e-02   0.542 0.587892
## ptratio             5.678e-01  2.274e-01   2.496 0.012550 *
## hghRad_drv1         1.526e+02  1.598e+03   0.096 0.923907
## dis_logTsfm         3.344e+01  9.435e+00   3.545 0.000393 ***
## poly(medv, 2)1     -8.765e+01  8.793e+01  -0.997 0.318869
```

```
## poly(medv, 2)2     -3.600e+01  5.872e+01  -0.613 0.539908
## tax:poly(medv, 2)1  7.002e-01  3.367e-01   2.079 0.037584 *
## tax:poly(medv, 2)2  2.785e-01  2.096e-01   1.329 0.183821
## nox:rad            -4.354e+01  1.684e+01  -2.585 0.009730 **
## rad:dis_logTsfm    -5.527e+00  1.919e+00  -2.881 0.003966 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 451.625  on 325  degrees of freedom
## Residual deviance:  74.811  on 310  degrees of freedom
## AIC: 106.81
##
## Number of Fisher Scoring iterations: 23
```

```
##  McFadden
## 0.8343523
```

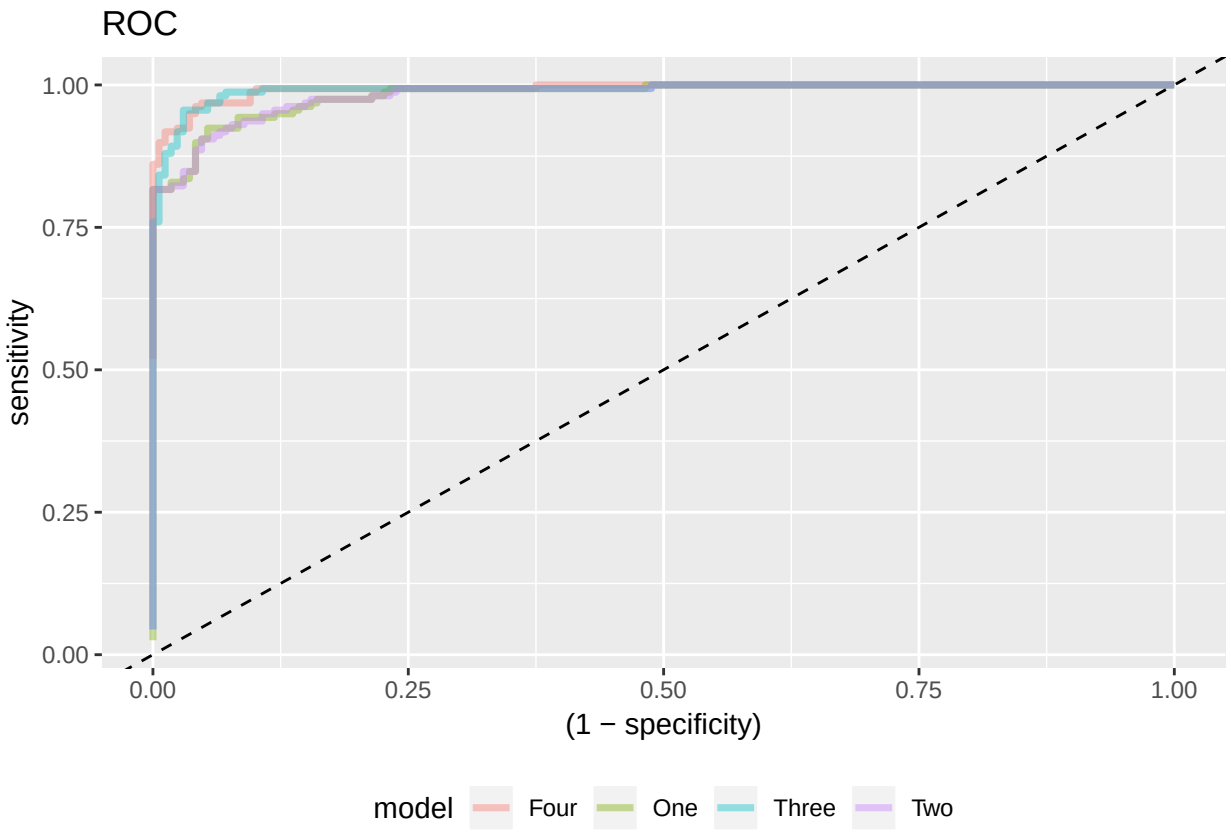
5 Select model

Reviewing the model statistics the AIC and McFadden R^2 suggest that model four is the most performant, but has a more complex model.

The project description does not describe the ultimate purpose of this analysis, in lieu of an explicit objective, the model decision will be based on accuracy of the predictions on the reserved training data.

5.1 ROC

The plot below shows the ROC curve for all models. These models and underlying data will be employed to select the optimal threshold for classification.



5.2 Model statistics

The table below shows the accuracy of the predictions on the reserved training data. Despite having a slighly higher McFadden R^2 model four is no better at predicting whether a neighborhood is more likely to have a higher than median crime rate. Given the same accuracy the more easily interpreted model 3 will be the selected model.

model	Accuracy
mdl1	0.9
mdl2	0.9
mdl3	0.9071
mdl4	0.9071

5.3 Prediction

Predictions of `target` using the evaluation data and selected model are included in the provided comma separated values (CSV) file.

6 Appendix

6.1 Session info

```
## Session info -------------------------------------------------------------
##  setting  value
##  version  R version 3.5.3 (2019-03-11)
##  system   x86_64, linux-gnu
##  ui       X11
##  language en_US
##  collate  en_US.UTF-8
##  tz       America/New_York
##  date     2019-04-22
## Packages -----------------------------------------------------------------
##  package      * version    date       source
##  abind          1.4-5      2016-07-21 CRAN (R 3.5.1)
##  assertthat     0.2.1      2019-03-21 CRAN (R 3.5.3)
##  backports      1.1.3      2018-12-14 cran (@1.1.3)
##  base         * 3.5.3      2019-03-11 local
##  boot           1.3-20     2017-07-30 CRAN (R 3.5.0)
##  broom          0.5.1      2018-12-05 cran (@0.5.1)
##  car          * 3.0-0      2018-04-02 CRAN (R 3.5.1)
##  carData      * 3.0-1      2018-03-28 CRAN (R 3.5.1)
##  caret        * 6.0-80     2018-05-26 CRAN (R 3.5.1)
##  cellranger     1.1.0      2016-07-27 CRAN (R 3.5.1)
##  class          7.3-15     2019-01-01 CRAN (R 3.5.2)
##  codetools      0.2-16     2018-12-24 CRAN (R 3.5.2)
##  colorspace     1.4-1      2019-03-18 CRAN (R 3.5.3)
##  compiler       3.5.3      2019-03-11 local
##  crayon         1.3.4      2017-09-16 CRAN (R 3.5.0)
##  curl           3.2        2018-03-28 CRAN (R 3.5.1)
##  CVST           0.2-2      2018-05-26 CRAN (R 3.5.1)
##  data.table     1.11.4     2018-05-27 CRAN (R 3.5.1)
##  datasets     * 3.5.3      2019-03-11 local
##  ddalpha        1.3.4      2018-06-23 CRAN (R 3.5.1)
##  DEoptimR       1.0-8      2016-11-19 CRAN (R 3.5.1)
##  DescTools    * 0.99.28    2019-03-17 CRAN (R 3.5.3)
##  devtools       1.13.6     2018-06-27 CRAN (R 3.5.1)
##  digest         0.6.18     2018-10-10 CRAN (R 3.5.1)
##  dimRed         0.1.0      2017-05-04 CRAN (R 3.5.1)
##  dplyr        * 0.8.0.1    2019-02-15 cran (@0.8.0.1)
##  DRR            0.0.3      2018-01-06 CRAN (R 3.5.1)
##  e1071        * 1.7-0      2018-07-28 CRAN (R 3.5.1)
##  evaluate       0.13       2019-02-12 CRAN (R 3.5.2)
##  expm           0.999-4    2019-03-21 CRAN (R 3.5.3)
##  forcats        0.3.0      2018-02-19 CRAN (R 3.5.1)
##  foreach        1.4.4      2017-12-12 CRAN (R 3.5.1)
##  foreign        0.8-71     2018-07-20 CRAN (R 3.5.1)
##  generics       0.0.2      2018-11-29 cran (@0.0.2)
##  geometry       0.3-6      2015-09-09 CRAN (R 3.5.1)
##  GGally       * 1.4.0      2018-05-17 CRAN (R 3.5.1)
##  ggcorrplot   * 0.1.2      2018-09-11 CRAN (R 3.5.1)
```

```
##  ggmosaic     * 0.2.0      2018-09-12 CRAN (R 3.5.3)
##  ggplot2      * 3.1.1      2019-04-07 CRAN (R 3.5.3)
##  glue           1.3.1      2019-03-12 CRAN (R 3.5.3)
##  gower          0.1.2      2017-02-23 CRAN (R 3.5.1)
##  graphics     * 3.5.3      2019-03-11 local
##  grDevices    * 3.5.3      2019-03-11 local
##  grid           3.5.3      2019-03-11 local
##  gtable         0.3.0      2019-03-25 CRAN (R 3.5.3)
##  haven          2.1.0      2019-02-19 CRAN (R 3.5.3)
##  hms            0.4.2      2018-03-10 CRAN (R 3.5.1)
##  htmltools      0.3.6      2017-04-28 CRAN (R 3.5.0)
##  htmlwidgets    1.3        2018-09-30 CRAN (R 3.5.1)
##  httr           1.3.1      2017-08-20 CRAN (R 3.4.1)
##  ipred          0.9-7      2018-08-14 CRAN (R 3.5.1)
##  iterators      1.0.10     2018-07-13 CRAN (R 3.5.1)
##  jsonlite       1.6        2018-12-07 CRAN (R 3.5.1)
##  kernlab        0.9-27     2018-08-10 CRAN (R 3.5.1)
##  knitr          1.22       2019-03-08 CRAN (R 3.5.2)
##  labeling       0.3        2014-08-23 CRAN (R 3.5.0)
##  lattice      * 0.20-38    2018-11-04 CRAN (R 3.5.1)
##  lava           1.6.3      2018-08-10 CRAN (R 3.5.1)
##  lazyeval       0.2.2      2019-03-15 CRAN (R 3.5.3)
##  lubridate      1.7.4      2018-04-11 CRAN (R 3.5.1)
##  magic          1.5-9      2018-09-17 CRAN (R 3.5.1)
##  magrittr       1.5        2014-11-22 CRAN (R 3.5.0)
##  manipulate     1.0.1      2014-12-24 CRAN (R 3.5.3)
##  MASS         * 7.3-51.3   2019-03-31 CRAN (R 3.5.3)
##  Matrix         1.2-17     2019-03-22 CRAN (R 3.5.3)
##  memoise        1.1.0      2017-04-21 CRAN (R 3.4.1)
##  methods      * 3.5.3      2019-03-11 local
##  ModelMetrics   1.2.0      2018-08-10 CRAN (R 3.5.1)
##  munsell        0.5.0      2018-06-12 CRAN (R 3.5.0)
##  mvtnorm        1.0-8      2018-05-31 CRAN (R 3.5.1)
##  nlme           3.1-139    2019-04-09 CRAN (R 3.5.3)
##  nnet           7.3-12     2016-02-02 CRAN (R 3.5.0)
##  openxlsx       4.1.0      2018-05-26 CRAN (R 3.5.1)
##  pander       * 0.6.2      2018-07-08 CRAN (R 3.5.1)
##  pillar         1.3.1      2018-12-15 CRAN (R 3.5.2)
##  pkgconfig      2.0.2      2018-08-16 cran (@2.0.2)
##  plotly         4.8.0      2018-07-20 CRAN (R 3.5.1)
##  pls            2.7-0      2018-08-21 CRAN (R 3.5.1)
##  plyr           1.8.4      2016-06-08 CRAN (R 3.5.0)
##  pROC         * 1.13.0     2018-09-24 CRAN (R 3.5.2)
##  prodlim        2018.04.18 2018-04-18 CRAN (R 3.5.1)
##  productplots   0.1.1      2016-07-02 CRAN (R 3.5.3)
##  pscl         * 1.5.2      2017-10-10 CRAN (R 3.5.1)
##  purrr        * 0.3.0      2019-01-27 cran (@0.3.0)
##  R6             2.4.0      2019-02-14 CRAN (R 3.5.2)
##  RColorBrewer   1.1-2      2014-12-07 CRAN (R 3.5.0)
##  Rcpp           1.0.0      2018-11-07 cran (@1.0.0)
##  RcppRoll       0.3.0      2018-06-05 CRAN (R 3.5.1)
##  readxl         1.1.0      2018-04-20 CRAN (R 3.5.1)
##  recipes        0.1.3      2018-06-16 CRAN (R 3.5.1)
##  reshape        0.8.8      2018-10-23 CRAN (R 3.5.1)
```

```
##  reshape2       1.4.3      2017-12-11 CRAN (R 3.5.0)
##  rio            0.5.10     2018-03-29 CRAN (R 3.5.1)
##  rlang          0.3.4      2019-04-07 CRAN (R 3.5.3)
##  rmarkdown      1.11       2018-12-08 CRAN (R 3.5.2)
##  robustbase     0.93-2     2018-07-27 CRAN (R 3.5.1)
##  rpart          4.1-15     2019-04-12 CRAN (R 3.5.3)
##  scales       * 1.0.0      2018-08-09 CRAN (R 3.5.1)
##  sfsmisc        1.1-2      2018-03-05 CRAN (R 3.5.1)
##  splines        3.5.3      2019-03-11 local
##  stats        * 3.5.3      2019-03-11 local
##  stats4         3.5.3      2019-03-11 local
##  stringi        1.4.3      2019-03-12 CRAN (R 3.5.3)
##  stringr        1.4.0      2019-02-10 CRAN (R 3.5.2)
##  survival       2.44-1.1   2019-04-01 CRAN (R 3.5.3)
##  tibble       * 2.0.1      2019-01-12 cran (@2.0.1)
##  tidyr        * 0.8.2      2018-10-28 cran (@0.8.2)
##  tidyselect     0.2.5      2018-10-11 cran (@0.2.5)
##  timeDate       3043.102   2018-02-21 CRAN (R 3.5.1)
##  tools          3.5.3      2019-03-11 local
##  utils        * 3.5.3      2019-03-11 local
##  viridisLite    0.3.0      2018-02-01 CRAN (R 3.5.0)
##  withr          2.1.2      2018-03-15 CRAN (R 3.5.0)
##  xfun           0.3        2018-07-06 CRAN (R 3.5.1)
##  yaml           2.2.0      2018-07-25 CRAN (R 3.5.1)
##  zip            1.0.0      2017-04-25 CRAN (R 3.5.1)
```

6.2 R source code

See included Rmarkdown (rmd) document