HW 3

Team 2

April 10, 2019

Contents

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

1. `zn`: proportion of residential land zoned for large lots (over 25000 square feet) (predictor variable)
2. `indus`: proportion of non-retail business acres per suburb (predictor variable)
3. `chas`: a dummy var. for whether the suburb borders the Charles River (1) or not (0) (predictor variable)
4. `nox`: nitrogen oxides concentration (parts per 10 million) (predictor variable)
5. `rm`: average number of rooms per dwelling (predictor variable)
6. `age`: proportion of owner-occupied units built prior to 1940 (predictor variable)
7. `dis`: weighted mean of distances to five Boston employment centers (predictor variable)
8. `rad`: index of accessibility to radial highways (predictor variable)
9. `tax`: full-value property-tax rate per $10,000 (predictor variable)
10. `ptratio`: pupil-teacher ratio by town (predictor variable)
11. `black`: 1000(Bk - 0.63)2 where Bk is the proportion of blacks by town (predictor variable)

12. `lstat`: lower status of the population (percent) (predictor variable)
13. `medv`: median value of owner-occupied homes in $1000s (predictor variable)
14. `target`: whether the crime rate is above the median crime rate (1) or not (0) (response variable)

Objective

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).

Dependencies

Replication of our work requires the following packages in Rstudio:

```
#install.packages('corrplot')

require(ggplot2)
require(corrplot)
require(dplyr)
require(tidyr)
require(randomForest)
require(forecast)
require(olsrr)
require(boot)
```

Data Exploration

First, we read the data as a csv then performed some simple statistical calculations so that we could explore the data. Below we can see a sample of the data output as it was read from the csv.

zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv	target
0	19.58	0	0.605	7.929	96.2	2.0459	5	403	14.7	3.70	50.0	1
0	19.58	1	0.871	5.403	100.0	1.3216	5	403	14.7	26.82	13.4	1
0	18.10	0	0.740	6.485	100.0	1.9784	24	666	20.2	18.85	15.4	1
30	4.93	0	0.428	6.393	7.8	7.0355	6	300	16.6	5.19	23.7	0
0	2.46	0	0.488	7.155	92.2	2.7006	3	193	17.8	4.82	37.9	0

We can explore how many `NAs` are in each column to see if we need to impute any of the variables:

zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv	target
466	466	466	466	466	466	466	466	466	466	466	466	466

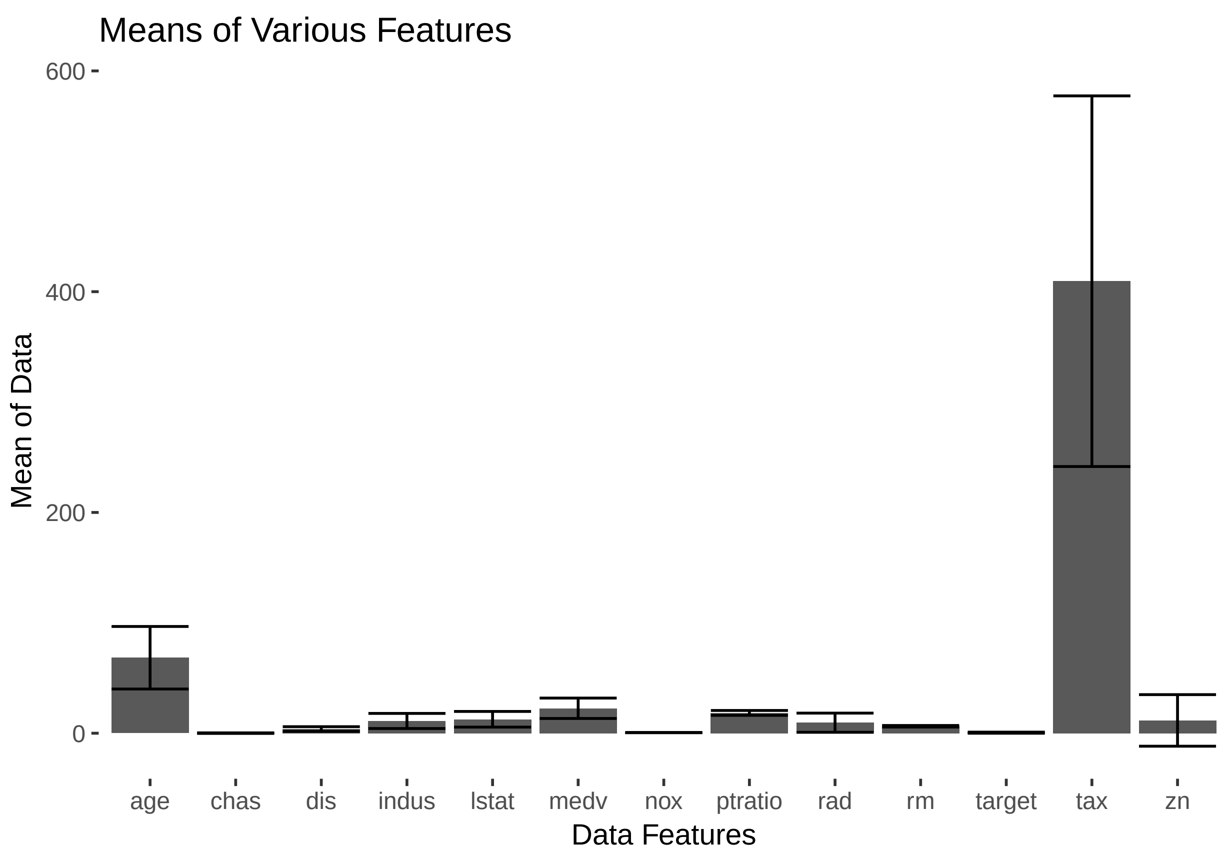
As we can see, each data vector has the same number of entries, 466. Thus, imputation will not be necessary.

Summary Statistics

We then calculated the mean and standard deviation for each data vector:

	means	sds
zn	11.5772532	23.3646511
indus	11.1050215	6.8458549
chas	0.0708155	0.2567920
nox	0.5543105	0.1166667
rm	6.2906738	0.7048513
age	68.3675966	28.3213784
dis	3.7956929	2.1069496
rad	9.5300429	8.6859272
tax	409.5021459	167.9000887
ptratio	18.3984979	2.1968447
lstat	12.6314592	7.1018907
medv	22.5892704	9.2396814
target	0.4914163	0.5004636

Below is a bar chart that illutrates the average and standard deviation for each of our data vectors. As we can see, the `tax` vector is a totally different magnitude than the rest. Models involving this vector will benefit from normalization or scaling.

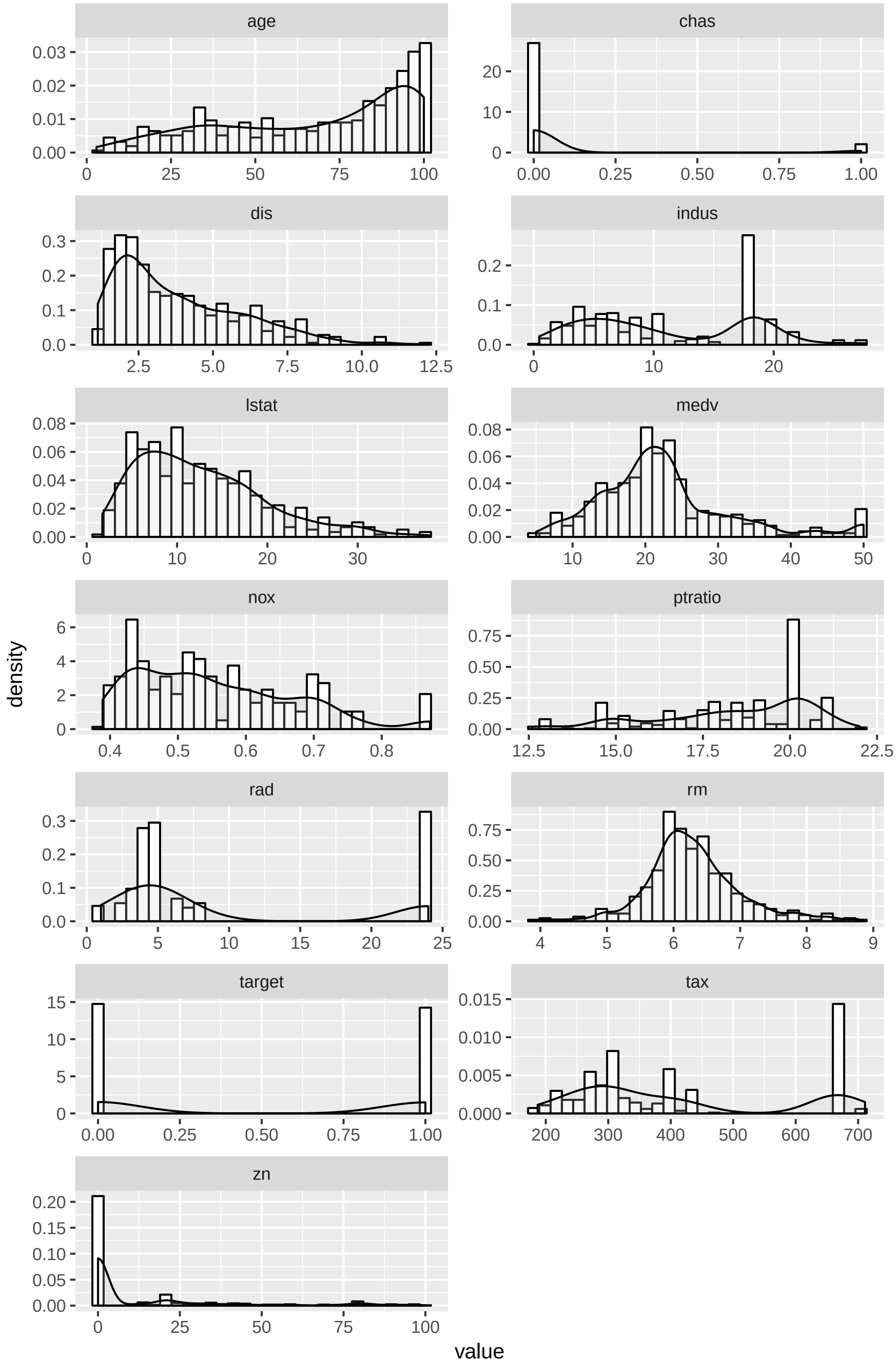


Histogram

The following histograms help visualize the spread and skewness of the raw data.

```
ggplot(data = gather(training), mapping = aes(x = value)) +
  geom_histogram(aes(y=..density..), colour="black", fill="white")+
```

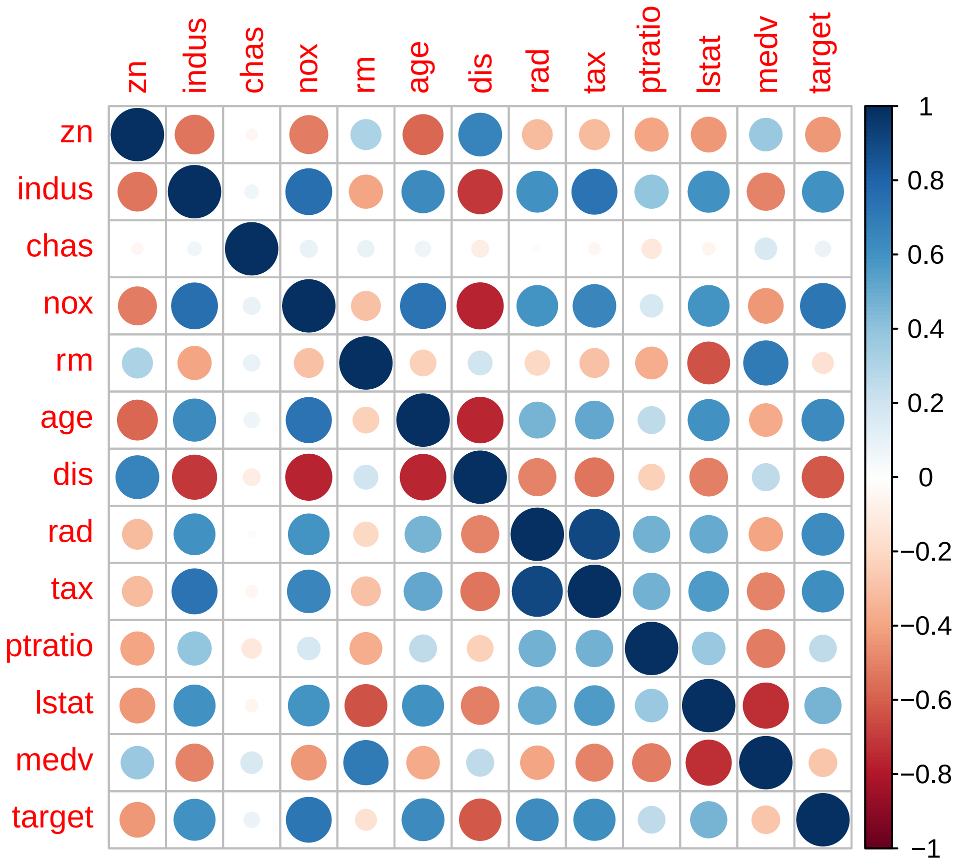
```
  geom_density(alpha=.2, fill="lightgrey")+
  facet_wrap(~key, ncol = 2, scales = 'free')
```



age
chas
dis
indus
lstat
medv
nox
ptratio
rad
rm
target
tax
zn
density
value

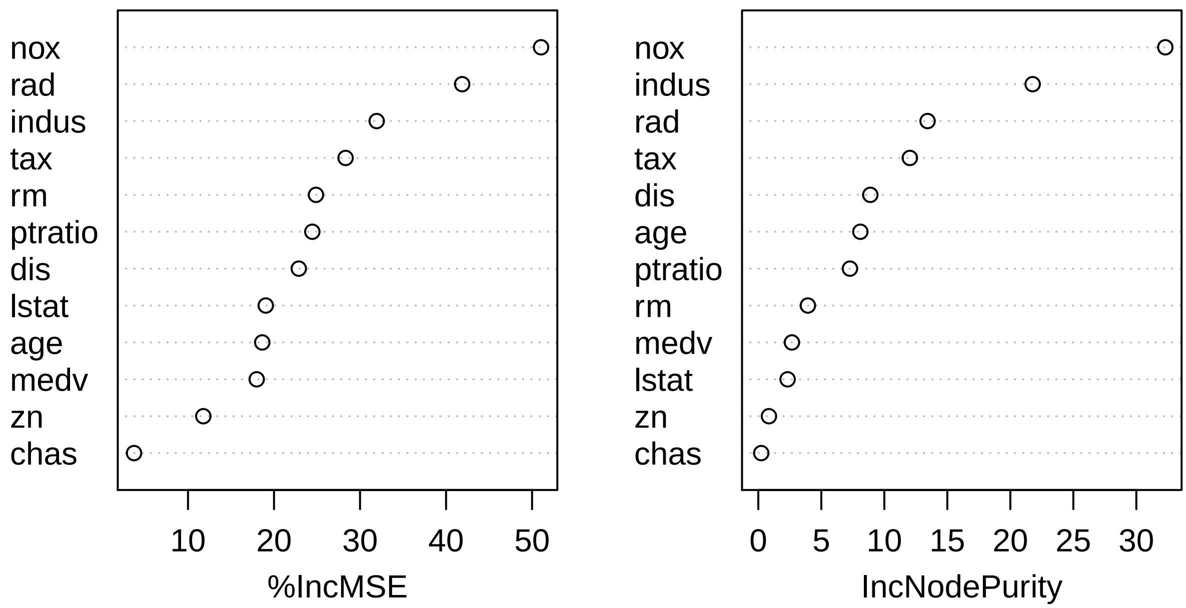
Correlation

We can see our correlation matrix below. A dark blue circle represents a strong positive relationship and a dark red circle represents a strong negative relationship between two variables. We can see that `indus`, `nox`, `target`, and `dis` have the most colinearity. Likewise, these vectors are the best predictors for the target value. Note that this plot only includes rows tha have data in each column.



Finally, we can use the `randomforest` package to verify our assumptions from the correlation plot.

fit



We verified our assumptions above using 1000 random forests. The `nox`, `rad`, `indus`, and `tax` have the most effect. While `dis`is strongly colinear, it has less effect on the target. This is likely due to it encoding information stored redundantly in another vector.

Data Preparation

In the following section, we will prepare and transform our variables for our model:

Transformations for Multicollinearity

We saw some correlation between our predictor variables in our exploritory correlation plots. We can test this correlation using variance inflation factors (VIF) to ensure our model is not affected by multicollinearity.

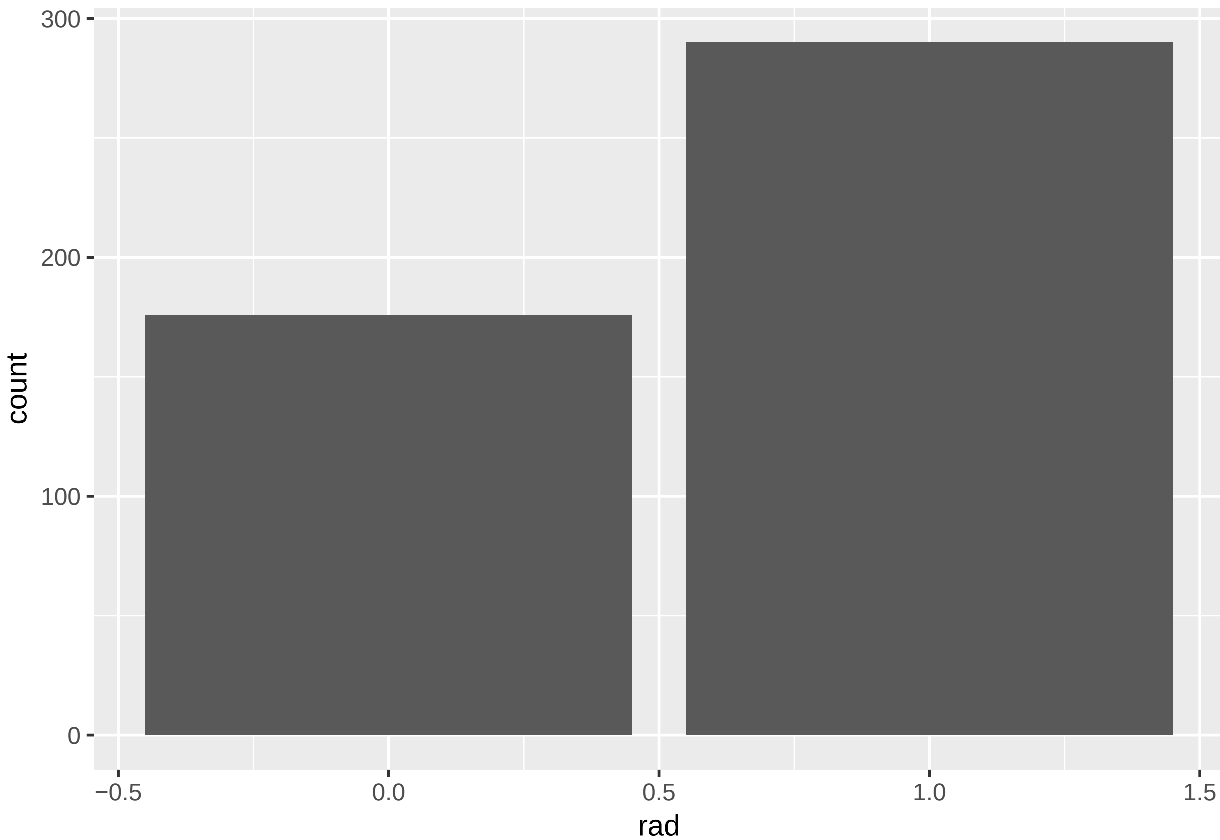
Variables	Tolerance	VIF	Standard_Error
rad	0.1474632	6.781354	2.604103
tax	0.1084925	9.217228	3.035989

This test shows us that the `rad` and `tax` variables have high multicollinearity above 5. Both variables should not be used together, without transformation in our model. The above table shows that as the standard error for both exceeds 2 times the amount then if these variables were not related.

Rad is an index variable that represents accessibility to radial highways. We choose to bifucate this data using the median value, 5.

Variables	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv
Tolerance	0.4137525	0.2483370	0.9221830	0.2210510	0.4298120	0.3213630	0.2354190	0.5836380	0.2531420	0.4765870	0.2799080	0.2775283
VIF	2.416904	4.026775	1.084383	4.523839	2.326598	3.111737	4.247738	1.713388	3.950339	2.098250	3.572597	3.603236
Standard_Errors	1.554639	2.006683	1.041337	2.126932	1.525319	1.764012	2.061004	1.308965	1.987546	1.448534	1.890131	1.898219

```
ggplot(training2, aes(x=rad))+geom_bar()
```



Through this change, the `tax` and `rad` variables are no longer affected by multicollinearity.

Log Transformations

While logistic modeling does not require normalized data, we choose to apply log transformations to the `age`, `lstat`, and `tax` variables.

```
training2 <- training2 %>%
  mutate_at(.vars = vars(age, lstat, tax), .funs = log)
```

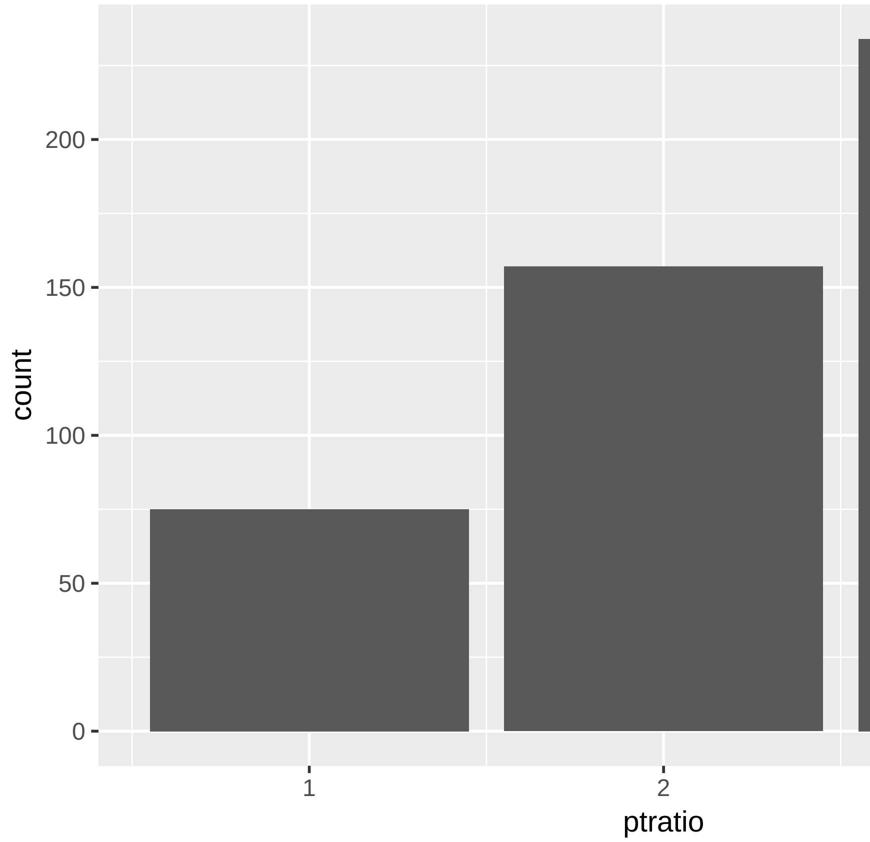
This transformation helps center the `age` and normalize the `lstat` and `tax` variables.

New Variables

We choose to create several variables from our initial dataset.

ptratio

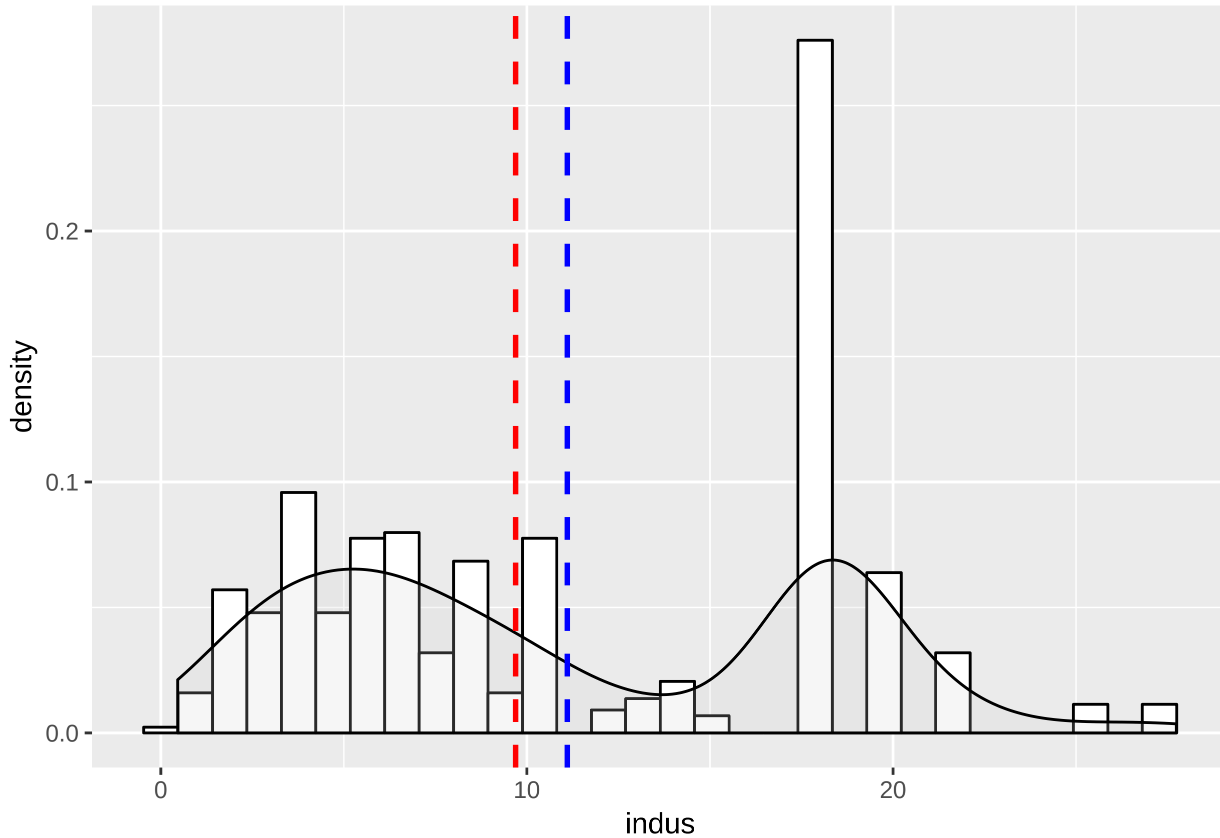
We first changed `ptratio`, a pupil-teacher ratio measurement, into a categorial variable. In the new variable, 0 represents small, 1 represents medium, and 3 represents large ratios.



Our new variable for `ptratio` now looks like this:

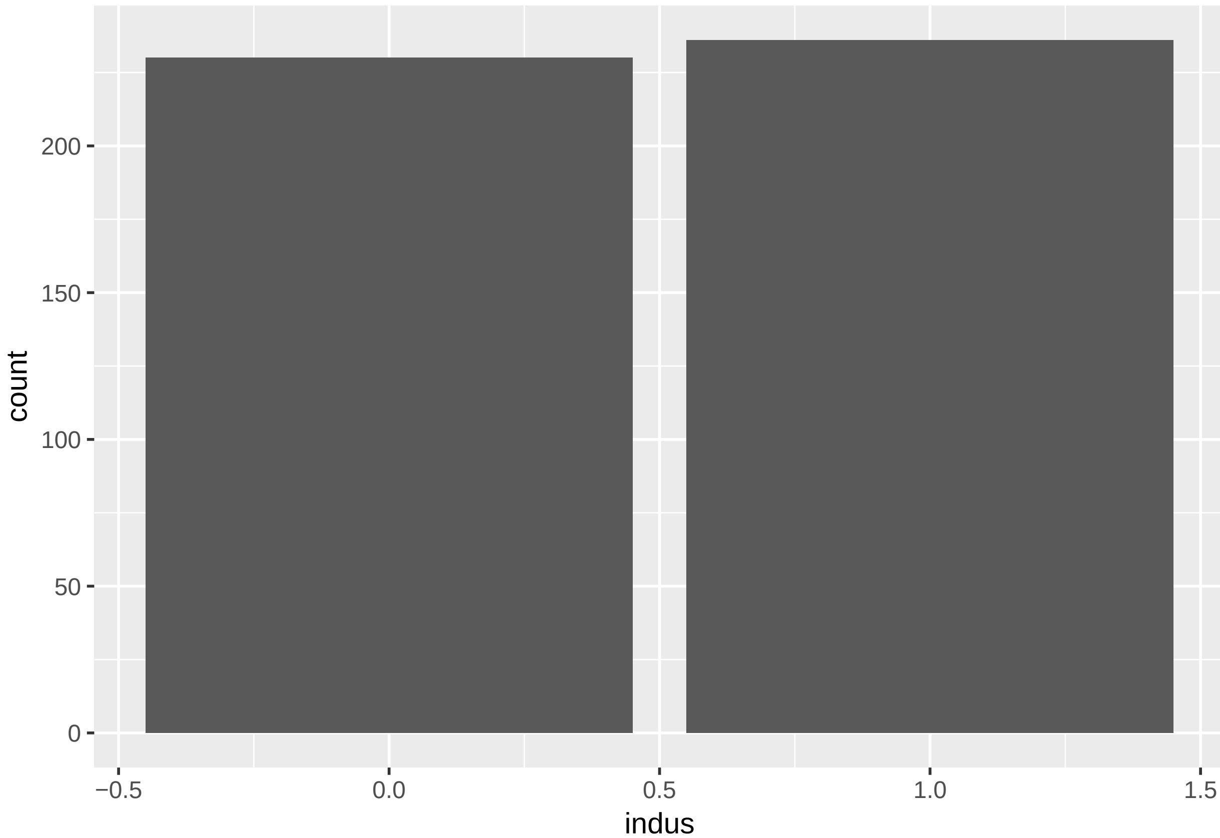
indus

This variable represents the proportion of non-retail business acres per suburb. The plots below show the `indus` data is bimodal, skewed right, and centered around 10. The red line shows the median, whereas the blue line depicts the mean value for this variable.




```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.460   5.145   9.690  11.105  18.100  27.740
```

We choose to bifuncate this variable using its median value.




```
##   zn indus chas   nox    rm      age    dis rad      tax ptratio    lstat
## 1  0     1    0 0.605 7.929 4.566429 2.0459   1 5.998937       1 1.308333
## 2  0     1    1 0.871 5.403 4.605170 1.3216   1 5.998937       1 3.289148
## 3  0     1    0 0.740 6.485 4.605170 1.9784   1 6.501290       3 2.936513
## 4 30     0    0 0.428 6.393 2.054124 7.0355   1 5.703782       2 1.646734
## 5  0     0    0 0.488 7.155 4.523960 2.7006   0 5.262690       2 1.572774
## 6  0     0    0 0.520 6.781 4.266896 2.8561   1 5.950643       3 2.037317
##   medv target
## 1 50.0      1
## 2 13.4      1
## 3 15.4      1
## 4 23.7      0
## 5 37.9      0
## 6 26.5      0
```

Build Models

- [ ] 3 binary logistic models
- [ ] forward, stepwise, random forest, etc
- [ ] Inferences
- [ ] Coefficients

Select Models

- [ ] Use Log Likelihood, AIC, ROC curve,
- [ ] Evaluate Training Set
- [ ] Accuracy, Error, Precision, Sensitivity, Specificity, F1 score, AUC, conf matrix (hint: use assignment 2, and check outthis link)
- [ ] Make predictions with test set and interpret