HW-3-YQ

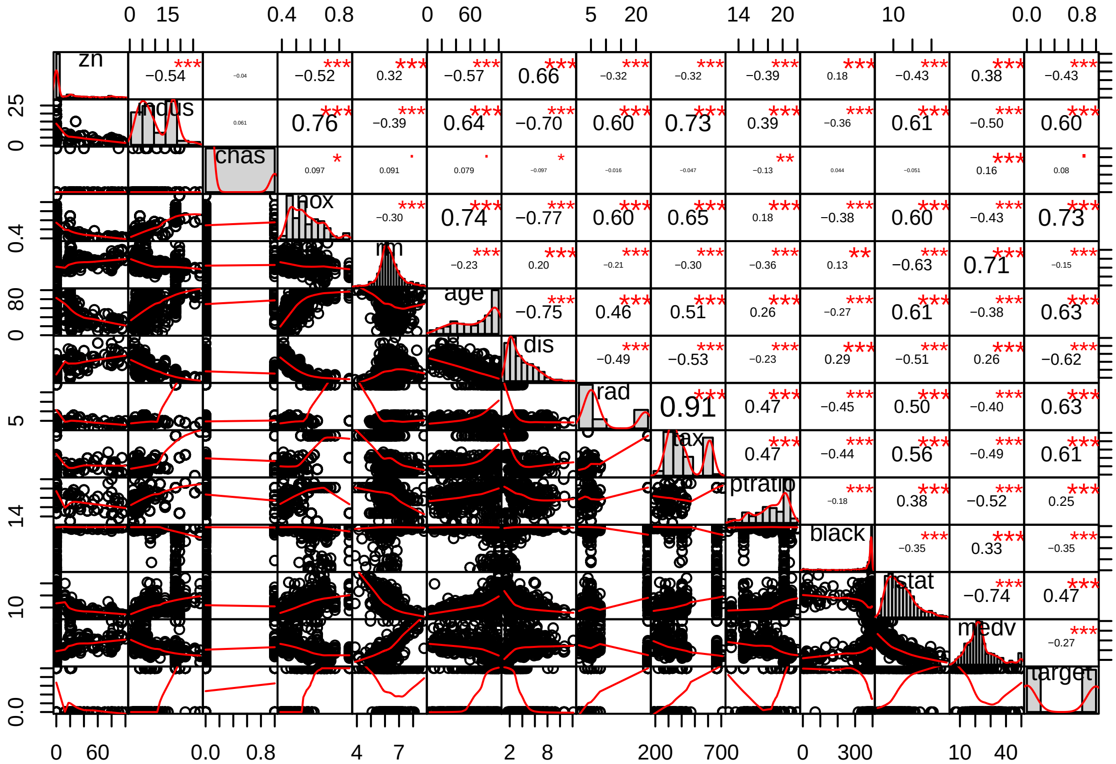
Youqing Xiang

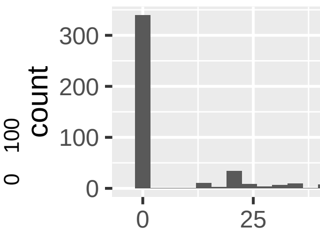
June 25, 2016

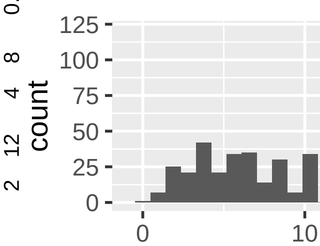
DATA EXPLORATION

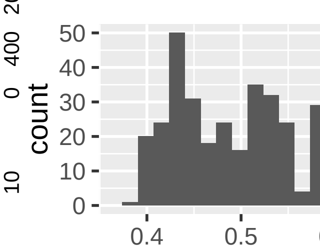
```
##        zn             indus             chas              nox
##  Min.   :  0.00   Min.   : 0.460   Min.   :0.00000   Min.   :0.3890
##  1st Qu.:  0.00   1st Qu.: 5.145   1st Qu.:0.00000   1st Qu.:0.4480
##  Median :  0.00   Median : 9.690   Median :0.00000   Median :0.5380
##  Mean   : 11.58   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         black            lstat
##  Min.   :187.0   Min.   :12.6   Min.   :  0.32   Min.   : 1.730
##  1st Qu.:281.0   1st Qu.:16.9   1st Qu.:375.61   1st Qu.: 7.043
##  Median :334.5   Median :18.9   Median :391.34   Median :11.350
##  Mean   :409.5   Mean   :18.4   Mean   :357.12   Mean   :12.631
##  3rd Qu.:666.0   3rd Qu.:20.2   3rd Qu.:396.24   3rd Qu.:16.930
##  Max.   :711.0   Max.   :22.0   Max.   :396.90   Max.   :37.970
##       medv           target
##  Min.   : 5.00   Min.   :0.0000
##  1st Qu.:17.02   1st Qu.:0.0000
##  Median :21.20   Median :0.0000
##  Mean   :22.59   Mean   :0.4914
##  3rd Qu.:25.00   3rd Qu.:1.0000
##  Max.   :50.00   Max.   :1.0000
```

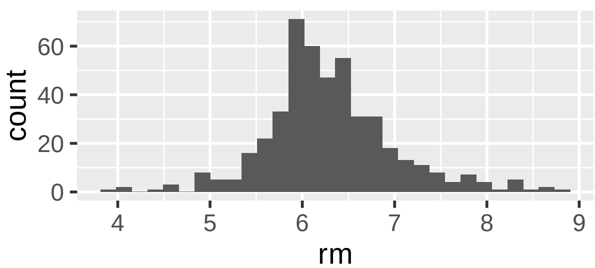
```
## [1] 466  14
```

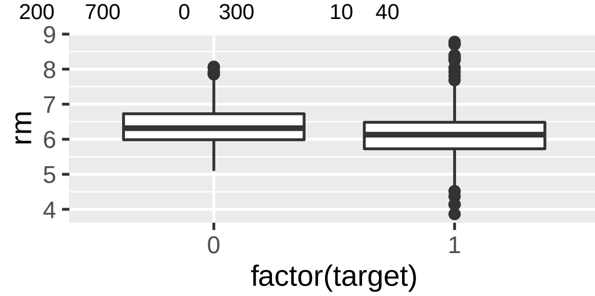

zn
indus
chas
nox
rm
age
dis
rad
tax
ptratio
black
lstat
medv
target

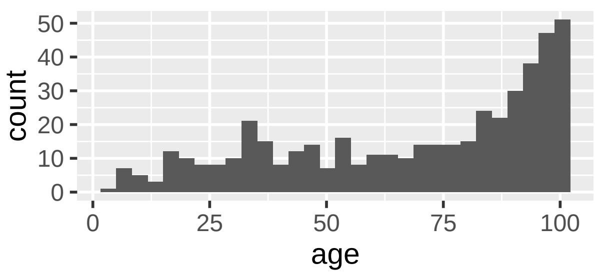

count

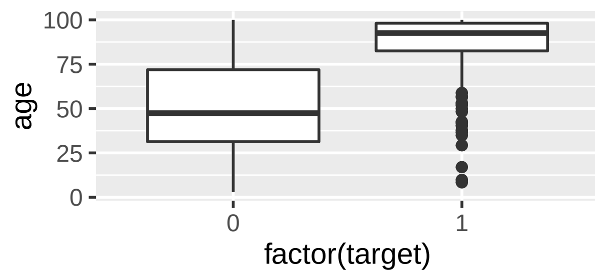

count

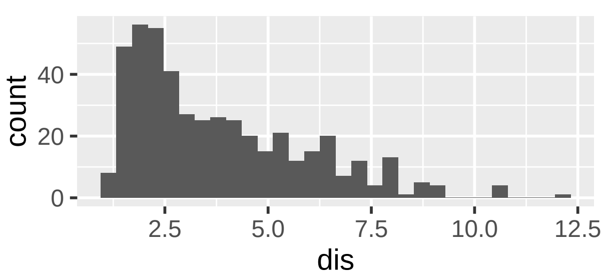

count

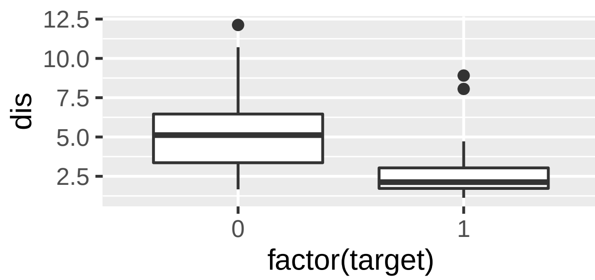

count
rm

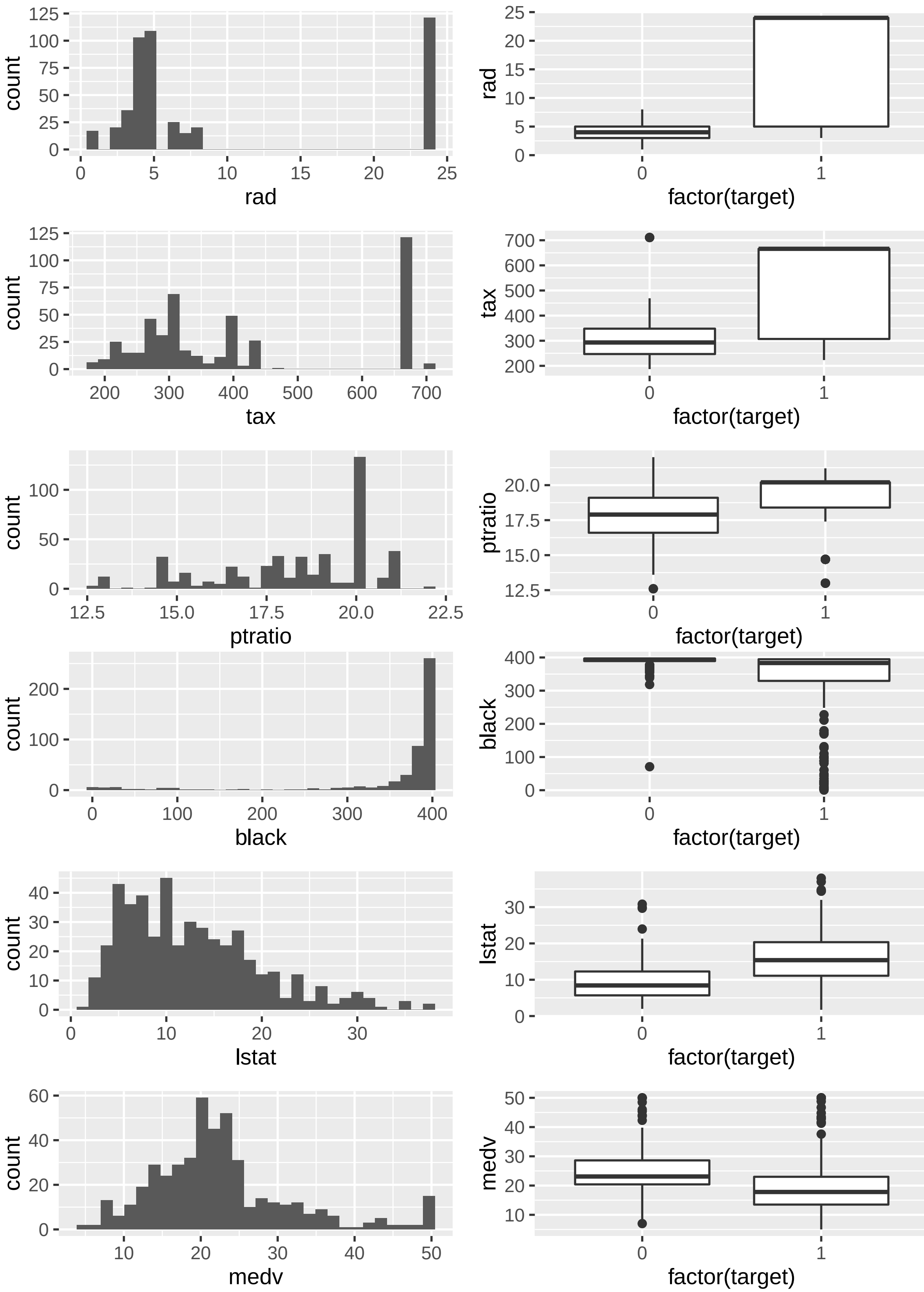

rm
factor(target)


count
age


age
factor(target)


count
dis


dis
factor(target)


count
rad
0 5 10 15 20 25
0 25 50 75 100 125
factor(target)
0 1
0 5 10 15 20 25
count
tax
200 300 400 500 600 700
0 25 50 75 100 125
factor(target)
0 1
200 300 400 500 600 700
count
ptratio
12.5 15.0 17.5 20.0 22.5
0 50 100
factor(target)
0 1
12.5 15.0 17.5 20.0
count
black
0 100 200 300 400
0 100 200
factor(target)
0 1
0 100 200 300 400
count
lstat
0 10 20 30
0 10 20 30 40
factor(target)
0 1
0 10 20 30
count
medv
10 20 30 40 50
0 20 40 60
factor(target)
0 1
10 20 30 40 50

Chas	Target	Freq
0	0	225
1	0	12
0	1	208
1	1	21

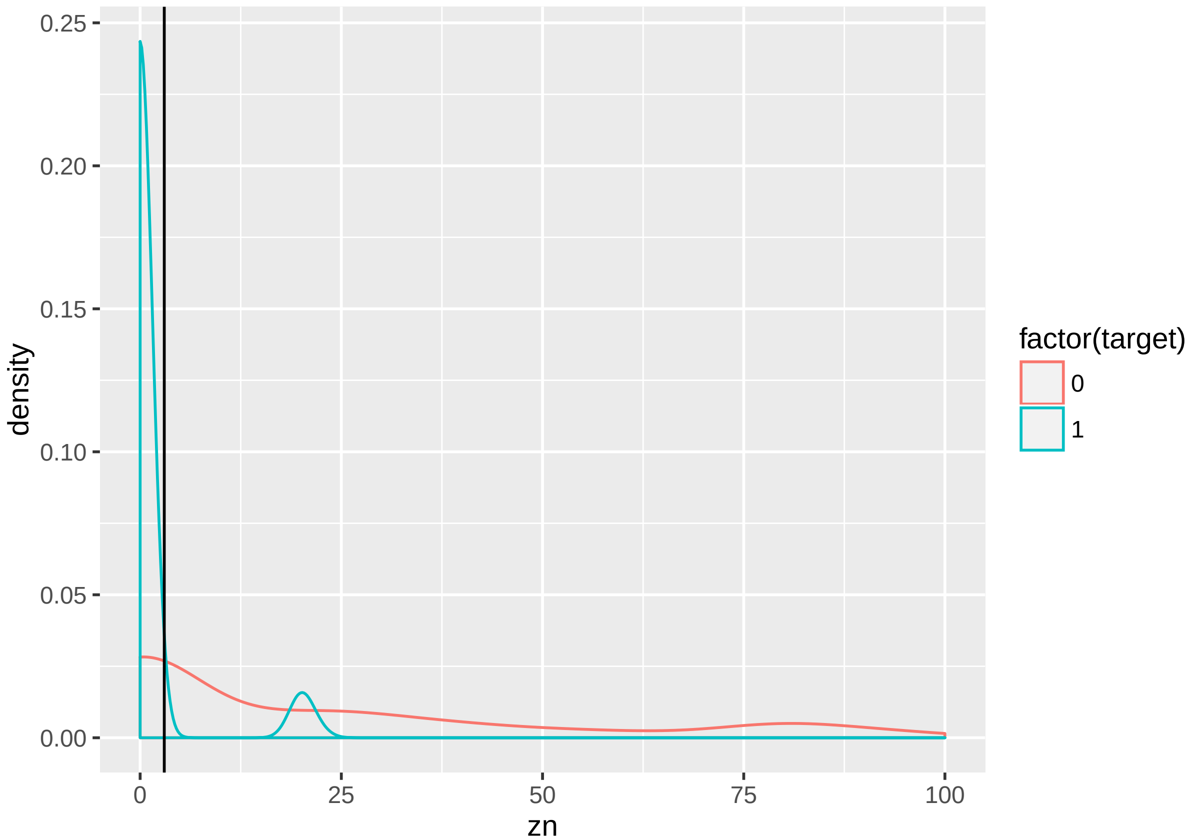
- For this data set, we have 14 columns (13 variables and 1 predictor) and total 466 observations. Among 13 variables, only `chas` is category data and others are numeric data. And we don't see any missing data.
- From Scatterplot Matrix, we can see there are strong correlationship among variables, such as `indus` verse `nox` and `lstat` verse `medv`. So, **multicollinearity** is one issue that we have to pay close attention to and PCA analysis should be considered during modeling.
- From histogram plots of variables and boxplots of variable grouped by predictor, we can see that there are some outliers we might want to deal with. Meanwhile, we could consider to do some data transformation, such as transforming `zn` from numberic to categorcial. In addition, we can tell that some variables could be very important to predict `target`, such as `zn`, `indus`, `dis` and `rad`; `chas` and `rm` might not be very useful.

DATA PREPARATION

For this data set, we don't see any missing data and obivious nonsense data. So, the section will focus on dealing with some outliers and data transformation.

1. zn

```
ggplot(crime, aes(x=zn)) + geom_density(aes(colour=factor(target))) + xlim(0,100) +
  geom_vline(xintercept = 3)
```



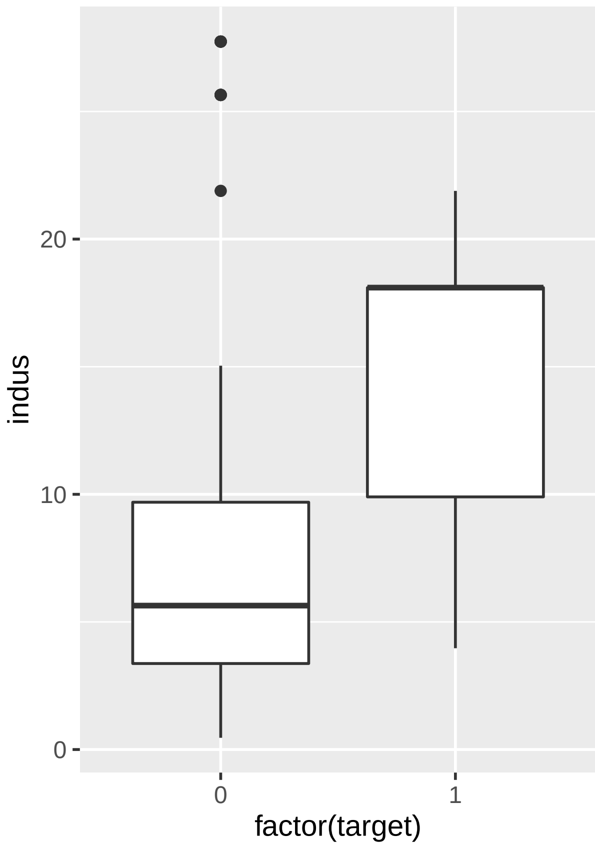

```
crime$znN <- ifelse(crime$zn > 3, 1, 0)
crime$znN <- as.factor(crime$znN)
t <- as.data.frame(table(znN=crime$znN, Target=crime$target))
kable(t)
```

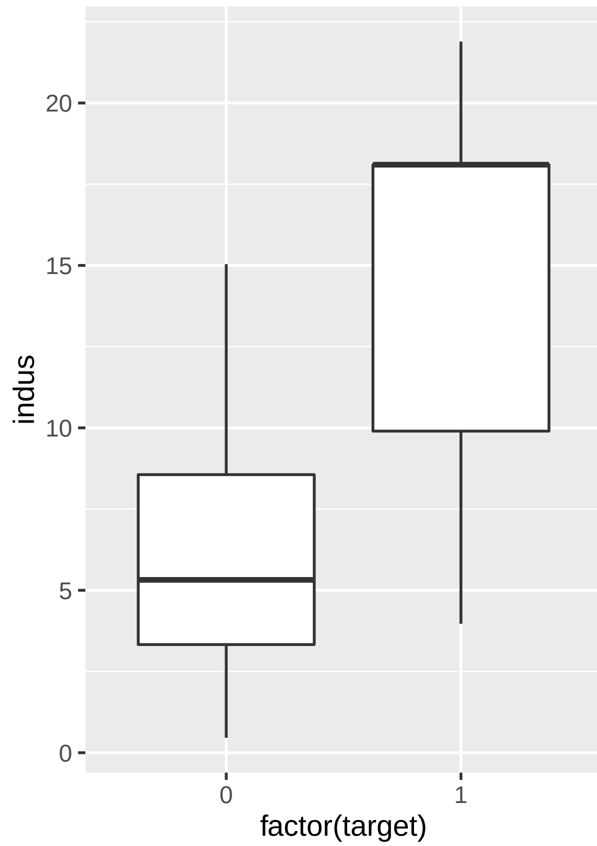
znN	Target	Freq
0	0	125
1	0	112
0	1	214
1	1	15

From the above density plot, we can see it is worth to try transforming numberic zn variable to a new categorical variable. Here I set up a new variable **znN**: *1* means more than 3% of residential land zoned for large lots (over 25000 square feet) and *0* means less than or equal to 3% of residential land zoned for large lots (over 25000 square feet).

2. indus

```
attach(crime)
p0 <- ggplot(crime, aes(factor(target), indus)) + geom_boxplot()
crime <- crime[-which(target==0 & indus > 20),]
p1 <- ggplot(crime, aes(factor(target), indus)) + geom_boxplot()
grid.arrange(p0, p1,ncol=2,nrow=1)
```






```
detach(crime)
```

Here I removed the rows which `indus` is greater than 20 while `target` is 0.

3. nox

Nothing is done with this variable.

4. rm

```
t <- as.data.frame(table(Rm=round(crime$rm), Target=crime$target))
kable(t)
```

Rm	Target	Freq
4	0	0
5	0	1
6	0	140
7	0	73
8	0	12
9	0	0
4	1	4
5	1	33
6	1	136

Rm	Target	Freq
7	1	42
8	1	11
9	1	3

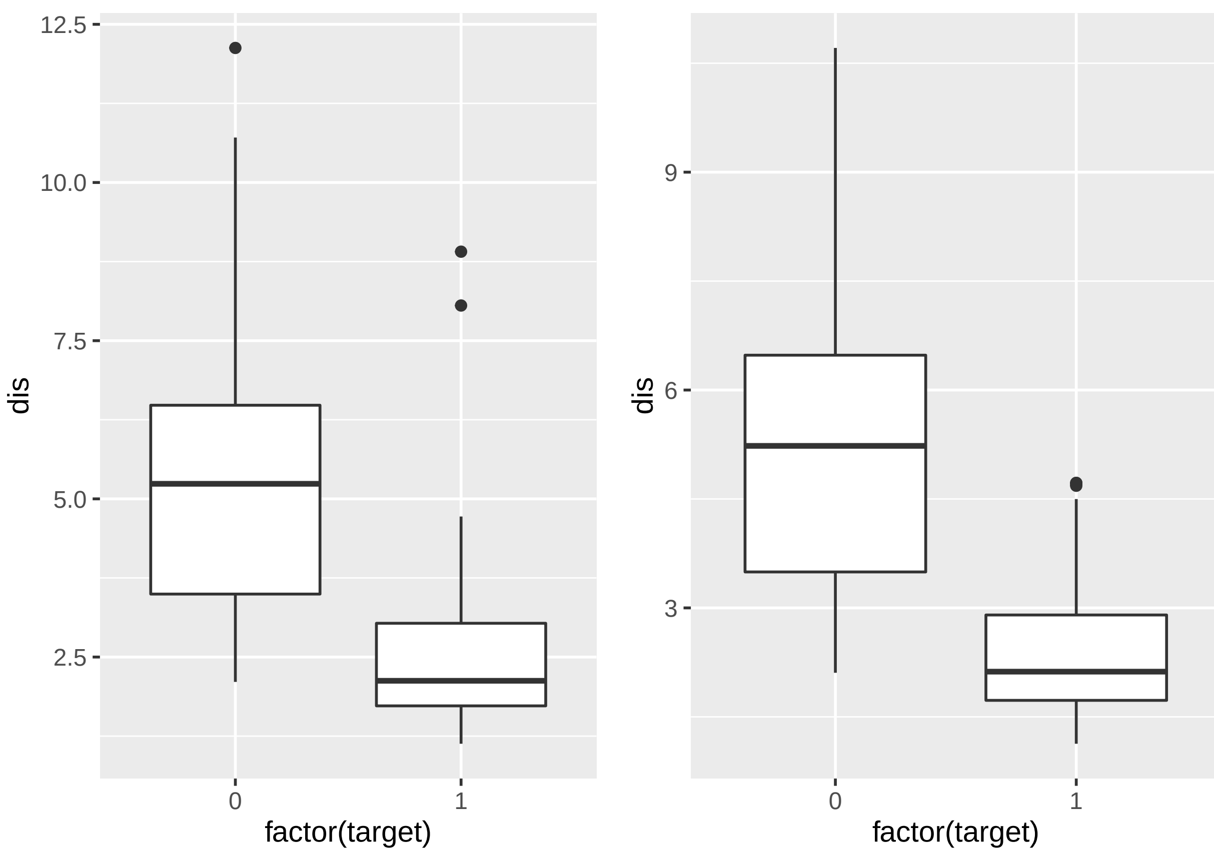
It looks like there is not a obvious relationship between rm and target. So nothing is done with this variable.

5. age

The maxium of `age` is 100 years and the data is strongly right skewed. Although it is possible that the buildings which are older than 100 years were recorded as 100 years, I do nothing due to lacking of detailed information about this variable.

6. dis

```
attach(crime)
p0 <- ggplot(crime, aes(factor(target), dis)) + geom_boxplot()
crime <- crime[-which(target==0 & dis > 11),]
crime <- crime[-which(target==1 & dis > 7.5),]
p1 <- ggplot(crime, aes(factor(target), dis)) + geom_boxplot()
grid.arrange(p0, p1, ncol=2,nrow=1)
```

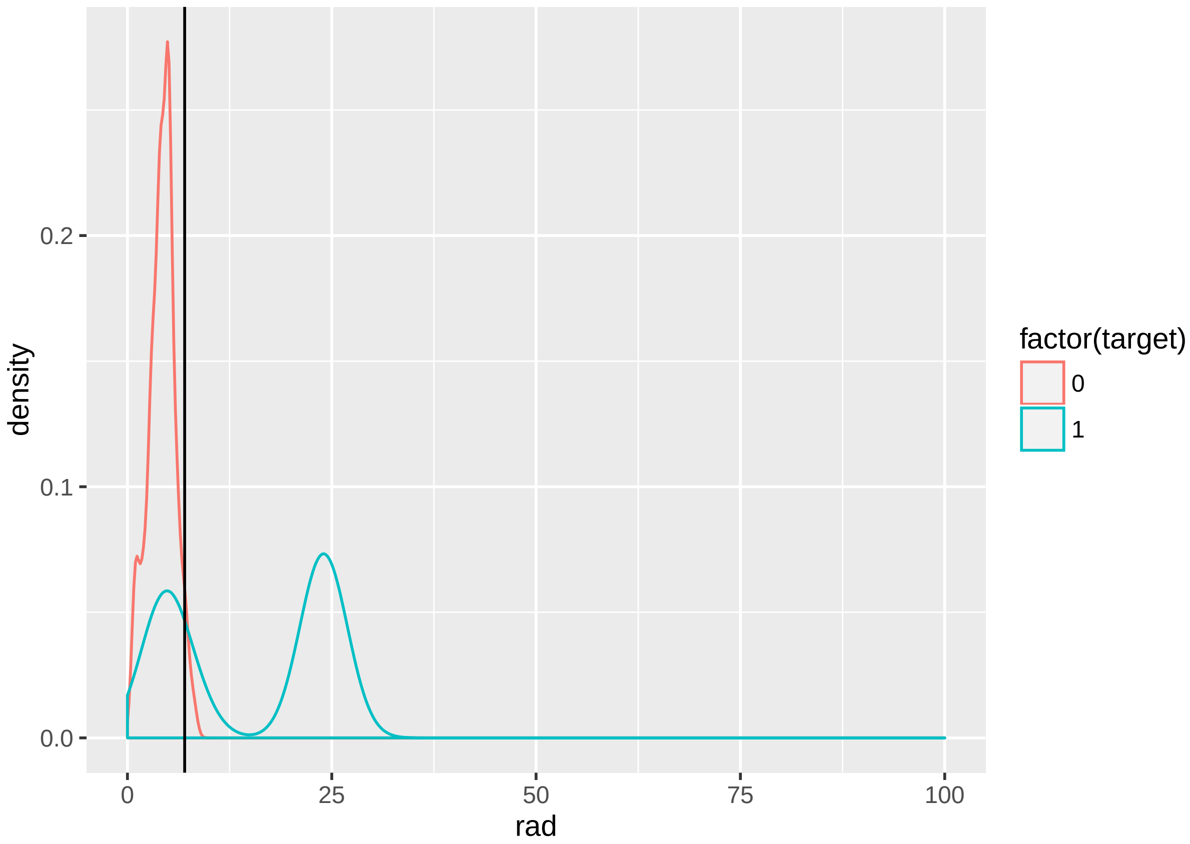


```r
detach(crime)
```

Here I removed the rows which `dis` is greater than 11 while `target` is 0 and the rows which `dis` is greater than 7.5 while `target` is 1.

7. rad

```r
ggplot(crime, aes(x=rad)) + geom_density(aes(colour=factor(target))) + xlim(0,100) + geom_vline(xinterc
```



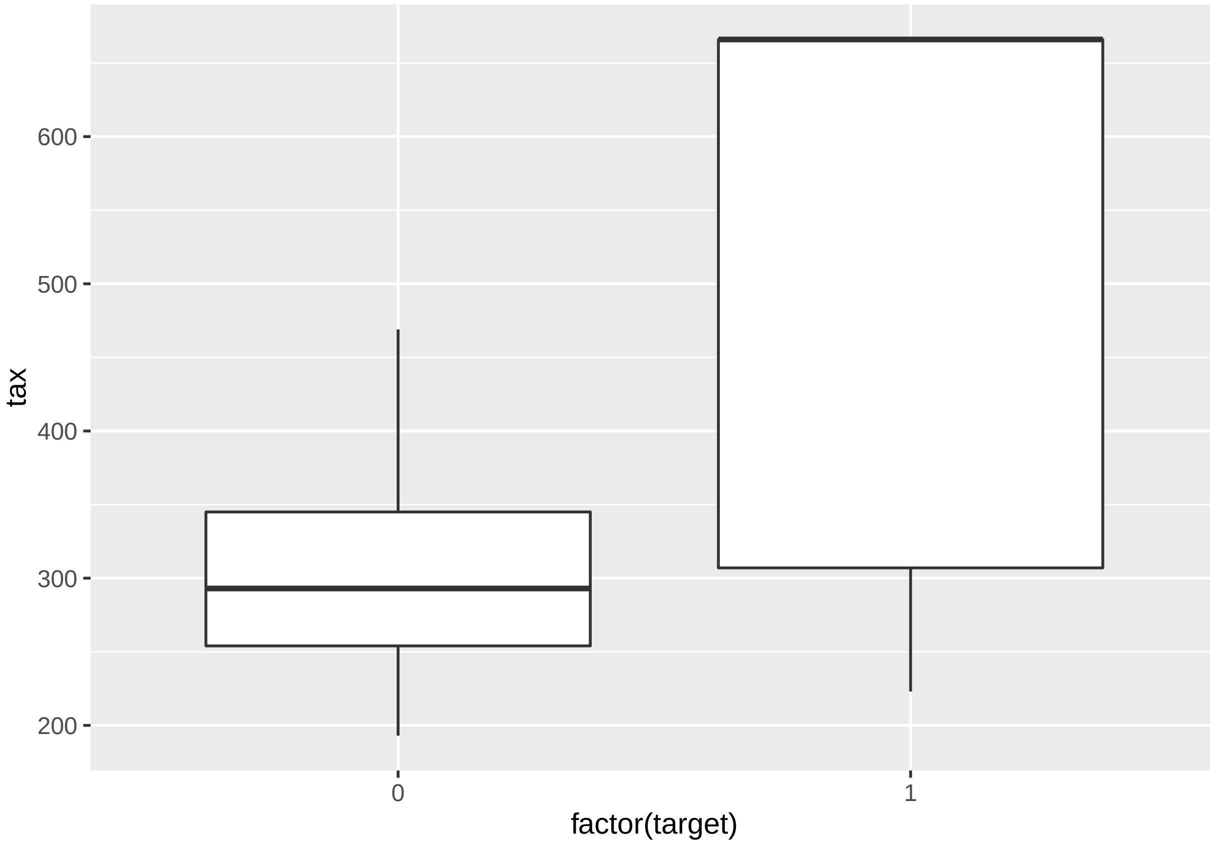

```r
crime$radN <- ifelse(crime$rad > 7, 1, 0)
crime$radN <- as.factor(crime$radN)
t <- as.data.frame(table(radN=crime$radN, Target=crime$target))
kable(t)
```

radN	Target	Freq
0	0	221
1	0	4
0	1	90
1	1	137

Here I applied the same strategy as `zn`. I set up a new variable `radN`: *1* means index of accessibility to radial highways is greater than 7 and *0* means index of accessibility to radial highways is less than or equal to 7.

8. tax

```
p0 <- ggplot(crime, aes(factor(target), tax)) + geom_boxplot()
p0
```



For `tax` variable, the outliner I saw in blox plot at data Exploration part was already removed. So, do nothing to this variable here.

9. ptratio, black, lstat, medv

For these variables, I also see outliners on boxplot. But if we try to remove outliners, we would lose more data points. So, nothing is done with them.

10. chas

```
crime$chas <- as.factor(crime$chas)
```

`chas` is a category variable, so here I changed the data type of `chas`.

11. Summary after data preparation

```r
names(crime)
```

```
##  [1] "zn"      "indus"   "chas"    "nox"     "rm"      "age"     "dis"
##  [8] "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"    "target"
## [15] "znN"     "radN"
```

```r
dim(crime)
```

```
## [1] 452  16
```

At the end, we removed 14 rows and added 2 new variables: `znN` and `radN`.

Modeling

Split the data into train and test data sets for model1 and model2

```r
set.seed(45)
inTrain <- createDataPartition(y=crime$target, p=0.7,list=FALSE)
training <- crime[inTrain,]
testing <- crime[-inTrain,]
```

I split the data into `training` for modeling and `testing` for evaluating models.

Model 1-using the original variances

```r
m11 <- glm(target ~ . -znN-radN, data=training, family = binomial(link='probit'))
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
#summary(m11)
m12 <- update(m11, .~. - zn-chas-rm-dis-black)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
m1 <- m12
summary(m1)
```

```
##
## Call:
## glm(formula = target ~ indus + nox + age + rad + tax + ptratio +
##     lstat + medv, family = binomial(link = "probit"), data = training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -2.343  -0.026   0.000   0.000   2.882
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -24.171357   4.325079  -5.589 2.29e-08 ***
## indus         0.134991   0.060490   2.232 0.025639 *
## nox          28.874385   5.510981   5.239 1.61e-07 ***
## age           0.015291   0.007279   2.101 0.035654 *
## rad           0.491912   0.136730   3.598 0.000321 ***
## tax          -0.017047   0.004355  -3.914 9.07e-05 ***
## ptratio       0.344176   0.103985   3.310 0.000933 ***
## lstat         0.099590   0.035125   2.835 0.004579 **
## medv          0.079078   0.029423   2.688 0.007196 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 438.75  on 316  degrees of freedom
## Residual deviance: 102.98  on 308  degrees of freedom
## AIC: 120.98
##
## Number of Fisher Scoring iterations: 11
```

Model2 - Using the three new created variances

```
m21 <- glm(target ~ .-age-zn-rad, data=training,
               family = binomial(link='probit'))
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
#summary(m21)
m22 <- update(m21, .~. - indus-chas-rm-dis-black-znN)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
#summary(m22)
m23 <- update(m22, .~. -medv)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
#summary(m23)
m24 <- update(m23, .~. -ptratio)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
m2 <- m24
summary(m2)
```

```
##
## Call:
## glm(formula = target ~ nox + tax + lstat + radN, family = binomial(link = "probit"),
##     data = training)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.32446  -0.04097   0.00000   0.02465   2.62981
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -16.708448   2.552283  -6.546 5.89e-11 ***
## nox          32.430637   5.246318   6.182 6.35e-10 ***
## tax          -0.005576   0.002087  -2.672  0.00754 **
## lstat         0.080376   0.027954   2.875  0.00404 **
## radN1         2.932473   0.556149   5.273 1.34e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 438.75  on 316  degrees of freedom
## Residual deviance: 107.01  on 312  degrees of freedom
## AIC: 117.01
##
## Number of Fisher Scoring iterations: 9
```

Model3 - PCA

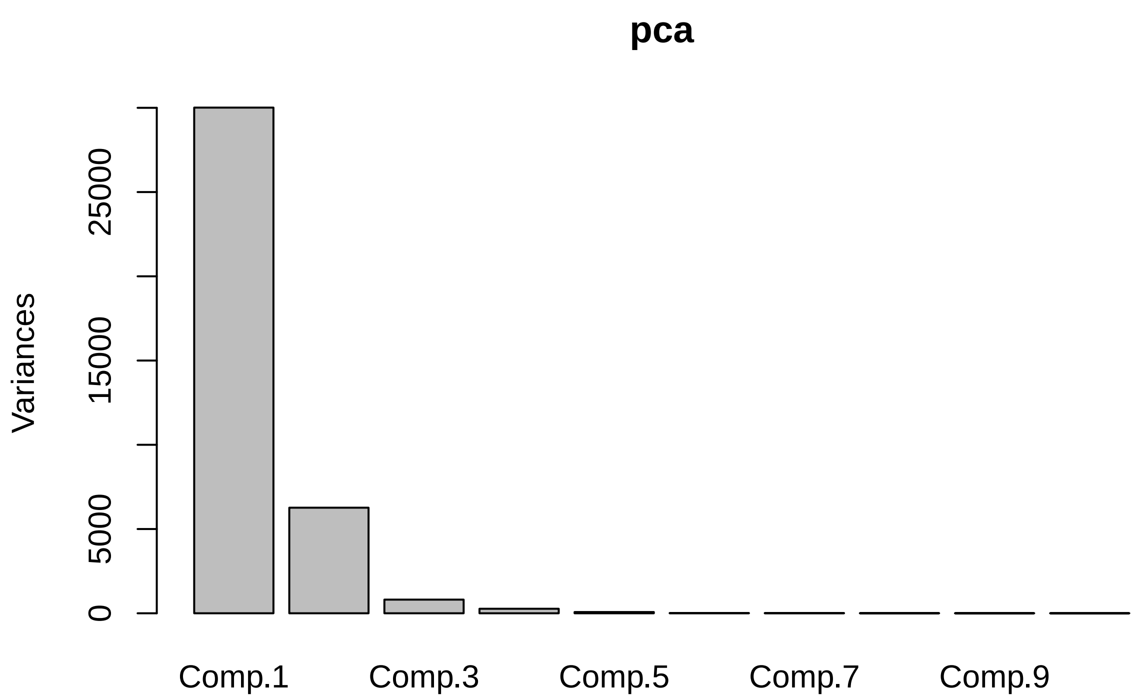
```
crime_pca <- crime[,1:14]
crime_pca <- select(crime_pca,-chas)
names(crime_pca)
```

```
##  [1] "zn"      "indus"   "nox"     "rm"      "age"     "dis"     "rad"
##  [8] "tax"     "ptratio" "black"   "lstat"   "medv"    "target"
```

```
target <- crime_pca$target
A <- as.matrix(select(crime_pca,-target))
pca <- princomp(A,center=T,scale.=T)
```

```
## Warning: In princomp.default(A, center = T, scale. = T) :
##  extra arguments 'center', 'scale.' will be disregarded
```

```
plot(pca)
```




```r
summary(pca)
```

```
## Importance of components:
##                            Comp.1     Comp.2      Comp.3       Comp.4
## Standard deviation     173.243309 79.1629681 28.47173155 16.356643297
## Proportion of Variance   0.801022  0.1672537  0.02163512  0.007140357
## Cumulative Proportion    0.801022  0.9682757  0.98991084  0.997051201
##                             Comp.5       Comp.6       Comp.7       Comp.8
## Standard deviation     8.522310494 3.7350582651 3.6494348424 2.5580438979
## Proportion of Variance 0.001938413 0.0003723285 0.0003554535 0.0001746415
## Cumulative Proportion  0.998989614 0.9993619425 0.9997173960 0.9998920375
##                              Comp.9      Comp.10      Comp.11      Comp.12
## Standard deviation     1.658030e+00 1.042133e+00 4.551563e-01 5.421923e-02
## Proportion of Variance 7.336962e-05 2.898531e-05 5.529078e-06 7.845818e-08
## Cumulative Proportion  9.999654e-01 9.999944e-01 9.999999e-01 1.000000e+00
```

```r
pca <- as.data.frame(pca$scores[,1:2])
crime_pca <- cbind(target=target,pca)
```

```r
head(crime_pca)
```

```
##   target      Comp.1     Comp.2
## 1      1    7.982215  -10.79214
## 2      1   14.663799  -36.79382
## 3      1 -237.185595 -109.13296
## 4      0  115.531924   17.45648
## 5      0  214.333148   31.11811
## 6      0   35.789981  -29.36113
```

```r
set.seed(45)
inTrain_pca <- createDataPartition(y=crime_pca$target, p=0.7,list=FALSE)
training_pca <- crime_pca[inTrain_pca,]
testing_pca <- crime_pca[-inTrain_pca,]
m3 <- glm(target ~ ., data=training_pca)
summary(m3)
```

```
##
## Call:
## glm(formula = target ~ ., data = training_pca)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.59191  -0.27404  -0.10935   0.04818   0.83770
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.5116620  0.0217455  23.530   <2e-16 ***
## Comp.1      -0.0018248  0.0001246 -14.639   <2e-16 ***
## Comp.2      -0.0001726  0.0002618  -0.659     0.51
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.1496637)
##
##     Null deviance: 79.073  on 316  degrees of freedom
## Residual deviance: 46.994  on 314  degrees of freedom
## AIC: 302.49
##
## Number of Fisher Scoring iterations: 2
```

Model Evaluation

1. Confusion Matrix

```r
# Model1
predict_1 <- predict(m12, newdata=testing, type='response')
glm.pred1 = ifelse(predict_1 > 0.5, 1, 0)
cM1 <- confusionMatrix(glm.pred1, testing$target, positive = "1")

# Model2
predict_2 <- predict(m2, newdata=testing, type='response')
glm.pred2 = ifelse(predict_2 > 0.5, 1, 0)
cM2 <- confusionMatrix(glm.pred2, testing$target, positive = "1")

# Model3
predict_3 <- predict(m3,newdata=testing_pca,type='response')
glm.pred3 = ifelse(predict_3 > 0.5, 1, 0)
cM3 <- confusionMatrix(glm.pred3, testing_pca$target, positive = "1")
```

```r
# Put results together
df1b <- as.data.frame(cM1$byClass)
df1a <- as.data.frame(cM1$overall)
colnames(df1a) <- 'Model1'
colnames(df1b) <- 'Model1'
df1 <- rbind(df1a, df1b)

df2b <- as.data.frame(cM2$byClass)
df2a <- as.data.frame(cM2$overall)
colnames(df2a) <- 'Model2'
colnames(df2b) <- 'Model2'
df2 <- rbind(df2a, df2b)

df3b <- as.data.frame(cM3$byClass)
df3a <- as.data.frame(cM3$overall)
colnames(df3a) <- 'Model3'
colnames(df3b) <- 'Model3'
df3 <- rbind(df3a, df3b)

df <- cbind(df1,df2,df3)
kable(df,caption='Confusion Matrix')
```

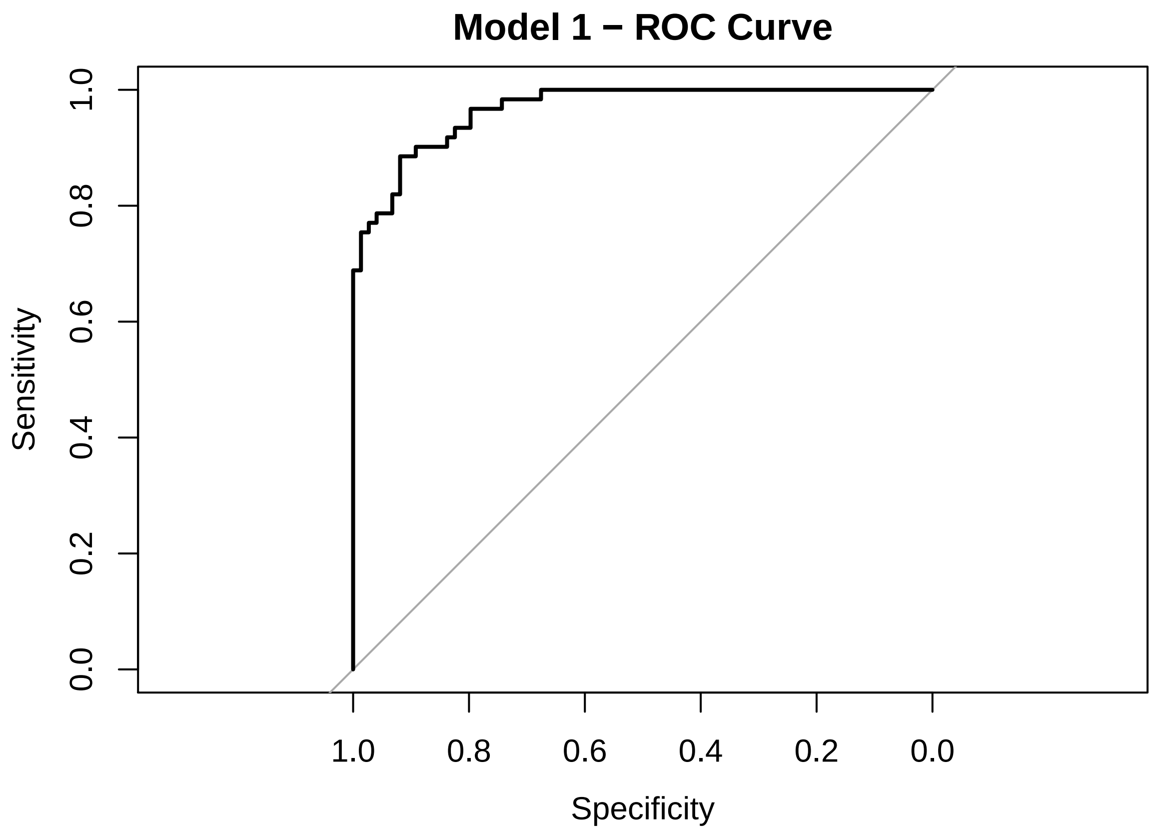
Table 5: Confusion Matrix

	Model1	Model2	Model3
Accuracy	0.8814815	0.9037037	0.8222222
Kappa	0.7621145	0.8075447	0.6315670
AccuracyLower	0.8146770	0.8409602	0.7471282
AccuracyUpper	0.9307163	0.9477237	0.8826473
AccuracyNull	0.5481481	0.5481481	0.5481481
AccuracyPValue	0.0000000	0.0000000	0.0000000
McnemarPValue	0.4532547	0.0960923	0.0005202
Sensitivity	0.9016393	0.9508197	0.6557377
Specificity	0.8648649	0.8648649	0.9594595
Pos Pred Value	0.8461538	0.8529412	0.9302326
Neg Pred Value	0.9142857	0.9552239	0.7717391
Prevalence	0.4518519	0.4518519	0.4518519
Detection Rate	0.4074074	0.4296296	0.2962963
Detection Prevalence	0.4814815	0.5037037	0.3185185
Balanced Accuracy	0.8832521	0.9078423	0.8075986

2. ROC Curve and Area under the Curve

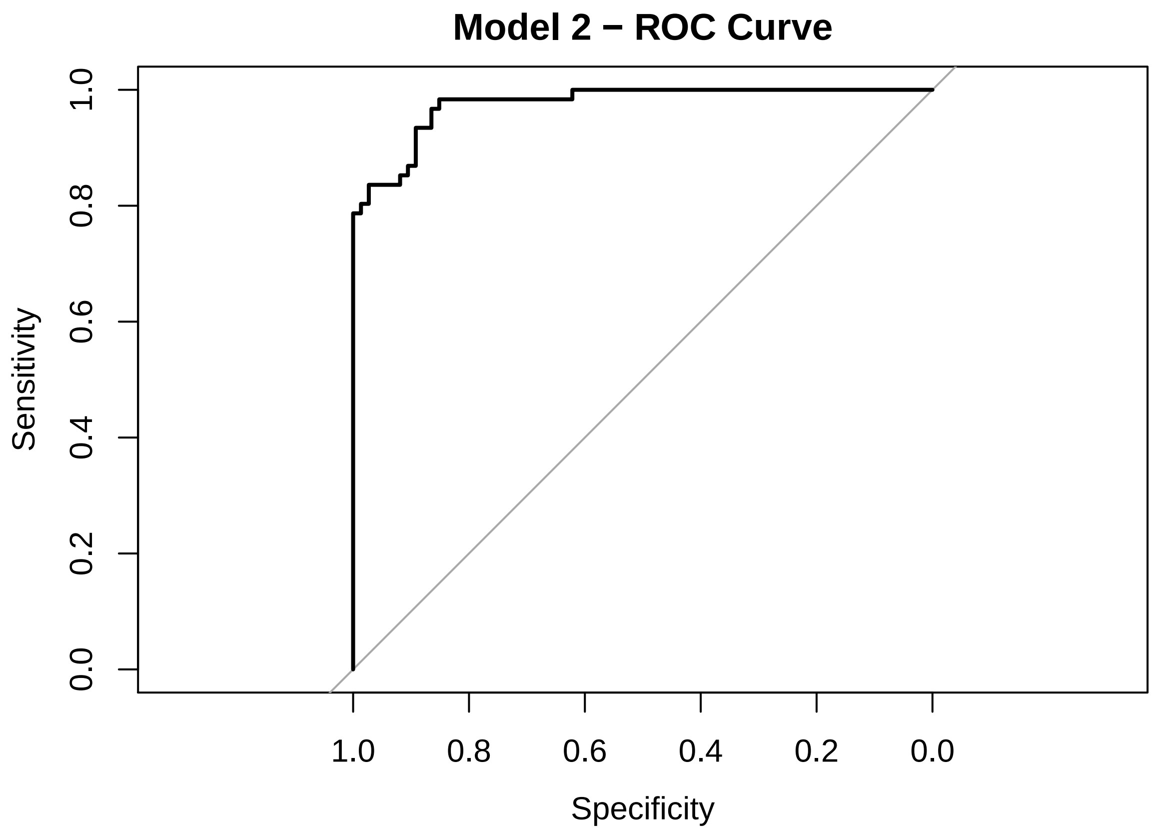
```r
rc1 <- roc(factor(target) ~ predict_1, data=testing)
rc2 <- roc(factor(target) ~ predict_2, data=testing)
rc3 <- roc(factor(target) ~ predict_3, data=testing_pca)

plot(rc1,main='Model 1 - ROC Curve')
```



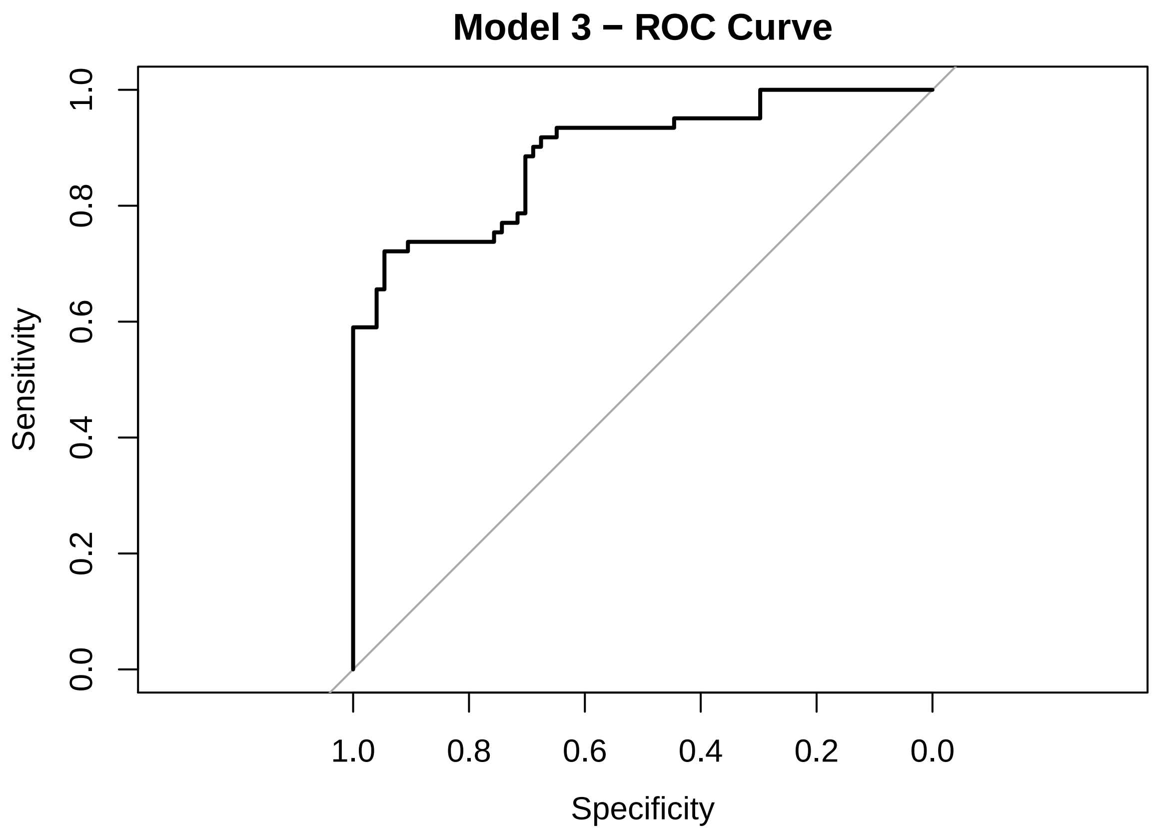

```
##
## Call:
## roc.formula(formula = factor(target) ~ predict_1, data = testing)
##
## Data: predict_1 in 74 controls (factor(target) 0) < 61 cases (factor(target) 1).
## Area under the curve: 0.967
```

```
plot(rc2,main='Model 2 - ROC Curve')
```




```
##
## Call:
## roc.formula(formula = factor(target) ~ predict_2, data = testing)
##
## Data: predict_2 in 74 controls (factor(target) 0) < 61 cases (factor(target) 1).
## Area under the curve: 0.9759
```

```
plot(rc3,main='Model 3 - ROC Curve')
```




```
##
## Call:
## roc.formula(formula = factor(target) ~ predict_3, data = testing_pca)
##
## Data: predict_3 in 74 controls (factor(target) 0) < 61 cases (factor(target) 1).
## Area under the curve: 0.8903
```

```
model <- c('Model 1', 'Model 2', 'Model 3')
area <- c(auc(rc1),auc(rc2),auc(rc3))
df <- data.frame(Model=model,AUC=area)
kable(df,caption='Area under the curve')
```

Table 6: Area under the curve

Model	AUC
Model 1	0.9669916
Model 2	0.9758529
Model 3	0.8903412

3.Log-likelihood/AIC/BIC

```
LL.1 <- logLik(m1)
LL.2 <- logLik(m2)
```

```r
LL.3 <- logLik(m3)
LL <- rbind(LL.1, LL.2, LL.3) %>% round(2)
```

Akaike's 'An Information Criterion'

```r
AIC.1 <- AIC(m1)
AIC.2 <- AIC(m2)
AIC.3 <- AIC(m3)
AIC <- rbind(AIC.1, AIC.2, AIC.3) %>% round(2)
```

Coefficient of Determination

```r
# http://stats.stackexchange.com/questions/577/is-there-any-reason-to-prefer-the-aic-or-bic-over-the-ot
BIC.1 <- BIC(m1)
BIC.2 <- BIC(m2)
BIC.3 <- BIC(m3)
BIC <- rbind(BIC.1, BIC.2, BIC.3) %>% round(2)
```

```r
eval.table <- cbind(LL, AIC, BIC)

rownames(eval.table) <- c("Model 1", "Model 2", "Model 3")
colnames(eval.table) <- c("Log Likelihood", "AIC", "BIC")

kable(eval.table,caption = 'Log-likelihood/AIC/BIC')
```

Table 7: Log-likelihood/AIC/BIC

	Log Likelihood	AIC	BIC
Model 1	-51.49	120.98	154.81
Model 2	-53.51	117.01	135.81
Model 3	-147.25	302.49	317.53