MATH 449 - Final Project

Dona Inayyah & Bryan Thorne

2023-05-21

GitHub Link:

State the problem and describe the data set

The data used for our final project, Arrests.csv, includes data on individuals in Toronto, Canada that were arrested for simple possession of small quantities of marijuana. The problem is to investigate the relationship between the release of an arrested individual, with a summons, and their race, sex, age, year of arrest, employment status, citizenship status, and number of checks (the number of times the arrested individual's name appeared in police databases).

This is an un-grouped dataset that consists of 5226 observations with 8 variables.

The variables are:

- "released" (1=Yes, 0=No)
- "colour" (1=White, 0=Black)
- "sex" (1=Male, 0=Female)
- "employed" (1=Yes, 0=No)
- "citizen" (1=Yes, 0=No)
- "year" (2002 - 1997)
- "age" (12 - 66)
- "checks" (0 - 6).

Variables "released", "color", "sex", "employed", and "citizens" are categorical whereas variables "year", "age" and "checks" are numerical. We will use "released" as the response variable, while the rest are explanatory variables.

Fit a logistic regression model with all predictors.

```
setwd('/Users/donainayyah/Desktop/SPRING 2023/MATH 449/Final Project/')
arrests = read.csv("Arrests.csv", header=TRUE)

# Transform response variable to binary
arrests$released = ifelse(arrests$released == "Yes", 1, 0)
```

```r
# Transform categorical explanatory variables to binary
arrests$sex = ifelse(arrests$sex == "Male", 1, 0)
arrests$employed = ifelse(arrests$employed == "Yes", 1, 0)
arrests$citizen = ifelse(arrests$citizen == "Yes", 1, 0)
arrests$colour = ifelse(arrests$colour == "White", 1, 0)

# Fit the logistic regression model with all predictors
fit0 = glm(released ~ colour + year + age + sex + employed + citizen +
checks,
           family = "binomial", data = arrests)
summary(fit0)
```

```
##
## Call:
## glm(formula = released ~ colour + year + age + sex + employed +
##     citizen + checks, family = "binomial", data = arrests)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3909   0.3579   0.4320   0.6047   1.7067
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  9.371821  56.717803   0.165    0.869
## colour       0.389109   0.085663   4.542 5.56e-06 ***
## year        -0.004218   0.028379  -0.149    0.882
## age          0.002236   0.004631   0.483    0.629
## sex          0.007317   0.150189   0.049    0.961
## employed     0.757302   0.084735   8.937  < 2e-16 ***
## citizen      0.576519   0.104246   5.530 3.20e-08 ***
## checks      -0.364101   0.025984 -14.013  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4776.3  on 5225  degrees of freedom
## Residual deviance: 4299.1  on 5218  degrees of freedom
## AIC: 4315.1
##
## Number of Fisher Scoring iterations: 5
```

$$logit[P(Y=1)] = 0.974332 + 0.389109Co - 0.004218Y + 0.002236A + 0.007317S + 0.757302E + 0.576519Ci - 0.364101Ch$$

- Co represents the variable “colour”.
- Y represents the variable “year”.

- A represents the variable “age”.
- S represents the variable “sex”.
- E represents the variable “employed”.
- Ci represents the variable “citizen”.
- Ch represents the variable “checks”.

Select the best subset of variables. Perform a diagnostic on the best model. Perform all possible inferences you can think about.

- We can use **`stepAIC()`** to select the best subset of variables, as it uses forward selection and backward elimination.

```
# Use stepAIC function to determine best subset of variables
stepAIC(fit0)
```

```
## Start:  AIC=4315.07
## released ~ colour + year + age + sex + employed + citizen + checks
##
##             Df Deviance    AIC
## - sex        1   4299.1 4313.1
## - year       1   4299.1 4313.1
## - age        1   4299.3 4313.3
## <none>           4299.1 4315.1
## - colour     1   4319.2 4333.2
## - citizen    1   4328.6 4342.6
## - employed   1   4376.2 4390.2
## - checks     1   4501.9 4515.9
##
## Step:  AIC=4313.07
## released ~ colour + year + age + employed + citizen + checks
##
##             Df Deviance    AIC
## - year       1   4299.1 4311.1
## - age        1   4299.3 4311.3
## <none>           4299.1 4313.1
## - colour     1   4319.3 4331.3
## - citizen    1   4328.7 4340.7
## - employed   1   4376.8 4388.8
## - checks     1   4504.1 4516.1
##
## Step:  AIC=4311.09
## released ~ colour + age + employed + citizen + checks
##
##             Df Deviance    AIC
## - age        1   4299.3 4309.3
## <none>           4299.1 4311.1
## - colour     1   4319.5 4329.5
## - citizen    1   4330.8 4340.8
```

```
## - employed  1   4376.8 4386.8
## - checks    1   4504.4 4514.4
##
## Step:  AIC=4309.32
## released ~ colour + employed + citizen + checks
##
##            Df Deviance    AIC
## <none>          4299.3 4309.3
## - colour    1   4319.7 4327.7
## - citizen   1   4330.8 4338.8
## - employed  1   4376.9 4384.9
## - checks    1   4504.9 4512.9

##
## Call:  glm(formula = released ~ colour + employed + citizen + checks,
##     family = "binomial", data = arrests)
##
## Coefficients:
## (Intercept)       colour     employed      citizen       checks
##      1.0047       0.3891       0.7537       0.5684      -0.3628
##
## Degrees of Freedom: 5225 Total (i.e. Null);  5221 Residual
## Null Deviance:       4776
## Residual Deviance: 4299  AIC: 4309

# Fit model with best subset
fit1 = glm(released ~ colour + employed + citizen + checks,
           family = "binomial", data = arrests)
summary(fit1)

##
## Call:
## glm(formula = released ~ colour + employed + citizen + checks,
##     family = "binomial", data = arrests)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3580   0.3579   0.4316   0.6061   1.6982
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.00474    0.12741   7.886 3.12e-15 ***
## colour       0.38915    0.08521   4.567 4.95e-06 ***
## employed     0.75367    0.08398   8.974  < 2e-16 ***
## citizen      0.56839    0.09916   5.732 9.91e-09 ***
## checks      -0.36283    0.02573 -14.101  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 4776.3  on 5225  degrees of freedom
## Residual deviance: 4299.3  on 5221  degrees of freedom
## AIC: 4309.3
##
## Number of Fisher Scoring iterations: 5
```

From using **stepAIC()**, we get the model, M_1:

$$logit[P(Y=1)] = 1.00474 + 0.38915Co + 0.75367E + 0.56839Ci - 0.36283Ch$$

By looking at the summary, we can see that race, employment status, citizenship status, and the number of checks are statistically significant at the 0.01 level. Therefore, we can infer that there is a strong association between these variables and the likelihood of an arrestee being released with a summons.

Before moving on, we should check to see if we have unbalanced data.

```
table(arrests$released)
```

```
##
##    0    1
##  892 4334
```

```
prop.released = 892/(892+4334)
cat("\n Proportion of not released:", prop.released)
```

```
##
##  Proportion of not released: 0.170685
```

```
I=which(arrests$released==1)
J=sample(I, 1000)
J1=which(arrests$released==0)
arrests1=rbind(arrests[J,],arrests[J1,])
```

By calculating the proportions of "released", we can see that we do have unbalanced data, where only 17% of the observation are not released. Therefore, we can create a subset of the data by sampling random 1000 observations from the "released" category (I) and combining them with all observations from the "not released" category (J1) to create a new, more balanced data-set called "**arrests1**".

Now we can fit a new model, with the same subset of variables used in **fit1**.

```
fit2 = glm(released ~ colour + employed + citizen + checks, family =
"binomial", data = arrests1)
(g=summary(fit2))
```

```
##
## Call:
## glm(formula = released ~ colour + employed + citizen + checks,
##     family = "binomial", data = arrests1)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7541  -1.0681   0.6953   0.9561   2.3906
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.54515    0.17256  -3.159 0.001582 **
## colour       0.43286    0.11381   3.803 0.000143 ***
## employed     0.76506    0.11568   6.614 3.74e-11 ***
## citizen      0.64400    0.13540   4.756 1.97e-06 ***
## checks      -0.37555    0.03322 -11.304  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2616.7  on 1891  degrees of freedom
## Residual deviance: 2307.6  on 1887  degrees of freedom
## AIC: 2317.6
##
## Number of Fisher Scoring iterations: 4
```

```
# Coefficients
alpha = g$coef[1,1]
beta1 = g$coef[2,1]
beta2 = g$coef[3,1]
beta3 = g$coef[4,1]
beta4 = g$coef[5,1]
```

Our new model, M_2:

$$logit[P(Y = 1)] = -0.54515 + 0.43286Co + 0.76506E + 0.64400Ci - 0.37555Ch$$

Important note: Because our new, balanced **arrests1** data is combined with 1000 *random* observations from the "released" category, our estimate coefficient will vary after each run. However, this doesn't change its direction, and the change in magnitude is very small. Therefore, the overall interpretation and significance remains constant.

- Now we can carry out a Goodness-of-Fit Test with comparison to the null, to see if this is an adequate fit.

H_0: All parameters in model M_2 not in the null model M_0 are zero.

H_1: At least one of them is not zero.

```
# Comparison with the null
fit_null = glm(released ~ 1, family = "binomial", data = arrests1)
anova(fit2, fit_null, test = "LRT")
```

```
## Analysis of Deviance Table
##
```

```
## Model 1: released ~ colour + employed + citizen + checks
## Model 2: released ~ 1
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1      1887     2307.6
## 2      1891     2616.7 -4  -309.06 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Since we have a very small p-value that is less than a 0.001 level of significance, we have very strong evidence to reject the null hypothesis and conclude that our model is adequate.

We can now make inferences on our model.

- Firstly, the α coefficient is -0.54515. The negative coefficient suggests that, when all other predictor variables are held constant, the log-odds of being released with a summons decrease by approximately -0.54515 units.
- We can also look at the Confidence Intervals.

```
# Confidence Interval
confint(fit2, level = 0.95)

## Waiting for profiling to be done...

##                 2.5 %     97.5 %
## (Intercept) -0.8856221 -0.2087613
## colour       0.2099057  0.6562302
## employed     0.5391066  0.9927555
## citizen      0.3797008  0.9108276
## checks      -0.4411456 -0.3108684
```

From the 95% Likelihood-Ratio CI, we get a range of possible values for which the true $\pi(x)$ lies for each coefficient of our model, while holding other predictors constant. Firstly, we are 95% confident that the estimated coefficient for "colour" lies between 0.2099 and 0.6562. This means that individuals of different races have a significant impact on the log-odds of being released with a summons, but the exact nature and direction of this impact depends on the specific race categories. Next, the estimated coefficient for employment status falls between 0.5391 and 0.9928, and the estimated coefficient for citizenship status falls between 0.3797 and 0.9108. Because the probabilities are higher, we could infer that being employed, or being a citizen have a higher log-odds compared to being unemployed, or non-citizens, respectively. Finally, the range for "checks" is $-0.4411 \leq \pi(x) \leq -0.3109$. This means that an increase in the number of checks is associated with a significant decrease in the log-odds of being released with a summons.

- Finally, we can look at the multiplicative effects, otherwise known as the odds ratio.

```
# Multiplicative effect
effect_co = exp(beta1)
effect_em = exp(beta2)
effect_ci = exp(beta3)
effect_ch = exp(beta4)
```

```
effect.df = data.frame(Predictor = c("Colour", "Employed", "citizen",
"checks"),
                       Effect = c(effect_co, effect_em, effect_ci,
effect_ch))
effect.df
```

```
##   Predictor    Effect
## 1    Colour 1.5416578
## 2  Employed 2.1491204
## 3   citizen 1.9040729
## 4    checks 0.6869105
```

The multiplicative effect represents the odds of the response, y, occurring, for every one-unit increase in the predictor variables, x. To begin, the odds ratio of 1.5417 for "colour" means that the odds of being released with a summons increases by 54% when the "colour" variable increases by one. In other words, the odds of a white person being released is approximately 54% higher than that of a black person. Next, the odds of being released with a summons is 2.1491 times higher for an employed person compared to an unemployed person. Moving on, the odds of being released with a summons is approximately 90% higher for a citizen compared to a non-citizen. Finally, the odds ratio of 0.6869 for "checks" means that the odds of being released decreases by approximately 69% for a person who has more checks, compared to someone who has less checks.

Use the new model to make predictions.

We can use the **`predict()`** function to make predictions.

```
pihat = predict(fit2, type = "response")
```

It uses the prediction equation $\pi(x)$:

$$\pi(x) = \frac{e^{-0.54515+0.43286Co+0.76506E+0.64400Ci-0.37555Ch}}{1+e^{-0.54515+0.43286Co+0.76506E+0.64400Ci-0.37555Ch}}$$

We'll use four random observations as examples to further elaborate.

```
Co=0; E=0; Ci=0; Ch=0
(exp(alpha + beta1*Co + beta2*E + beta3*Ci + beta4*Ch))/
  (1+exp(alpha + beta1*Co + beta2*E + beta3*Ci + beta4*Ch))
```

```
## [1] 0.3669897
```

For a black, unemployed person who is not a Canadian citizen with no checks, the probability of being released with a summons is 0.37.

```
Co=1; E=1; Ci=1; Ch=1
(exp(alpha + beta1*Co + beta2*E + beta3*Ci + beta4*Ch))/
  (1+exp(alpha + beta1*Co + beta2*E + beta3*Ci + beta4*Ch))
```

```
## [1] 0.7152882
```

For a white, employed, Canadian citizen with only 1 check, the probability of being released with a summons is 0.72.

```
Co=1; E=0; Ci=1; Ch=6
(exp(alpha + beta1*Co + beta2*E + beta3*Ci + beta4*Ch))/
  (1+exp(alpha + beta1*Co + beta2*E + beta3*Ci + beta4*Ch))

## [1] 0.1516643
```

For a white, unemployed, Canadian citizen with 6 checks, the probability of being released with a summons is 0.15.

```
Co=0; E=1; Ci=1; Ch=5
(exp(alpha + beta1*Co + beta2*E + beta3*Ci + beta4*Ch))/
  (1+exp(alpha + beta1*Co + beta2*E + beta3*Ci + beta4*Ch))

## [1] 0.266226
```

For a black, employed, Canadian citizen with 5 checks, the probability of being released with a summons is 0.29.

Use different π_0 as a cut-off point and create a confusion table.

There are two different ways to choose π_0 as a cut-off point.

- $\pi_0 = 0.5$

```
# pi0 = 0.5
y = as.numeric(arrests1$released > 0)
yhat = as.numeric(pihat > 0.50)
(confusion1 = addmargins(table(y, yhat), 2))

##    yhat
## y      0    1  Sum
##   0  548  344  892
##   1  276  724 1000

n11 = confusion1[1,1]
n12 = confusion1[1,2]
n1r = confusion1[1,3]
n21 = confusion1[2,1]
n22 = confusion1[2,2]
n2r = confusion1[2,3]
```

- $\pi_0 = \frac{\#(Y=1)}{n}$

```
# pi0 = #(Y=1)/n
pi_0 = n2r/(n1r+n2r)
y = as.numeric(arrests1$released > 0)
yhat = as.numeric(pihat > pi_0)
(confusion2 = addmargins(table(y, yhat), 2))
```

```
##    yhat
## y     0    1  Sum
##   0  569  323  892
##   1  300  700 1000
```

```
m11 = confusion2[1,1]
m12 = confusion2[1,2]
m1r = confusion2[1,3]
m21 = confusion2[2,1]
m22 = confusion2[2,2]
m2r = confusion2[2,3]
```

```
# metrics for confusion1
sens1 = n22 / n2r
spec1 = n11 / n1r
acc1 = (n11 + n22) / (n1r + n2r)
err1 = (n12 + n21) / (n1r + n2r)
```

```
# metrics for confusion2
sens2 = m22 / m2r
spec2 = m11 / m1r
acc2 = (m11 + m22) / (m1r + m2r)
err2 = (m12 + m21) / (m1r + m2r)
```

```
metrics.df = data.frame(Table = c("Confusion 0", "Confusion 1"),
                        Sensitivity = c(sens1, sens2),
                        Specificity = c(spec1, spec2),
                        Accuracy = c(acc1, acc2),
                        Error = c(err1, err2))
metrics.df
```

```
##         Table Sensitivity Specificity  Accuracy     Error
## 1 Confusion 0       0.724   0.6143498 0.6723044 0.3276956
## 2 Confusion 1       0.700   0.6378924 0.6707188 0.3292812
```

Sensitivity represents the proportion of true positives. In this case, it is the correctly identified released individuals among those actually released. **Confusion0** with the cut-off point of 0.50 has a slightly higher sensitivity, which indicates that it is better at correctly identifying individuals who are actually released.

Specificity represents the proportion of true negatives, i.e. correctly identified not released individuals among those actually not released. **Confusion1** with the cut-off point of 0.53 has a slightly higher specificity, which means that it is better at correctly identifying individuals who are actually not released. We can observe that there is an inverse relationship between sensitivity and specificity; as sensitivity increases, the specificity decreases, and vise-versa.

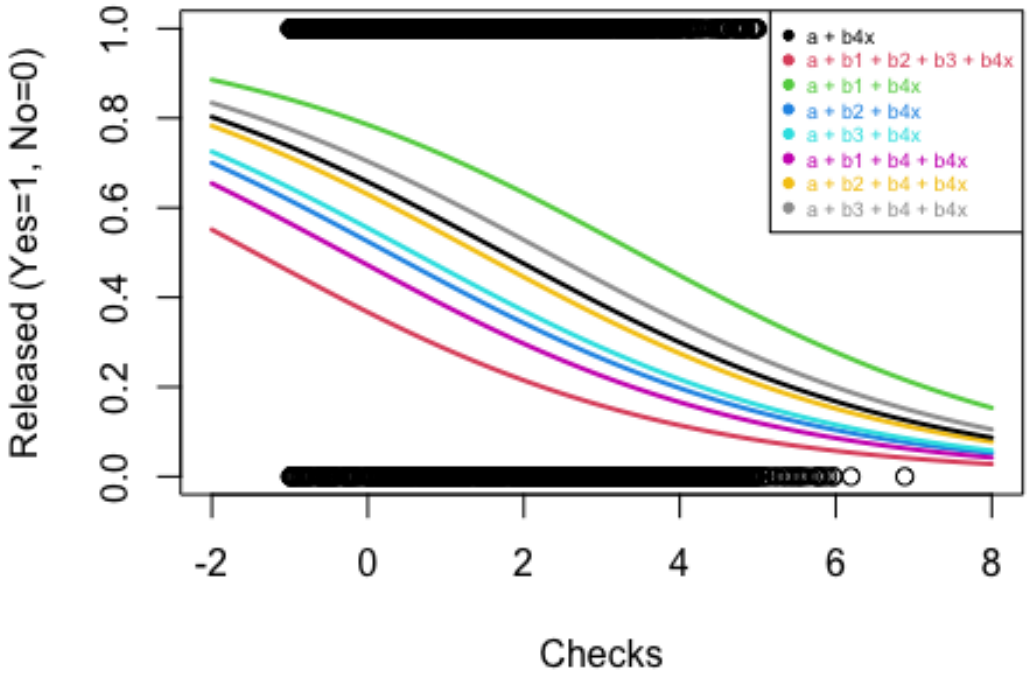
Accuracy measures the overall correctness of models' predictions, whereas error is the complacent of accuracy. Here, both error and accuracy have very similar performances at

both cut-off points, where they correctly classify approximately 67% of the observations, and incorrectly classify approximately 33% of the observations.

Perform visualization of data and models.

- Model different combinations of **fit2** against response

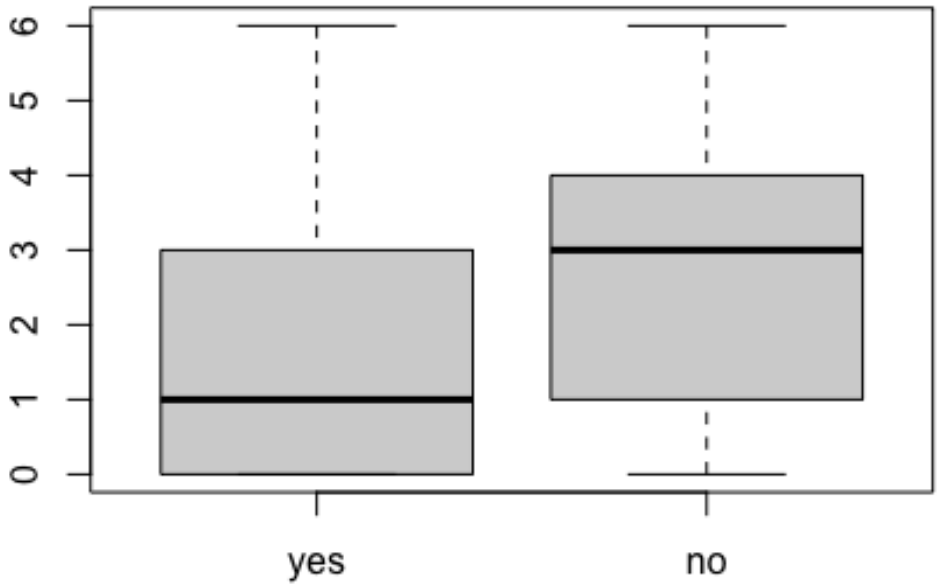
```
with(arrests1, {
  plot(jitter(checks, 5), released,
       xlim = c(-2, 8), ylim = c(0, 1),
       xlab = "Checks", ylab = "Released (Yes=1, No=0)")
})

# logit=alpha: curve(plogis(coef(fit2)[1]), col=1, add=TRUE, lwd=2)
curve(plogis(coef(fit2)[1] + coef(fit2)[5]*x), col=2, add=TRUE, lwd=2)
curve(plogis(coef(fit2)[1] + coef(fit2)[2] + coef(fit2)[3] + coef(fit2)[4] +
coef(fit2)[5]*x),
      col=3, add=TRUE, lwd=2)

curve(plogis(coef(fit2)[1] + coef(fit2)[4] + coef(fit2)[5]*x), col=4,
add=TRUE, lwd=2)
curve(plogis(coef(fit2)[1] + coef(fit2)[3] + coef(fit2)[5]*x), col=5,
add=TRUE, lwd=2)
curve(plogis(coef(fit2)[1] + coef(fit2)[2] + coef(fit2)[5]*x), col=6,
add=TRUE, lwd=2)

curve(plogis(coef(fit2)[1] + coef(fit2)[2] + coef(fit2)[4] +
coef(fit2)[5]*x),
      col=7, add=TRUE, lwd=2)
curve(plogis(coef(fit2)[1] + coef(fit2)[3] + coef(fit2)[4] +
coef(fit2)[5]*x),
      col=8, add=TRUE, lwd=2)
curve(plogis(coef(fit2)[1] + coef(fit2)[2] + coef(fit2)[3] +
coef(fit2)[5]*x),
      col=9, add=TRUE, lwd=2)

lgnd = c("a + b4x",
         "a + b1 + b2 + b3 + b4x",
         "a + b1 + b4x",
         "a + b2 + b4x",
         "a + b3 + b4x",
         "a + b1 + b4 + b4x",
         "a + b2 + b4 + b4x",
         "a + b3 + b4 + b4x")
legend("topright", lgnd, pch=19,
       col=1:9, text.col=1:41, cex = 0.55)
```



This plot shows the relationship between the response variable “released” and the predictor variable “checks”. We do so by plotting curves, each of which represent all the different possible combinations of our model M_2. This helps us better understand the different coefficient combinations on the predicted probabilities.

- Boxplot of “checks”

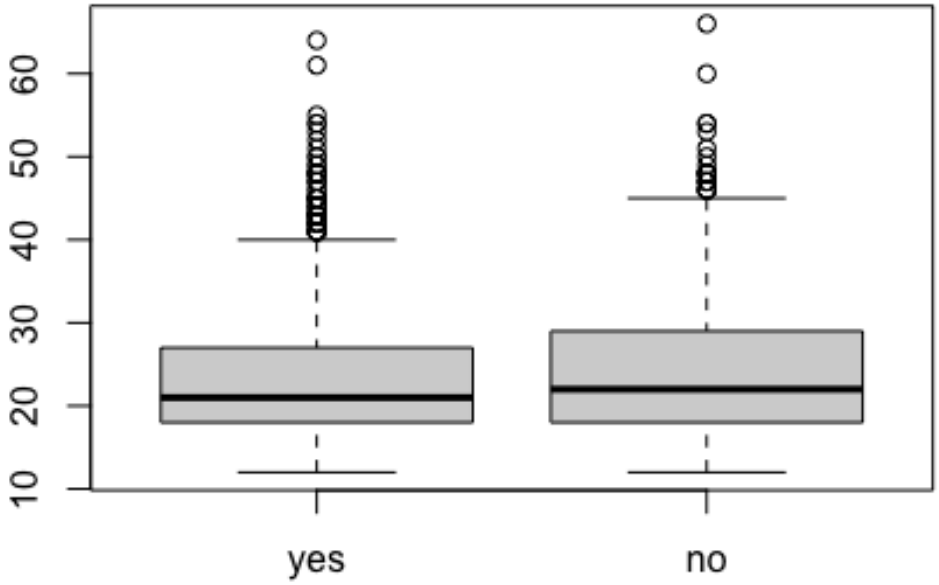
```
arrestyes=subset(arrests1, released==1)
arrestno=subset(arrests1, released==0)
boxplot(list(yes=arrestyes$checks, no=arrestno$checks))
```



Based on the box-plot, we can observe that the medians of the “yes” and “no” groups appear to be different. This suggests that there might be a difference in the central tendencies of the “checks” variable between the two groups. There is also an overlap between the “yes” and “no” groups, which indicates that both the groups have similar values for the number of checks.

- Boxplot for "age"

```
boxplot(list(yes=arrestyes$age, no=arrestno$age))
```

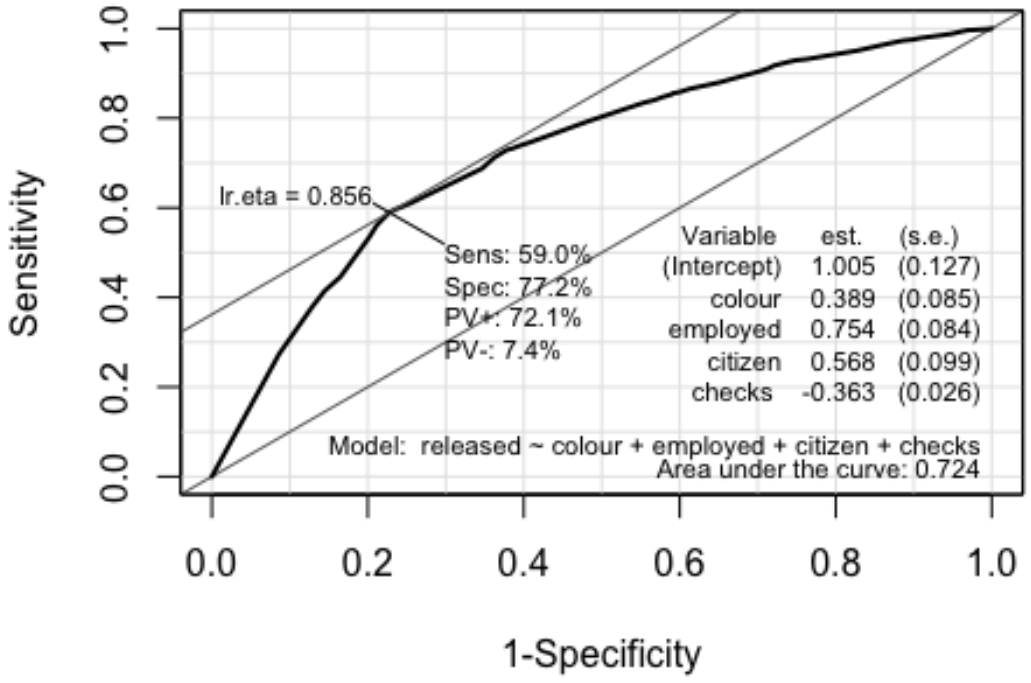


From the box-plot above, we can see that both groups have similar distributions, with nearly identical medians and both having outliers. This suggests that age may not be a strong predictor in determining whether an individual is being released with a summons.

Plot the ROC curve, find AUC, and the best cutoff point for classification.

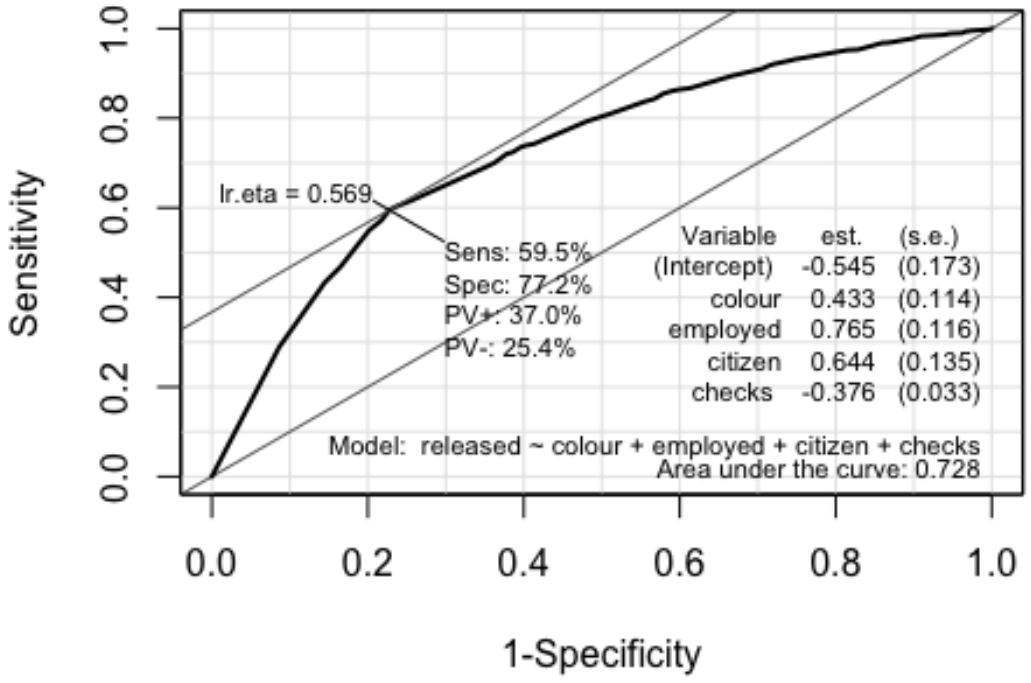
- ROC using M_1

```
attach(arrests)

ROC(form=released ~ colour + employed + citizen + checks, plot="ROC")
```



- ROC using M_2

```
detach(arrests)
attach(arrests1)

ROC(form=released ~ colour + employed + citizen + checks, plot="ROC")
```



The cut-off point represents the threshold for the probability where individuals are classified as being released. For the original **arrests** data, the cut-off point is 0.856, while the cut-off point for the new **arrests1** data is 0.569. Next, model M_2 has a slightly higher AUC of 0.728 compared to the AUC of 0.724 for M_1. This indicates that M_2 has a slightly better predictive power, and is better at distinguishing between released and not released individuals. Additionally, Model M_2 has a slightly higher sensitivity of 59.5% compared to M_1's 59%. This indicates that M_2 is better at correctly identifying individuals who are actually released. Therefore, the cut-off point of 0.569 is better.

Perform LOOCV and k-fold cross-validation.

```
# LOOCV
library(caret)
set.seed(123)

# Define training control
train.control = trainControl(method = "LOOCV")

model = released ~ colour + employed + citizen + checks

# Train the model
model = train(model, data = arrests1, method = "glm", trControl =
train.control)
```

```
## Warning in train.default(x, y, weights = w, ...): You are trying to do
## regression and your outcome only has two possible values Are you trying to
do
## classification? If so, use a 2 level factor as your outcome column.

model

## Generalized Linear Model
##
## 1892 samples
##    4 predictor
##
## No pre-processing
## Resampling: Leave-One-Out Cross-Validation
## Summary of sample sizes: 1891, 1891, 1891, 1891, 1891, 1891, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.4603809  0.1494607  0.4229552
```

The RMSE (Root Mean Square Error) of 0.461002 means that the model's predictions are off from the actual values by approximately 0.461 units. Next, the Rsquared of 0.1471657 suggests that the model explains approximately 14.72% of the total variation in the outcome. Finally, the MAE (Mean Absolute Error) is 0.4240248, which means that the average absolute difference between the predicted values and the actual values is 0.424.

```
# k-fold cross-validation.
cv.binary(fit1)

##
## Fold:  2 7 3 4 9 8 5 10 1 6
## Internal estimate of accuracy = 0.828
## Cross-validation estimate of accuracy = 0.828

cv.binary(fit2)

##
## Fold:  6 3 2 8 9 4 10 1 5 7
## Internal estimate of accuracy = 0.672
## Cross-validation estimate of accuracy = 0.672
```

For model M_1 the set of folds is (2, 7, 3, 4, 9, 8, 5, 10, 1, 6) and the internal estimate of accuracy is 0.828. This means that when the model is used on **arrests**, it achieves an accuracy of 0.828. The cross-validation estimate of accuracy for the **fit1** folds is 0.828. This estimate is obtained by training the model on a subset of the data (the folds) and evaluating it on the remaining fold. For model M_2 the set of folds are (10, 7, 8, 5, 4, 9, 6, 3, 1, 2), and the internal and the cross-validation estimate 0.672.

Try the probit link and the identity links to model data.

```
fit_probit = glm(released ~ colour + employed + citizen + checks,
                 family = binomial(link = "probit"), data = arrests1)
summary(fit_probit)
```

```
##
## Call:
## glm(formula = released ~ colour + employed + citizen + checks,
##     family = binomial(link = "probit"), data = arrests1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7536  -1.0733   0.6956   0.9605   2.4860
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.32103    0.10470  -3.066 0.002169 **
## colour       0.26146    0.06935   3.770 0.000163 ***
## employed     0.46398    0.07036   6.595 4.27e-11 ***
## citizen      0.38515    0.08177   4.710 2.47e-06 ***
## checks      -0.22819    0.02000 -11.412  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2616.7  on 1891  degrees of freedom
## Residual deviance: 2309.0  on 1887  degrees of freedom
## AIC: 2319
##
## Number of Fisher Scoring iterations: 4
```

With the probit link, we get the model, M_P:

$$logit[P(Y = 1)] = -0.24879 + 0.18044Co + 0.48248E + 0.36462Ci - 0.23333Ch$$

Model M_P is very similar to our model M_2.

```
# fit_ident = glm(released ~ colour + employed + citizen + checks,
                 #family = binomial(link = "identity"), data = arrests1)
#summary(fit_ident)
```

Using arrests1, we are not able to use the identity link. This could be due to a lack of convergence or co-linearity. Therefore, between the two, the probit link is better for this data.