Analysis of influencing factors for force use in New York City’s Stop, Question, and Frisk

Yu Liang, Wenzheng Yin, Haiyang Lu

Group 23

All group members contributed equally

Introduction

This report focus on Analysis of **influencing factors for force use** in New York City’s Stop, Question, and Frisk. It consists of two main parts: exploratory part and analysis part.

In exploratory part, we draw the density plot for each variable in the dataset(1.1), handle the missing values(1.2), do feature engineering(1.3), draw "force use-univariate" plot(1.4), Spearman correlation matrix(1.5) and do the Add1(),drop1() and F-test(1.6).

In analysis part, we build model construction(2.1), model diagnostics(2.2), model assessment(2.3) and discussion(2.4).

The interval estimation part is put at the last and is relatively independent but can also be used as a supplement to the analysis part.

1 Exploratory part

From the summary of *d2*, we can see that the data types of many variables are inconsistent with their actual meanings, so firstly we change them.

From the summary, we can see that:

1. There are 8 variables including NA values: race2, gender, age2, daytime, inout2, offunif2, typeofid2, othpers2.
2. Apart from **force** and **age2**, other variables can be treated as categorical variables.

```
# change the data type
d2$force <- as.factor(d2$force)
d2$gender <- as.factor(d2$gender)
d2$age2 <- as.integer(d2$age2)
d2$daytime <- as.factor(d2$daytime)
d2$inout2 <- as.factor(d2$inout2)
d2$offunif2 <- as.factor(d2$offunif2)
d2$typeofid2 <- as.factor(d2$typeofid2)
d2$othpers2 <- as.factor(d2$othpers2)
d2$year <- as.numeric(d2$year)
# delete the ignorable columns(from the describtion in assignment)
d2n<-subset(d2,select = -c(force2,pct))
summary(d2n)
```

```
##      force            race2          gender             age2
##   0      :3910239   white: 492688   0    :4495436   Min.   :10
```

```
##  1      : 736051   black:2886843   1   : 348159   1st Qu.:19
##  3      : 156732   hisp :1215401   NA's: 140796   Median :24
##  2      : 122898   asian: 152053                  Mean   :28
##  5      :  31590   other: 235105                  3rd Qu.:34
##  6      :  17769   NA's :   2301                  Max.   :90
##  (Other):   9112                                  NA's   :41534
##
##  daytime         inout2         ac_incid    ac_time     offunif2
##  0   :2472074   0   :3809789   N:2204888   N:3158281   0   :1396968
##  1   :1871287   1   :1147150   Y:2779503   Y:1826110   1   :3586909
##  NA's: 641030   NA's:  27452                           NA's:    514
##
##
##
##
##  typeofid2      othpers2       cs_objcs    cs_descr    cs_casng    cs_lkout
##  O   :  84217   0   :3806808   N:4850690   N:4116587   N:3562617   N:4151151
##  P   :2640092   1   :1163298   Y: 133701   Y: 867804   Y:1421774   Y: 833240
##  R   : 107497   NA's:  14285
##  V   :2133368
##  NA's:  19217
##
##
##  cs_cloth    cs_drgtr    cs_furtv    cs_vcrim    cs_bulge    cs_other
##  N:4773826   N:4522192   N:2801105   N:4588325   N:4543022   N:3944522
##  Y: 210565   Y: 462199   Y:2183286   Y: 396066   Y: 441369   Y:1039869
##
##

##       year
##  Min.   :2003
##  1st Qu.:2006
##  Median :2009
##  Mean   :2008
##  3rd Qu.:2011
##  Max.   :2013
##
```

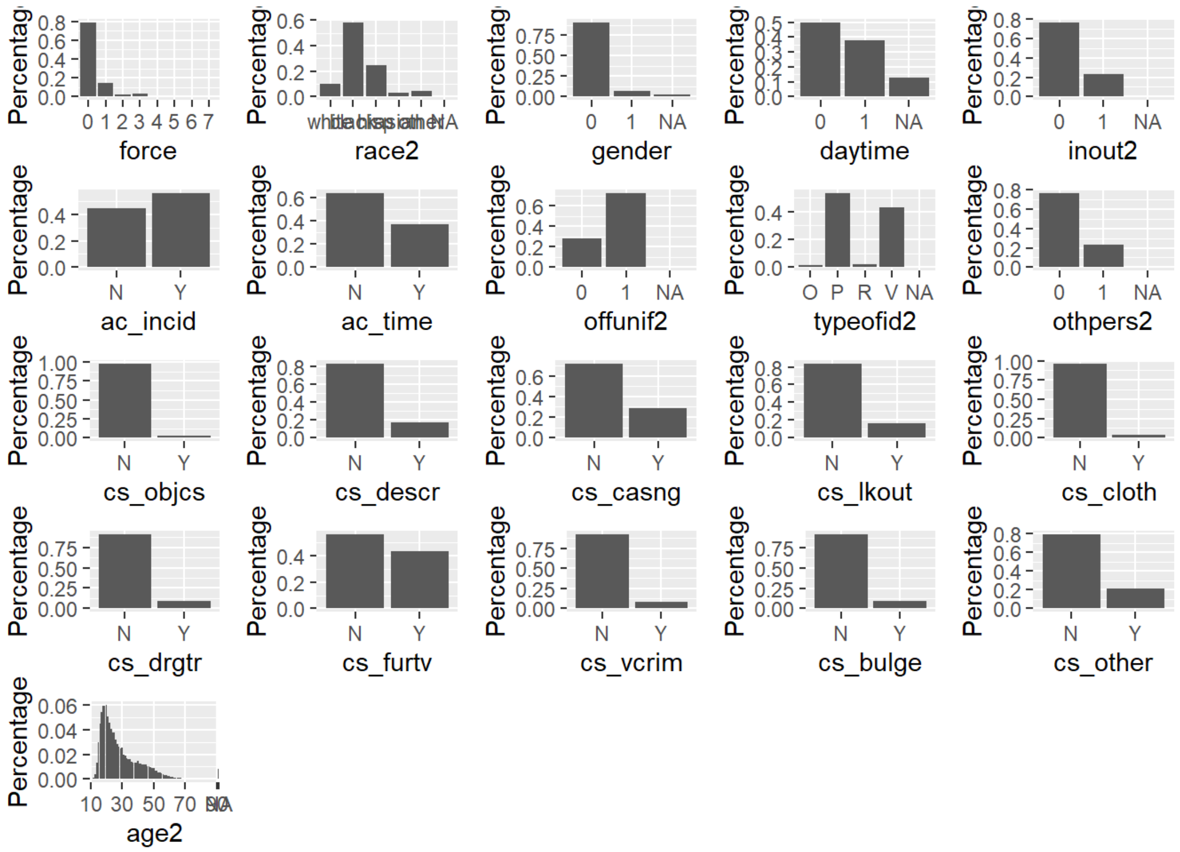
1.1 Density plot

From the bar and density plots, we get below information for different variables.

1. For force:
   In most cases, the police didn't use force in a stop and frisk interaction. The total number of remaining samples being used different types of force only accounts for a small portion. Therefore, we decide to focus on the problem **if** the police will use force in different situations.
   Specifically, we will consider the force taking values from 1~7 as the same value 1, representing that force has been used.
2. For variables related with personal information including race2, gender and age2:
   From the bar and density plots, we can notice that the civilian investigated are not evenly distributed based on the above information. For example, the black, the male or the pedestrians aged around 20 take up a larger proportion than others. What' more, these 3

variables exhibit an obvious skewed distribution but it's reasonable since all the samples in the dataset are "stopped" and not generated from random sampling.

3. For variables starting with 'cs':
   They describes Civilians' suspicious behaviors. And most of these variables take the value of "N". Since the accumulated number of "Y" in these variables can reflect the civilian's **suspicion level**, we want to establish a new feature so that all of this information can be aggregated.
   Specifically, we will create a new variable **cs** whose value is the accumulated number of "Y" in these 10 "cs-"starting variables. Obviously the larger **cs** is, the more suspicious we consider the civilian to be. Here we define **cs** as a *numeric* variable because we think the sort of its numerical values is meaningful. We do not assign different weights to the 'cs' variables for simplicity though different behaviors may contribute differently to the suspicion level.
   By creating **cs**, we ignore the specific suspicious behavior while retaining a measure of its level of suspicion, which can reduce noise in our subsequent analysis.
4. A remarkable variable is **daytime**, the missing data percentage of which is 13% and is much larger than any other variable. In addition, another variable "ac_time" also includes the impact of time on force use. So we decide to remove this variable in subsequent analysis.
5. As for ac-incid and ac-time, there is no missing value. And there is no much difference between the proportion of "N" and "Y" for both.
6. inouts, offunit2 and otherpers2, all of two variables with little missing data are observed a slight skew.
7. typeofid2 has four levels, the first three of which can be regarded as the same level: agree to provide ID. Specifically, we will transfer typeofid2 into a binary variable, denoting "R" as "0" and the other three as "1".
8. As for years, we think it cannot be used for prediction. Because intuitively, the passage of years does not directly affect the force use. Even if there is an impact, it is also indirect (such as through the implementation of some certain policies). Therefore, we won't be able to explain the meaning of its coefficient in the model, nor can we guarantee the explanatory power of the model for future data. As a result we decide to delete this variable before we build the model.
9. For **skewness**, since the data set is quite large, there are still enough observations in each categorical group. So we decide to leave it without special processing.



1.2 Handle the missing value

1.2.1 The background of Multiple imputation

Multiple imputation typically involves three main steps:

1. Imputation: Impute missing values in each dataset. Imputation can be done using various methods, such as regression imputation, predictive mean matching, or other imputation models. Each imputed dataset represents one possible set of missing data replacements.
2. Analysis: Analyze each imputed dataset separately using the statistical analysis you intend to perform (e.g., regression analysis, hypothesis testing, or data visualization).
3. Pooling: Combine the results from each imputed dataset to obtain a single set of parameter estimates and standard errors. This is typically done using Rubin's rules, which take into account both within-imputation variability and between-imputation variability.

But since this dataset is too large and pooling will be very computationally expensive so we only analysis on one imputed dataset.

In order to achieve better results in multiple imputation, we hope to utilize the relationships between various variables as much as possible. Therefore we aim to exclude situations where missing values center at one specific case. We draw the cross tabulation of 2 variables to achieve this objective. We've checked all possible combinations of variables. Here we take **race2** and **gender** for example.

```
table(d2n$race2, is.na(d2n$gender)) # why we can impute the data
```

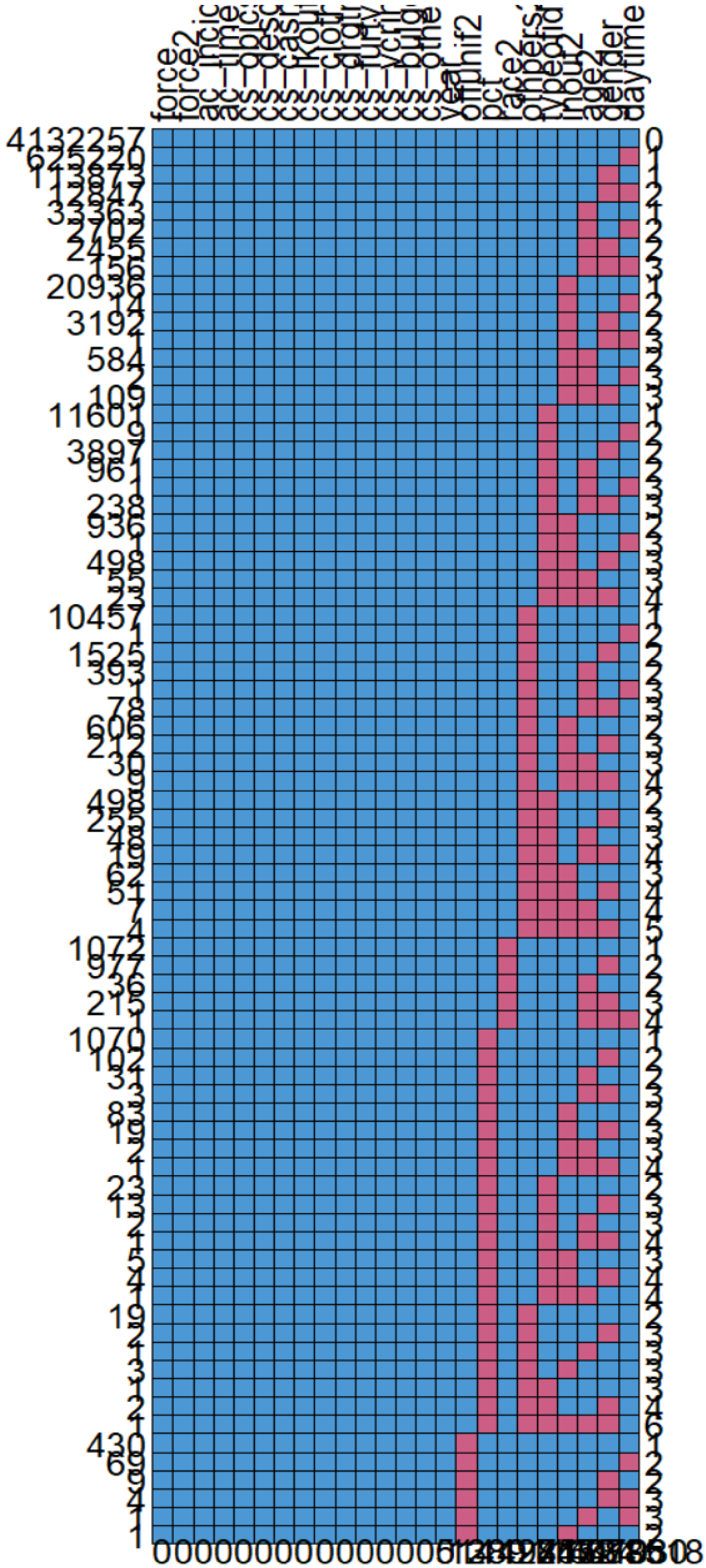
```
##
##           FALSE    TRUE
##   white  489222    3466
##   black 2864694   22149
##   hisp  1203612   11789
##   asian  150988    1065
##   other  133971  101134
```

There is no abnormality displayed in the table.

What's more we can draw the missing data pattern.

Each column in this section represents a variable. The red squares mark the specific missing pattern. The right column represents the number of missing variables, which is also the number of the red squares. The left column represents the number of missing samples for the corresponding pattern.

From this plot we can see that although sometimes multiple variables are missing together, there are no dominant patterns. So we can assume that the data is not **MNAR**.



1.2.2 Imputation with mice

```r
set.seed(123)
# Perform imputation
mice_data <- mice(d2n, m = 1)
# Export imputed data
imputed_data <- complete(mice_data,1)
```

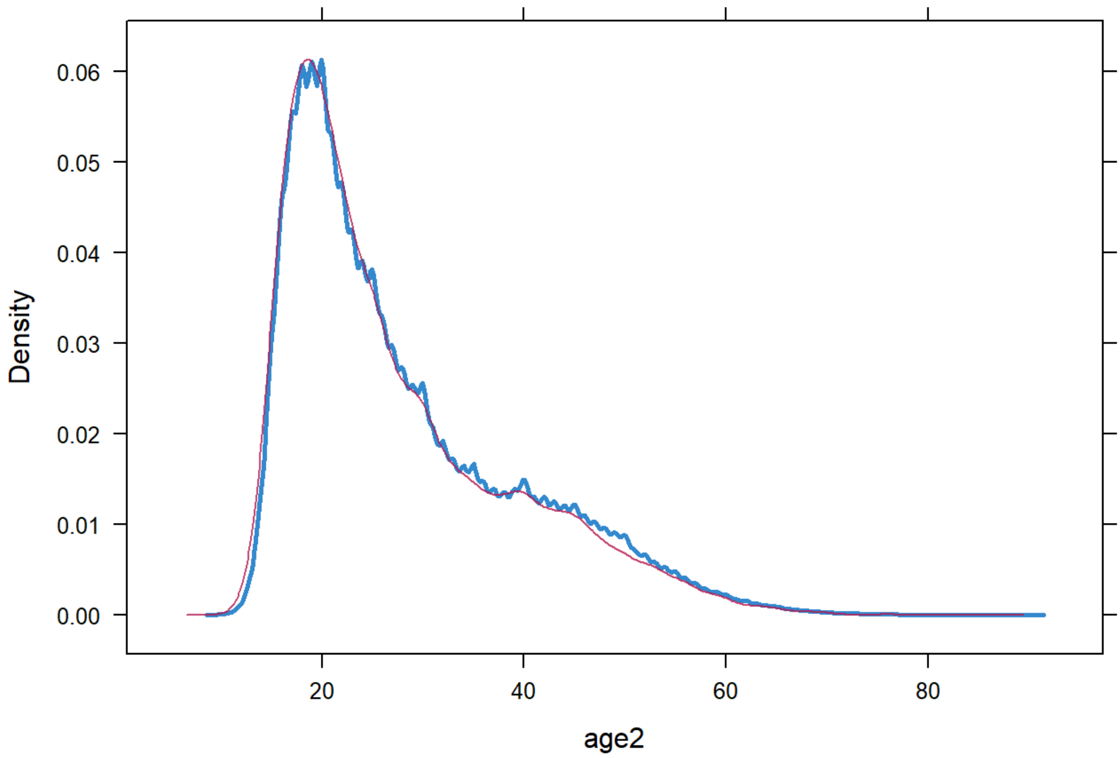
```r
# imputation methods for different variables
methods_used<-mice_data$method
print(methods_used)
```

```
##    force     race2    gender      age2   daytime    inout2
##       "" "polyreg"  "logreg"     "pmm"  "logreg"  "logreg"
## ac_incid   ac_time  offunif2 typeofid2  othpers2  cs_objcs
##       ""        ""  "logreg" "polyreg"  "logreg"        ""
## cs_descr  cs_casng  cs_lkout  cs_cloth  cs_drgtr  cs_furtv
##       ""        ""        ""        ""        ""        ""
## cs_vcrim  cs_bulge  cs_other      year
##       ""        ""        ""        ""
```

1.2.3 Multiple imputation diagnostics

We do the imputation with *mice()* and we can do diagnostics directly. We observe the distribution of **age2** before and after the imputation and it doesn't change much, which implies the quality of the imputed dataset is acceptable. We didn't show all the results but they are all similar.

```r
densityplot(mice_data)
```



1.3 Feature engineering

From the discussion above, we build new features based on the imputed data.

```
# build "cs"
for (i in (ncol(imputed_data) - 10):(ncol(imputed_data)-1)) {
  imputed_data[,i] <- ifelse(imputed_data[,i] == "Y", 1, ifelse(imputed_data[,i]
== "N", 0, imputed_data[,i]))
}
imputed_data$cs <- rowSums(imputed_data[, (ncol(imputed_data) - 10):
(ncol(imputed_data)-1)])

# change force&typeofid2 into a 0-1 variable
imputed_data$force <- ifelse(imputed_data$force != 0, 1, 0)
imputed_data$force <- as.factor(imputed_data$force)
imputed_data$typeofid2 <- ifelse(imputed_data$typeofid2 == "R",0,1)
imputed_data$typeofid2 <- as.factor(imputed_data$typeofid2)

#delete the "cs-" starting variables
imputed_data <- subset(imputed_data,select = -c(cs_objcs, cs_descr, cs_casng,
cs_lkout, cs_cloth, cs_drgtr, cs_furtv, cs_vcrim, cs_bulge,
cs_other,year,daytime))

summary(imputed_data)
```

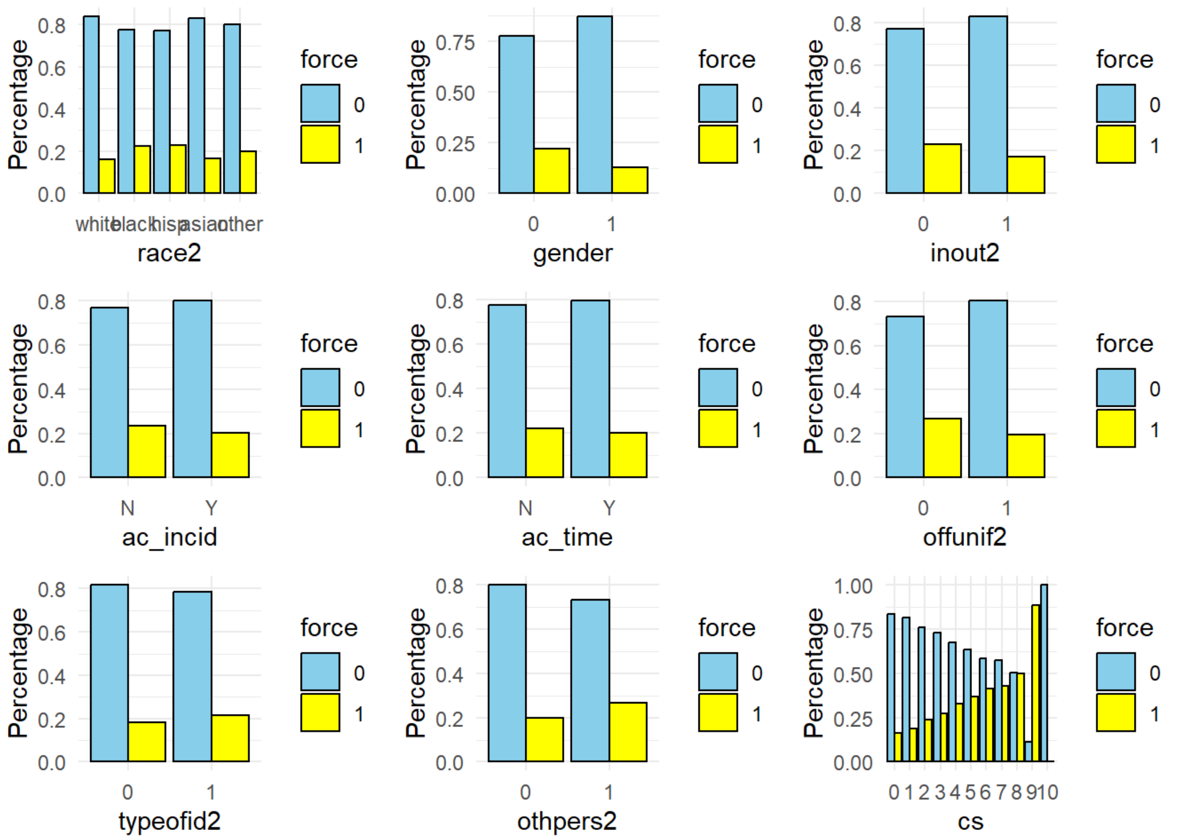
```
##  force          race2          gender           age2       inout2       ac_incid
##  0:3910239   white: 492976   0:4624573   Min.   :10   0:3830915   N:2204888
##  1:1074152   black:2888035   1: 359818   1st Qu.:19   1:1153476   Y:2779503
##              hisp :1215959               Median :24
##              asian: 152136               Mean   :28
##              other: 235285               3rd Qu.:34
```

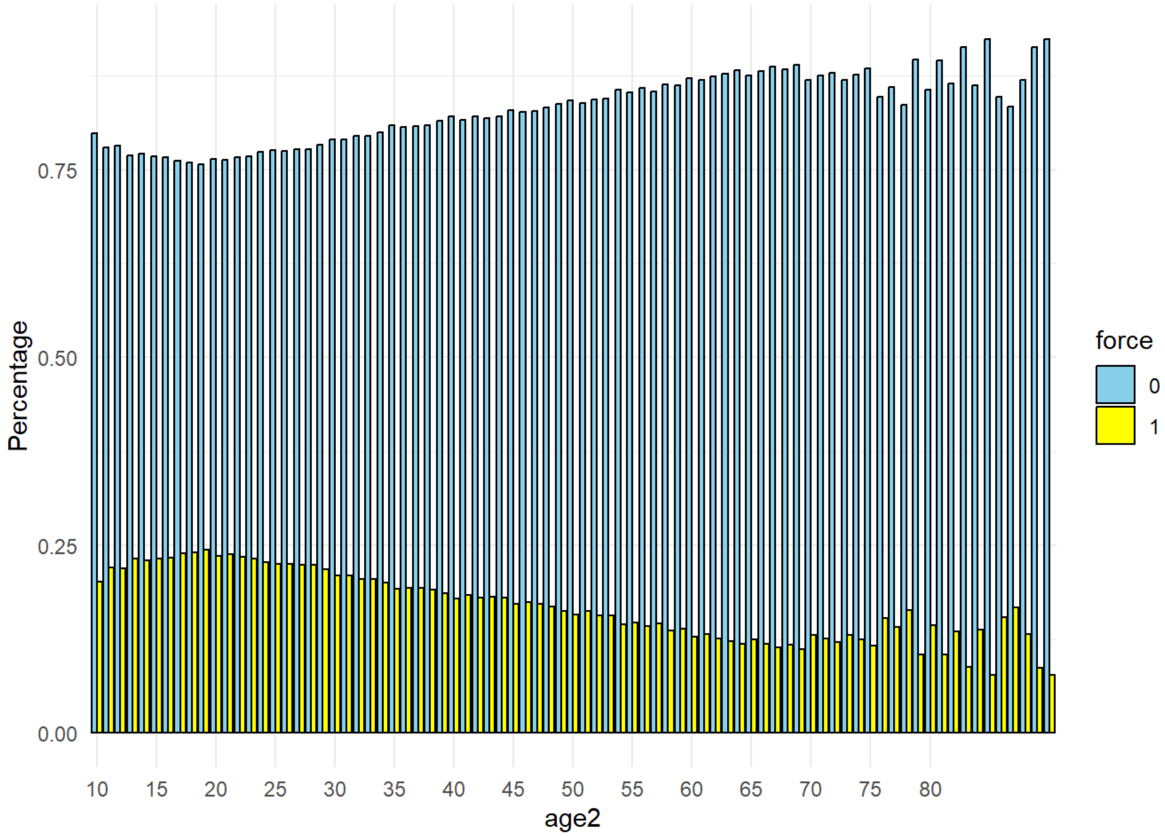
```
##                                         Max.   :90
##  ac_time     offunif2    typeofid2   othpers2         cs
##  N:3158281   0:1397119   0: 108154   0:3816720   Min.   : 0.000
##  Y:1826110   1:3587272   1:4876237   1:1167671   1st Qu.: 1.000
##                                                  Median : 1.000
##                                                  Mean   : 1.603
##                                                  3rd Qu.: 2.000
##                                                  Max.   :10.000
```

1.4 Force use - univariate plot

From these plot we can form a basic impression on how each variable affect the force.

For example, there is a significant difference in force use in different **race2**, **gender**,...,groups, which implies that these variables maybe strong predictors in our following analysis and shows their specific impact on force use. For example, force is used more on gender0 group(male) than gender1 group(female). For other variables the analysis is similar.

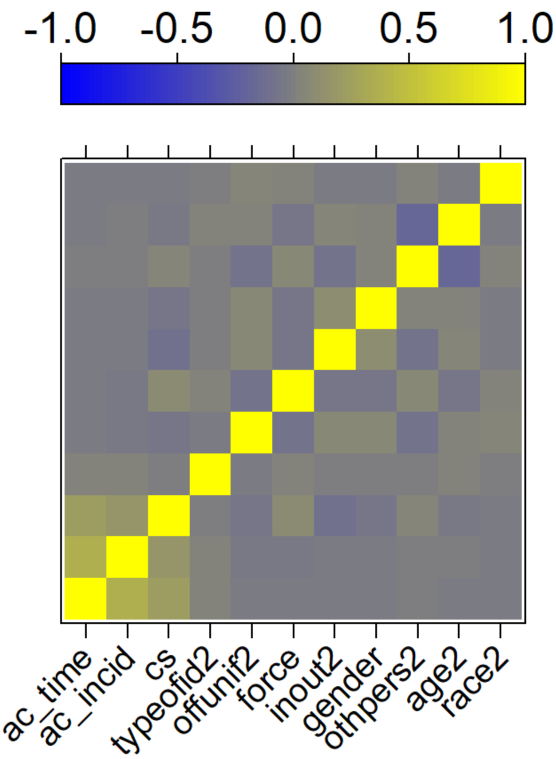




1.5 Spearman correlation matrix

We draw the correlation plot to explore the potential collinearity between variables.

From the plot we can observe a positive correlation between ac_time and ac_incid but not very strong. There is not enough evidence for us to delete any variable in this step.



1.6 Add1(),drop1() and F-test

We test the model with only an intercept against the alternative where a single predictor is included. Since all F-test results are significant, we can say all these variables are strong predictors from an explorative view. So we will include all of them in the following analysis part.

```
form1 <- force~race2 + gender + age2 + inout2 + ac_incid + ac_time + offunif2 +
typeofid2 + othpers2 + cs

model_full <- glm(form1,family = binomial(link = "logit"),
                  contrasts = list(race2="contr.sum", gender="contr.sum",
inout2="contr.sum", ac_incid="contr.sum", ac_time="contr.sum",
offunif2="contr.sum", typeofid2="contr.sum", othpers2="contr.sum"),
                  data=imputed_data)

model_null <- glm(force~1, family = binomial(link = "logit"), data =
imputed_data)

AddTermModel<-add1(model_null,model_full, test="LRT")
DropTermModel<-drop1(model_full, test="LRT")

print(AddTermModel)
```

```
## Single term additions
##
## Model:
## force ~ 1
##             Df Deviance     AIC   LRT  Pr(>Chi)
## <none>         5195303 5195305
## race2        4  5181029 5181039 14274 < 2.2e-16 ***
## gender       1  5175751 5175755 19552 < 2.2e-16 ***
## age2         1  5172397 5172401 22906 < 2.2e-16 ***
## inout2       1  5177891 5177895 17412 < 2.2e-16 ***
## ac_incid     1  5187503 5187507  7800 < 2.2e-16 ***
## ac_time      1  5192446 5192450  2857 < 2.2e-16 ***
## offunif2     1  5165268 5165272 30035 < 2.2e-16 ***
## typeofid2    1  5194544 5194548   759 < 2.2e-16 ***
## othpers2     1  5171525 5171529 23778 < 2.2e-16 ***
## cs           1  5153827 5153831 41476 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
print(DropTermModel)
```

```
## Single term deletions
##
## Model:
## force ~ race2 + gender + age2 + inout2 + ac_incid + ac_time +
##     offunif2 + typeofid2 + othpers2 + cs
##             Df Deviance     AIC   LRT  Pr(>Chi)
## <none>         5034030 5034058
```

```
## race2       4  5051809 5051829 17779 < 2.2e-16 ***
## gender      1  5046586 5046612 12556 < 2.2e-16 ***
## age2        1  5045196 5045222 11166 < 2.2e-16 ***
## inout2      1  5042549 5042575  8520 < 2.2e-16 ***
## ac_incid    1  5045160 5045186 11131 < 2.2e-16 ***
## ac_time     1  5038025 5038051  3995 < 2.2e-16 ***
## offunif2    1  5056014 5056040 21984 < 2.2e-16 ***
## typeofid2   1  5034900 5034926   870 < 2.2e-16 ***
## othpers2    1  5047371 5047397 13341 < 2.2e-16 ***
## cs          1  5073723 5073749 39693 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2 Analysis part

We choose the logit regression model for our study since it's a binary classification problem.

2.1 Model construction

2.1.1 logit model

First of all, we will build a *logit_m1* with all variables included.

Intuitively, there may exist a nonlinear effect in **age**. Because for too young and too old civilians the police should be less inclined to use force than adult civilians.

What's more, we consider an interaction effect between **ac_incid** and **ac_time**. We guess if the stop occurs both in an area of high crime incidence and at a time of day that fit crime incidence, the police may be more alert than usual and be more likely to use force.

We build three more models which consider a nonlinear effect(*logit_md2*), an interaction effect(*logit_m3*), a nonlinear and interaction effect(*logit_md4*).

```
# 0-1 response with no interaction & splines
form1 <-
force~race2+gender+age2+inout2+ac_incid+ac_time+offunif2+typeofid2+othpers2+cs
# 0-1 response with ac_incid*ac_time
form2 <-
force~race2+gender+age2+inout2+ac_incid*ac_time+offunif2+typeofid2+othpers2+cs
# 0-1 response with splines
form3 <-
force~race2+gender+ns(age2,df=3)+inout2+ac_incid+ac_time+offunif2+typeofid2+othpe
rs2+cs
# 0-1 response with ac_incid*ac_time & splines
form4 <-
force~race2+gender+ns(age2,df=3)+inout2+ac_incid*ac_time+offunif2+typeofid2+othpe
rs2+cs

# build the logit models
logit_md1 <- glm(form1, family = binomial(link=logit), data = imputed_data)
logit_md2 <- glm(form2, family = binomial(link=logit), data = imputed_data)
logit_md3 <- glm(form3, family = binomial(link=logit), data = imputed_data)
logit_md4 <- glm(form4, family = binomial(link=logit), data = imputed_data)
```

2.1.2 Anova

From the result of anova, we can say that both interaction and nonlinear effects are worth consideration.

```
anova(logit_md1, logit_md2, test = "LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: force ~ race2 + gender + age2 + inout2 + ac_incid + ac_time +
##     offunif2 + typeofid2 + othpers2 + cs
## Model 2: force ~ race2 + gender + age2 + inout2 + ac_incid * ac_time +
##     offunif2 + typeofid2 + othpers2 + cs
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1   4984377    5034030
## 2   4984376    5032768  1   1262.1 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(logit_md1, logit_md3, test = "LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: force ~ race2 + gender + age2 + inout2 + ac_incid + ac_time +
##     offunif2 + typeofid2 + othpers2 + cs
## Model 2: force ~ race2 + gender + ns(age2, df = 3) + inout2 + ac_incid +
##     ac_time + offunif2 + typeofid2 + othpers2 + cs
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1   4984377    5034030
## 2   4984375    5033076  2   953.46 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(logit_md3, logit_md4, test = "LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: force ~ race2 + gender + ns(age2, df = 3) + inout2 + ac_incid +
##     ac_time + offunif2 + typeofid2 + othpers2 + cs
## Model 2: force ~ race2 + gender + ns(age2, df = 3) + inout2 + ac_incid *
##     ac_time + offunif2 + typeofid2 + othpers2 + cs
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1   4984375    5033076
## 2   4984374    5031810  1   1265.9 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2.1.3 Explanation of the models

1. All the variables are significant and there is no variable with a very small coefficient, so their values are meaningful.
2. From summary of model, we can find all these four models' number of Fisher Scoring iterations are 4, which means that the convergence speed of the model is relatively fast, and the convergence is relatively good. The Std. Errors are also small, which means the estimation of coefficients is reliable.
3. The value of variables represents the way how they influence the probability of being used force when stopped. Take *logit_md1* for example, the coefficient of **gender1** means that when all other variables remain the same, the *log-odds* decreases by 0.54 for **gender1**(female).

```
summary(logit_md1)
```

```
## Call:
## glm(formula = form1, family = binomial(link = logit), data = imputed_data)

Coefficients:
              Estimate Std. Error  z value Pr(>|z|)
(Intercept) -1.6038421  0.0097862 -163.888  < 2e-16 ***
race2black   0.4788686  0.0042118  113.696  < 2e-16 ***
race2hisp    0.4734980  0.0045139  104.898  < 2e-16 ***
race2asian   0.0495047  0.0080017    6.187 6.14e-10 ***
race2other   0.3346889  0.0065613   51.009  < 2e-16 ***
gender1     -0.5502069  0.0051920 -105.973  < 2e-16 ***
age2        -0.0108429  0.0001041 -104.142  < 2e-16 ***
inout21     -0.2571139  0.0028258  -90.989  < 2e-16 ***
ac_incidY   -0.2545219  0.0024146 -105.409  < 2e-16 ***
ac_timeY    -0.1611732  0.0025583  -63.000  < 2e-16 ***
offunif21   -0.3565714  0.0023820 -149.694  < 2e-16 ***
typeofid21   0.2330935  0.0080771   28.859  < 2e-16 ***
othpers21    0.2972746  0.0025488  116.633  < 2e-16 ***
cs           0.2547699  0.0012617  201.928  < 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: 5195303  on 4984390  degrees of freedom
Residual deviance: 5034030  on 4984377  degrees of freedom
AIC: 5034058

Number of Fisher Scoring iterations: 4
```

```
summary(logit_md2)
```

```
Call:
glm(formula = form2, family = binomial(link = logit), data = imputed_data)
```

```
Coefficients:
                     Estimate Std. Error  z value Pr(>|z|)
(Intercept)        -1.5836751  0.0098027 -161.556  < 2e-16
race2black          0.4779903  0.0042123  113.475  < 2e-16
race2hisp           0.4736561  0.0045144  104.921  < 2e-16
race2asian          0.0493668  0.0080027    6.169 6.88e-10
race2other          0.3346747  0.0065624   50.999  < 2e-16
gender1            -0.5507426  0.0051927 -106.061  < 2e-16
age2               -0.0107979  0.0001041 -103.692  < 2e-16
inout21            -0.2560329  0.0028263  -90.589  < 2e-16
ac_incidY          -0.3063126  0.0028321 -108.157  < 2e-16
ac_timeY           -0.2925151  0.0045513  -64.270  < 2e-16
offunif21          -0.3546655  0.0023829 -148.837  < 2e-16
typeofid21          0.2342723  0.0080783   29.000  < 2e-16
othpers21           0.2965903  0.0025492  116.346  < 2e-16
cs                  0.2526608  0.0012630  200.049  < 2e-16
ac_incidY:ac_timeY  0.1933215  0.0054740   35.316  < 2e-16

(Intercept)        ***
race2black         ***
race2hisp          ***
race2asian         ***
race2other         ***
gender1            ***
age2               ***
inout21            ***
ac_incidY          ***
ac_timeY           ***
offunif21          ***
typeofid21         ***
othpers21          ***
cs                 ***
ac_incidY:ac_timeY ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 5195303  on 4984390  degrees of freedom
Residual deviance: 5032768  on 4984376  degrees of freedom
AIC: 5032798

Number of Fisher Scoring iterations: 4
```

```
summary(logit_md3)
```

```
Call:
glm(formula = form3, family = binomial(link = logit), data = imputed_data)

Coefficients:
                   Estimate Std. Error  z value Pr(>|z|)
(Intercept)       -1.899207   0.011892 -159.704  < 2e-16 ***
```

```
race2black          0.479222   0.004212  113.773  < 2e-16 ***
race2hisp           0.470215   0.004515  104.134  < 2e-16 ***
race2asian          0.050205   0.008002    6.274 3.52e-10 ***
race2other          0.334300   0.006562   50.947  < 2e-16 ***
gender1            -0.548261   0.005193 -105.572  < 2e-16 ***
ns(age2, df = 3)1 -0.280739   0.007550  -37.182  < 2e-16 ***
ns(age2, df = 3)2 -0.478025   0.020979  -22.786  < 2e-16 ***
ns(age2, df = 3)3 -0.941574   0.025438  -37.014  < 2e-16 ***
inout21            -0.252748   0.002829  -89.327  < 2e-16 ***
ac_incidY          -0.255172   0.002415 -105.667  < 2e-16 ***
ac_timeY           -0.161744   0.002559  -63.216  < 2e-16 ***
offunif21          -0.353866   0.002384 -148.444  < 2e-16 ***
typeofid21          0.237279   0.008079   29.371  < 2e-16 ***
othpers21           0.303390   0.002557  118.673  < 2e-16 ***
cs                  0.255407   0.001262  202.394  < 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: 5195303  on 4984390  degrees of freedom
Residual deviance: 5033076  on 4984375  degrees of freedom
AIC: 5033108

Number of Fisher Scoring iterations: 4
```

```
summary(logit_md4)
```

```
Call:
glm(formula = form4, family = binomial(link = logit), data = imputed_data)

Coefficients:
                    Estimate Std. Error  z value Pr(>|z|)
(Intercept)        -1.878834   0.011906 -157.808  < 2e-16
race2black          0.478341   0.004213  113.551  < 2e-16
race2hisp           0.470364   0.004516  104.155  < 2e-16
race2asian          0.050057   0.008003    6.255 3.98e-10
race2other          0.334286   0.006563   50.936  < 2e-16
gender1            -0.548795   0.005194 -105.660  < 2e-16
ns(age2, df = 3)1  -0.278933   0.007551  -36.940  < 2e-16
ns(age2, df = 3)2  -0.474870   0.020978  -22.636  < 2e-16
ns(age2, df = 3)3  -0.939098   0.025436  -36.920  < 2e-16
inout21            -0.251658   0.002830  -88.925  < 2e-16
ac_incidY          -0.307045   0.002832 -108.404  < 2e-16
ac_timeY           -0.293288   0.004552  -64.434  < 2e-16
offunif21          -0.351944   0.002385 -147.579  < 2e-16
typeofid21          0.238468   0.008080   29.514  < 2e-16
othpers21           0.302718   0.002557  118.392  < 2e-16
cs                  0.253296   0.001263  200.514  < 2e-16
ac_incidY:ac_timeY  0.193619   0.005474   35.368  < 2e-16

(Intercept)        ***
```

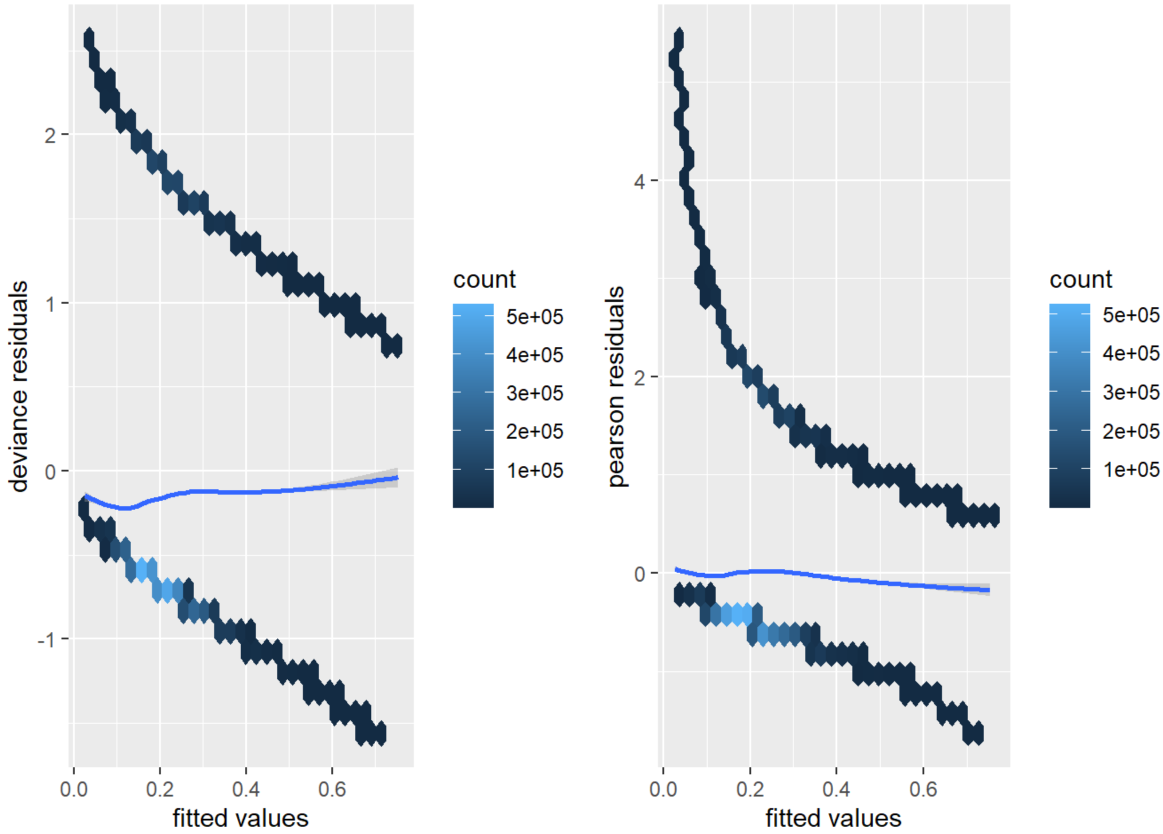
```
race2black         ***
race2hisp          ***
race2asian         ***
race2other         ***
gender1            ***
ns(age2, df = 3)1  ***
ns(age2, df = 3)2  ***
ns(age2, df = 3)3  ***
inout21            ***
ac_incidY          ***
ac_timeY           ***
offunif21          ***
typeofid21         ***
othpers21          ***
cs                 ***
ac_incidY:ac_timeY ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 5195303  on 4984390  degrees of freedom
Residual deviance: 5031810  on 4984374  degrees of freedom
AIC: 5031844

Number of Fisher Scoring iterations: 4
```

2.2 Model diagnostic

2.2.1 raw deviance residuals

Take the *logit_md1* for example. We plot its deviance/person residuals.

In this plot, we get two curves for both kinds of residuals. It's because the "force2" is a binary variable and consequently for any fitted response there are only 2 values for the deviance. As a result, this raw residual plot do not exhibit the same pattern and could not be used to test homoscedasticity which we might expect in linear regression model's residual plot.



2.2.2 binned residual plot

A binned residual plot is a graphical method for assessing the goodness of fit and the distribution of residuals in a regression model, especially when dealing with a large number of data points. It is commonly used in linear regression and generalized linear models, including logistic regression.

The binned residual-fitted plot involves the following steps:

1. Divide the range of fitted values into a series of bins or intervals.
2. Calculate the mean(or median) of the residuals for each bin. These are the observed residuals within each bin.
3. Plot the observed residuals against the corresponding bin's center or average fitted value. This results in a scatter plot with bins.
4. Optionally, overlay reference lines, such as a horizontal line at 0, to assess the overall distribution of residuals.

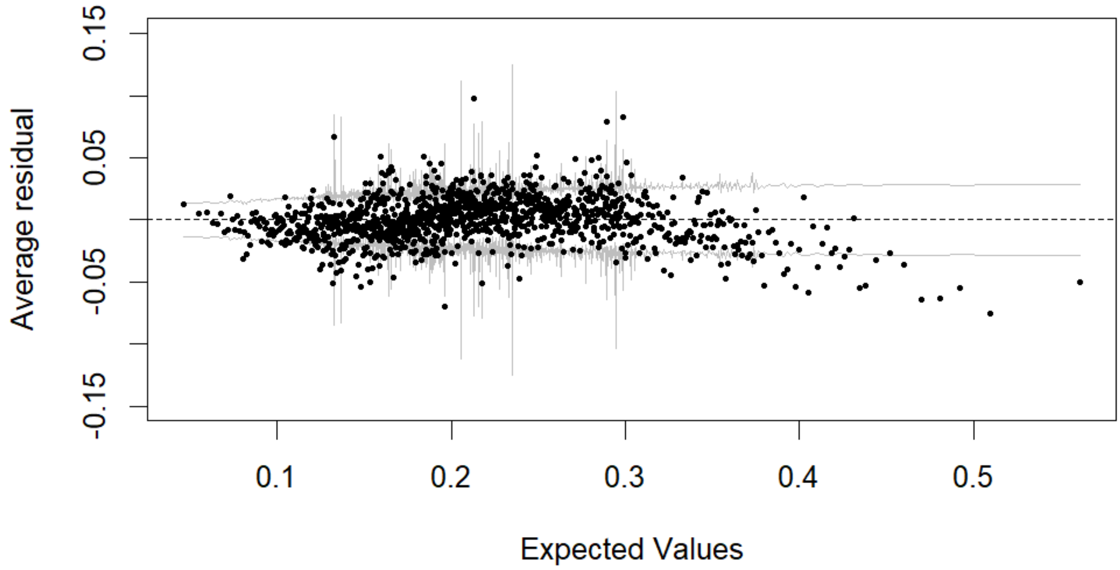
2.2.2.1 residual-fitted plot

We use fitting on subsets with 1,500,000 observations to obtain binned residual plots, as the entire dataset exceeded the computational power of binnedplot() and it will cause errors. "1,500,000" is the largest number of subsamples we tried that did not exceed the computational power of binnedplot().

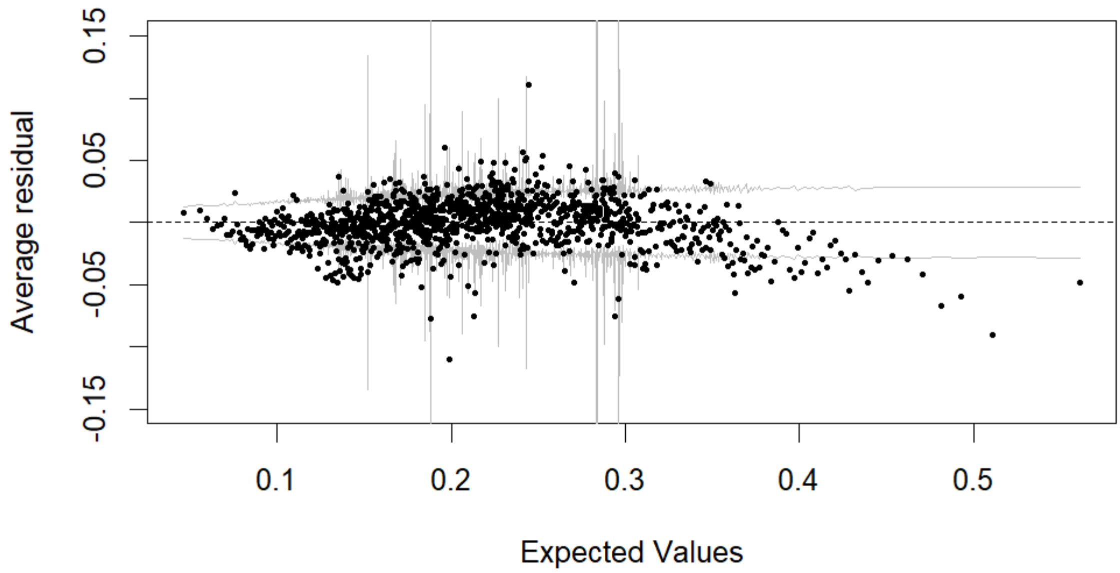
This is a valid way of doing it, even if choosing residuals in a specific way could bias the residual plots. After repeating these above process several times, we found the residual plots derived roughly the same. So we think this can also reflect our concerns because the subset is generated through random sampling and big enough.

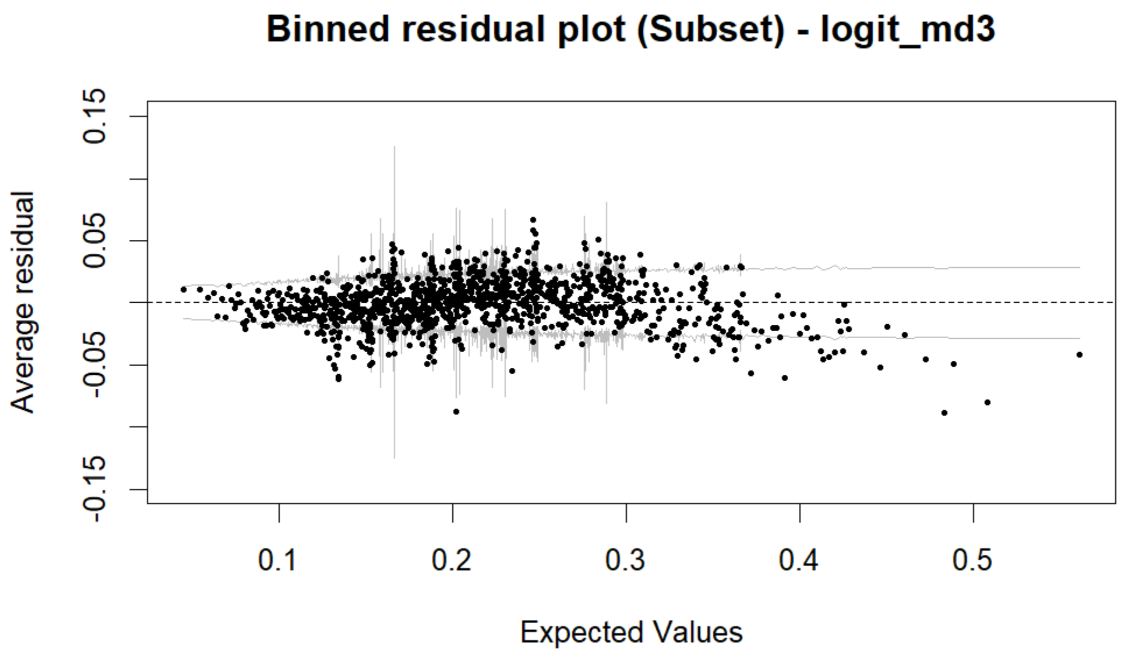
The binned residuals plots show that all 4 models are good fits since the means are around 0 and there are no significant patterns or overdispersion for the plot, which means **the model assumptions are met**.

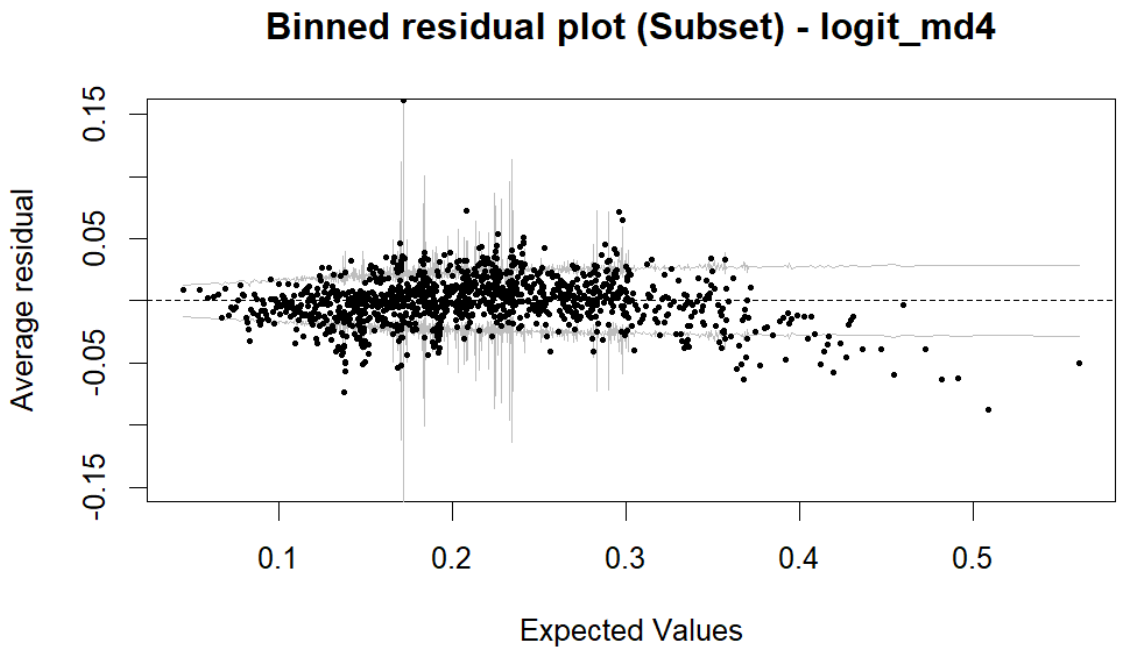
Binned residual plot (Subset) - logit_md1



Binned residual plot (Subset) - logit_md2





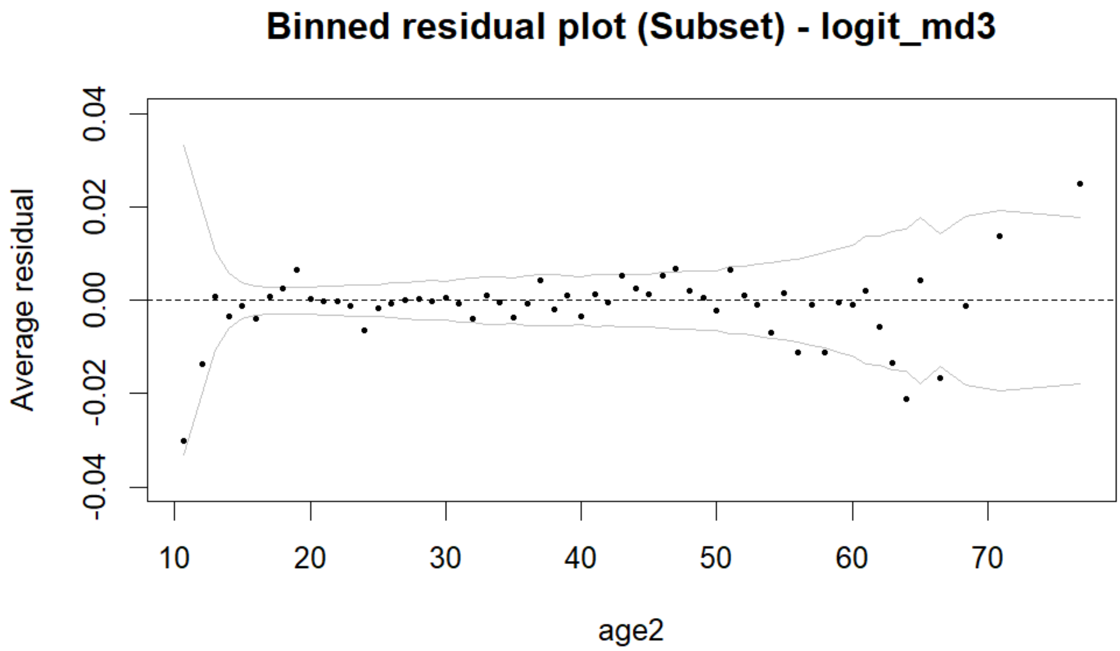


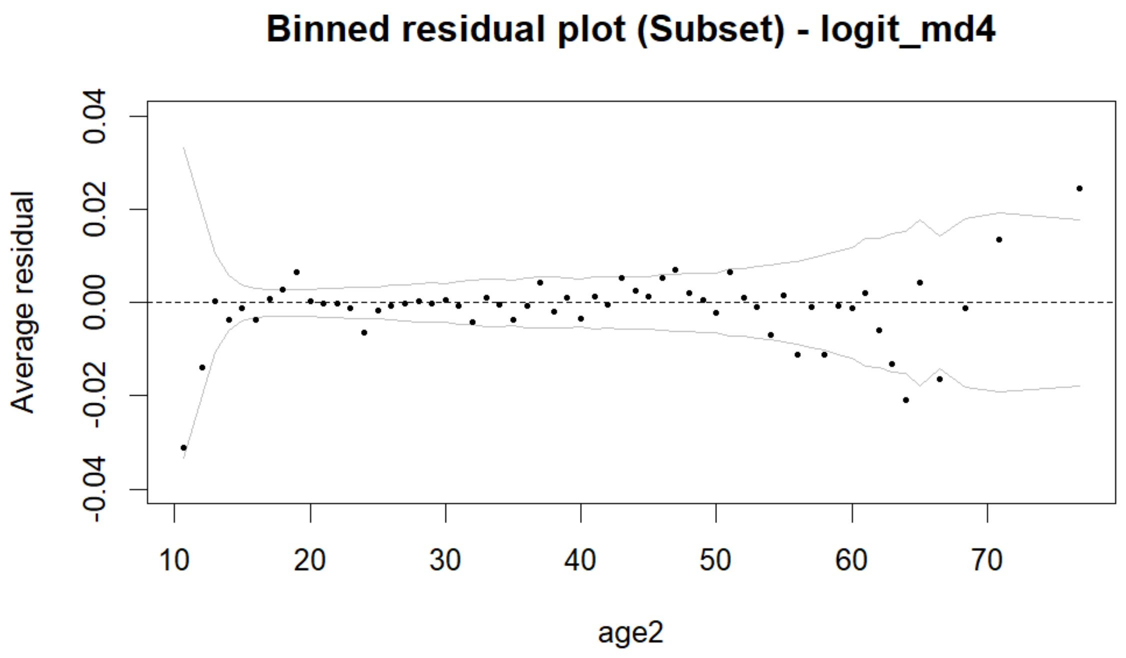
2.2.2.2 residual-univariate plot

For continous variable we make the residual-univariate plot. The residual should be around 0 and there should be no significant pattern.

The inclusion of splines is not changing the residual-age2 plot. All 3 are acceptable.


Binned residual plot (Subset) - logit_md1
Average residual
0.04
0.02
0.00
-0.02
-0.04
10
20
30
40
50
60
70
age2


Binned residual plot (Subset) - logit_md3
Average residual
0.04
0.02
0.00
-0.02
-0.04
10
20
30
40
50
60
70
age2



For *categorical variable* we calculate mean residuals for each group. It should be around 0 and the difference between different group shouldn't be too large. We only take some for example.

From the results we can see that only when **cs** values 9 and 10 the residual is abnormal. And it's because the samples with **cs**>=9 are very little. Since the number of samples with "**cs**>=9" is very small, it will not influence our analysis so we just leave it.

```
forceDiag%>%
  group_by(gender)%>%
  summarise(mean_resid=mean(.deviance))
```

```
## # A tibble: 2 × 2
##   gender mean_resid
##   <fct>       <dbl>
## 1 0          -0.161
## 2 1          -0.192
```

```
forceDiag%>%
  group_by(cs)%>%
  summarise(mean_resid=mean(.deviance))
```

```
## # A tibble: 11 × 2
##       cs mean_resid
##    <dbl>      <dbl>
##  1     0    -0.186
##  2     1    -0.180
##  3     2    -0.143
##  4     3    -0.146
##  5     4    -0.111
##  6     5    -0.106
##  7     6    -0.0913
##  8     7    -0.167
```

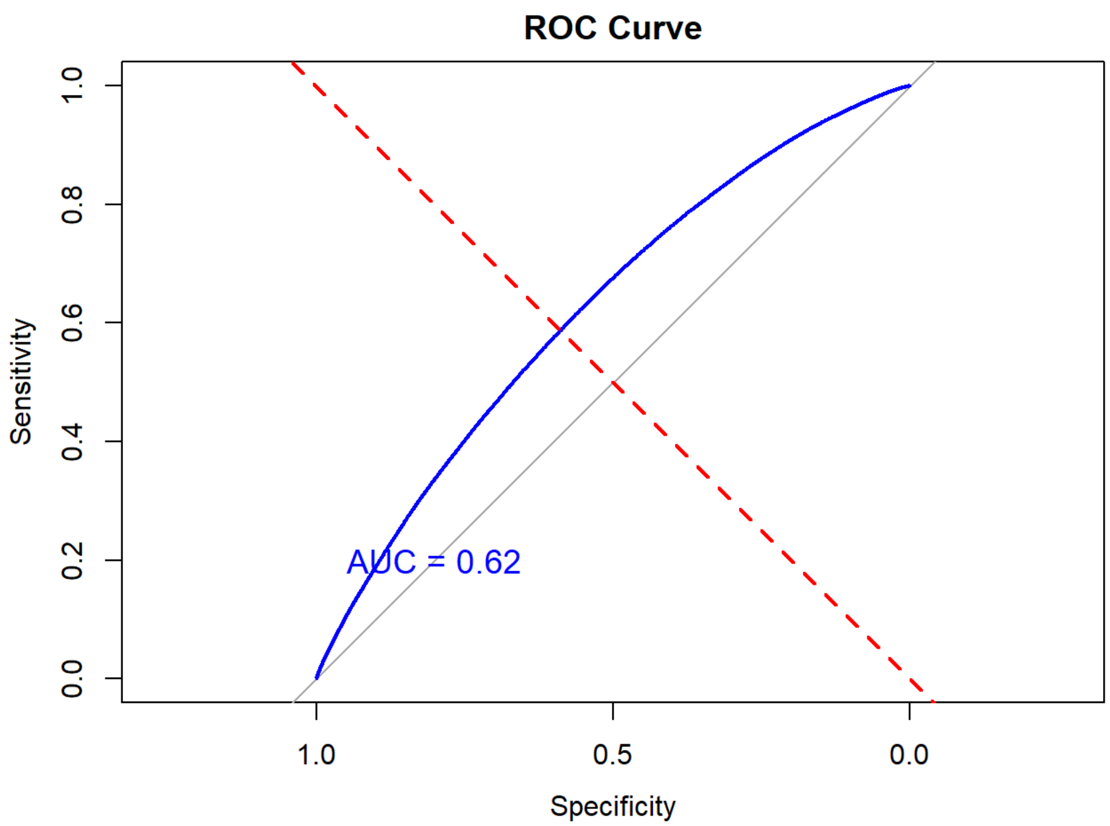
```
## 9       8      -0.0800
## 10      9       0.715
## 11     10      -1.55
```

2.3 Model assessment

2.3.1 ROC-curve

A ROC curve is constructed by plotting the true positive rate (TPR) against the false positive rate (FPR). It provides a visual representation of a model's ability to distinguish between two classes. The AUC measures the overall performance of the classification model. A perfect model has an AUC of 1, while a random model has an AUC of 0.5. The higher the AUC, the better the model's discriminatory power.

We only show ROC curve for one model but actually we've calculate the AUC for all 4 models. The AUC for 4 models are all 0.63, which means **there is no clear advantage of one model over the others in terms of Sensitivity and Specificity**.



2.3.2 Cross validation

We also use Cross Validation to evaluate 4 models' performances.

From the results of cross validation, we can see that the prediction accuracy for these models are basically the same, while *logit_md1* has the simplest form. In conclusion, **logit_md1** should be the final model we choose.

```
testCV <- function(form, data, B = 1, k = 5) {
  n <- nrow(data)
  PEcv <- vector("list", B)
  squared_devaince <- numeric(n)
  for(b in 1:B) {
```

```
    ## Generating the random division into groups
    group <- sample(rep(1:k, length.out = n))
    for(i in 1:k) {
      train_data=data[group != i, ]
      test_data=data[group == i, ]
      modelcv <- glm(form,family = binomial(link = "logit"), data = train_data)
      muhat <- predict(modelcv, newdata = test_data, type="response")
      squared_devaince[group == i]<-  -2*( test_data$force * log(muhat) + (1-
test_data$force) * log(1-muhat) )
    }
    PEcv[[b]] <- squared_devaince
  }
  mean(unlist(PEcv))
}
imputed_data$force<-as.integer(as.character(imputed_data$force))

deviance1<-testCV(logit_md1, imputed_data) #1.009966727
deviance2<-testCV(logit_md2, imputed_data) #1.009712631
deviance3<-testCV(logit_md3, imputed_data) #1.009773304
deviance4<-testCV(logit_md4, imputed_data) #1.009523049


deviance_df <- data.frame(VariableName = c("deviance1", "deviance2", "deviance3",
"deviance4"), Value = c(deviance1, deviance2, deviance3, deviance4))

print(deviance_df)
```

```
##   VariableName    Value
## 1    deviance1 1.009965
## 2    deviance2 1.009713
## 3    deviance3 1.009773
## 4    deviance4 1.009522
```

2.4 Conclusion and discussion

2.4.1 Conclusion from the model

From summary of *logit_md1*, we find that **all** variables included in the model are influencing factors for force use in New York City's stop, question, and frisk, among which **race2** and **gender** are the two most influential ones. From the coefficients we can see when all other variables remain the same, the *log-odds* is 0.47 larger for *black/hispanic* civilians than *white* civilians, and 0.54 larger for *male* civilians than *female* civilians.

Age reduces log-odds, while **cs** raises it. It implies that the police tend to use force on *younger* and *more suspicious* civilians.

What' more, if the stop occurs **indoors**, **in a high crime area**, **in a high crime time**, the civilian **refused to provide ID**, the officer was **in uniform**, or **other civilians were not stopped with the civilian**, the *log-odds* **decreases**. The first four results can be interpreted that the *police are more cautious* in corresponding situations. The last two results can be interpreted that the *civilians are more cautious* in corresponding situations. Both will lead to a lower probability of force use.

2.4.2 Discussion about the Dataset

From the discussion in the beginning, non-randomly collected data may suffer from selection bias, where the sample does not represent the population adequately. When the model based on the dataset is applied to an ordinary citizen(we don't know if he/she is stopped), it may not necessarily be accurate. This can lead to overemphasizing or neglecting specific types of observations in the model. To mitigate this bias, it's essential to have access to data on both stopped and not-stopped civilians, allowing for a more comprehensive analysis.

If we only cares about the force use on a civilian that we've already known **stopped**, this dataset does not bring any bias to the model establishment. But it could be better if the data in different variable groups (**gender** for example) is distributed more evenly, which results in a more explanatory variable.

Plus: Interval estimation

We derive 4 interval estimations for the coefficient for **age** and they are basically the same, which can also provide evidence that the model is good. Because if it’s not, there would be a difference between the intervals derived from parametric and nonparametric bootstrapping.

We do the interval estimation for the coefficient of *logit_md1*. We only take **age2** for example.

```
# The standard interval
confint.default(logit_md1,"age2")
```

```
##             2.5 %      97.5 %
## age2 -0.01104695 -0.01063882
```

```
# The likelihood interval
confint(logit_md1,"age2")
```

```
##       2.5 %      97.5 %
## -0.01104701 -0.01063888
```

```
# nonparametric bootstrap
B<- 5
n<- nrow(imputed_data)
beta<- numeric(B)
for(b in 1:B){
  i<- sample(n,n,replace=TRUE)
  bootGlm<-glm(form1,family = binomial(link = "logit"), data = imputed_data[i,])
  beta[b]<-coefficients(bootGlm)["age2"]
}

# parametric bootstrap
parbeta<-numeric(B)
d2Samp<-imputed_data
for(b in 1:B){
```

```r
  d2Samp$force<-simulate(logit_md1)[,1]
  bootGlm<-glm(form1,family = binomial(link = "logit"), data = d2Samp)
  parbeta[b]<-coefficients(bootGlm)["age2"]
}

sebeta<-sd(beta)
separbeta<-sd(parbeta)
# standard error based on ordinary analytic approximations
betahat<-coefficients(logit_md1)["age2"]

# interval based on nonparametric bootstrap
betahat+1.96*sebeta*c(-1,1)
```

```
## [1] -0.01118211 -0.01050366
```

```r
# interval based on parametric bootstrap
betahat+1.96*separbeta*c(-1,1)
```

```
## [1] -0.01110221 -0.01058356
```