Predicting Food Safety Violations in Toronto Restaurants

David Horan

November 24, 2016

Step 1: Load the data from the DineSafe inspection database. The data was sourced in XML format, and needed to be converted to a dataframe.

```
## Create a dataframe from the XML data.
library("XML")
```

```
## Warning: package 'XML' was built under R version 3.3.1
```

```
dinesafe.tmp <- "C:/Users/Jenn & Dave/Documents/Capstone/dinesafe.Sept17.xml"
dinesafe <- xmlTreeParse(dinesafe.tmp)
class(dinesafe)
```

```
## [1] "XMLDocument"         "XMLAbstractDocument"
```

```
xmltop <- xmlRoot(dinesafe)
inspections <- xmlSApply(xmltop, function(x) xmlSApply(x, xmlValue))
inspect_df <- data.frame(t(inspections),row.names=NULL)
```

Step 2: Download an address point directory so that street addresses of the establishments can be associated with specific areas of Toronto.

```
library("foreign")
Addresses <- read.dbf("C:/Users/Jenn & Dave/Documents/Capstone/ADDRESS_POINT_WGS84.dbf
")

## Create a complete street address that can be used to merge with the inspections dat
a.
Addresses$Num_St <- paste(Addresses$ADDRESS, Addresses$LFNAME, sep = " ")
```

Step 3: Label the columns of the inspection file and set them to the appropriate data types.

```
## Organize inspection data.
colnames(inspect_df)
```

```
##  [1] "ROW_ID"                      "ESTABLISHMENT_ID"
##  [3] "INSPECTION_ID"               "ESTABLISHMENT_NAME"
##  [5] "ESTABLISHMENTTYPE"           "ESTABLISHMENT_ADDRESS"
##  [7] "ESTABLISHMENT_STATUS"        "MINIMUM_INSPECTIONS_PERYEAR"
##  [9] "INFRACTION_DETAILS"          "INSPECTION_DATE"
## [11] "SEVERITY"                    "ACTION"
## [13] "COURT_OUTCOME"               "AMOUNT_FINED"
```

```
inspect_df$ROW_ID <- as.numeric(inspect_df$ROW_ID)
inspect_df$ESTABLISHMENT_ID <- as.numeric(inspect_df$ESTABLISHMENT_ID)
inspect_df$ESTABLISHMENT_ID <- as.factor(inspect_df$ESTABLISHMENT_ID)
inspect_df$INSPECTION_ID <- as.numeric(inspect_df$INSPECTION_ID)
inspect_df$ESTABLISHMENT_NAME <- as.character(inspect_df$ESTABLISHMENT_NAME)
inspect_df$ESTABLISHMENT_NAME <- toupper(inspect_df$ESTABLISHMENT_NAME)
inspect_df$ESTABLISHMENTTYPE <- as.character(inspect_df$ESTABLISHMENTTYPE)
inspect_df$ESTABLISHMENTTYPE <- as.factor(inspect_df$ESTABLISHMENTTYPE)
inspect_df$ESTABLISHMENT_ADDRESS <- as.character(inspect_df$ESTABLISHMENT_ADDRESS)
inspect_df$ESTABLISHMENT_ADDRESS <- as.factor(inspect_df$ESTABLISHMENT_ADDRESS)
inspect_df$ESTABLISHMENT_STATUS <- as.character(inspect_df$ESTABLISHMENT_STATUS)
inspect_df$ESTABLISHMENT_STATUS <- as.factor(inspect_df$ESTABLISHMENT_STATUS)
inspect_df$MINIMUM_INSPECTIONS_PERYEAR <- as.numeric(inspect_df$MINIMUM_INSPECTIONS_PE
RYEAR)
inspect_df$MINIMUM_INSPECTIONS_PERYEAR <- as.factor(inspect_df$MINIMUM_INSPECTIONS_PER
YEAR)
inspect_df$INFRACTION_DETAILS <- as.character(inspect_df$INFRACTION_DETAILS)
inspect_df$INFRACTION_DETAILS <- as.factor(inspect_df$INFRACTION_DETAILS)
inspect_df$INSPECTION_DATE <- as.character(inspect_df$INSPECTION_DATE)
inspect_df$INSPECTION_DATE <- as.numeric(as.POSIXct(inspect_df$INSPECTION_DATE))
inspect_df$SEVERITY <- as.character(inspect_df$SEVERITY)
inspect_df$SEVERITY <- as.factor(inspect_df$SEVERITY)
inspect_df$ACTION <- as.character(inspect_df$ACTION)
inspect_df$ACTION <- as.factor(inspect_df$ACTION)
inspect_df$COURT_OUTCOME <- as.character(inspect_df$COURT_OUTCOME)
inspect_df$COURT_OUTCOME <- as.factor(inspect_df$COURT_OUTCOME)
inspect_df$AMOUNT_FINED <- as.numeric(inspect_df$AMOUNT_FINED)

## View a sample of the data.
head(inspect_df)
```

```
##   ROW_ID ESTABLISHMENT_ID INSPECTION_ID          ESTABLISHMENT_NAME
## 1      1          1222579     103329697 SAI-LILA KHAMAN DHOKLA HOUSE
## 2      2          1222579     103329697 SAI-LILA KHAMAN DHOKLA HOUSE
## 3      3          1222579     103329697 SAI-LILA KHAMAN DHOKLA HOUSE
## 4      4          1222579     103329697 SAI-LILA KHAMAN DHOKLA HOUSE
## 5      5          1222579     103329697 SAI-LILA KHAMAN DHOKLA HOUSE
## 6      6          1222579     103420091 SAI-LILA KHAMAN DHOKLA HOUSE
##   ESTABLISHMENTTYPE ESTABLISHMENT_ADDRESS ESTABLISHMENT_STATUS
## 1     Food Take Out        870 MARKHAM RD                 Pass
## 2     Food Take Out        870 MARKHAM RD                 Pass
## 3     Food Take Out        870 MARKHAM RD                 Pass
## 4     Food Take Out        870 MARKHAM RD                 Pass
## 5     Food Take Out        870 MARKHAM RD                 Pass
## 6     Food Take Out        870 MARKHAM RD                 Pass
##   MINIMUM_INSPECTIONS_PERYEAR
## 1                           2
## 2                           2
## 3                           2
## 4                           2
## 5                           2
## 6                           2
##                                                  INFRACTION_DETAILS
## 1 FAIL TO PROVIDE TOWELS IN FOOD PREPARATION AREA O. REG  562/90 SEC. 20(1)(C)
## 2                                 Operator fail to properly maintain rooms
## 3                                 Operator fail to properly wash equipment
## 4                           Operator fail to properly wash surfaces in rooms
## 5                   Operator fail to sanitize garbage containers as required
## 6                                 Operator fail to properly wash equipment
##   INSPECTION_DATE        SEVERITY                     ACTION
## 1      1410235200 S - Significant Corrected During Inspection
## 2      1410235200       M - Minor            Notice to Comply
## 3      1410235200       M - Minor            Notice to Comply
## 4      1410235200       M - Minor            Notice to Comply
## 5      1410235200       M - Minor            Notice to Comply
## 6      1420693200       M - Minor            Notice to Comply
##   COURT_OUTCOME AMOUNT_FINED
## 1  character(0)           NA
## 2  character(0)           NA
## 3  character(0)           NA
## 4  character(0)           NA
## 5  character(0)           NA
## 6  character(0)           NA
```

Step 4: Create a list of all unique inspection IDs.

```
sub_insp <- subset(inspect_df,select = c(ESTABLISHMENT_ID:MINIMUM_INSPECTIONS_PERYEAR,
INSPECTION_DATE))
inspect_unique <- unique(sub_insp)

head(inspect_unique)
```

```
##    ESTABLISHMENT_ID INSPECTION_ID           ESTABLISHMENT_NAME
## 1           1222579     103329697 SAI-LILA KHAMAN DHOKLA HOUSE
## 6           1222579     103420091 SAI-LILA KHAMAN DHOKLA HOUSE
## 9           1222580     103490157  OYINGBO AFRICAN SUPERMARKET
## 10          1222580     103601595  OYINGBO AFRICAN SUPERMARKET
## 11          1222807     103355310                   PHO BO TO
## 12          1222807     103472815                   PHO BO TO
##    ESTABLISHMENTTYPE ESTABLISHMENT_ADDRESS ESTABLISHMENT_STATUS
## 1      Food Take Out        870 MARKHAM RD                 Pass
## 6      Food Take Out        870 MARKHAM RD                 Pass
## 9        Supermarket          1550 JANE ST                 Pass
## 10       Supermarket          1550 JANE ST                 Pass
## 11        Restaurant   1635 LAWRENCE AVE W                 Pass
## 12        Restaurant   1635 LAWRENCE AVE W                 Pass
##    MINIMUM_INSPECTIONS_PERYEAR INSPECTION_DATE
## 1                            2      1410235200
## 6                            2      1420693200
## 9                            1      1431403200
## 10                           1      1446440400
## 11                           3      1415163600
## 12                           3      1429761600
```

Step 5: Create a list of inspection IDs that resulted in either Significant (S) or Crucial (C) severity.

```
## Flag each of the unique inspections based on whether it resulted in a Significant o
r Crucial violation.
table(inspect_df$SEVERITY)
```

```
##
##         C - Crucial        character(0)           M - Minor
##                2114               30617               31394
## NA - Not Applicable     S - Significant
##                3693               20341
```

```
inspect_SevCru <- inspect_df[inspect_df$SEVERITY %in% c("C - Crucial","S - Significant
"),3]

## Start the timer for the following loop.
ptm <- proc.time()

Sev_Cru <- vector()
for (i in 1:length(inspect_unique$INSPECTION_ID)){
  if (inspect_unique$INSPECTION_ID[i] %in% inspect_SevCru){
    Sev_Cru[i] <- T}
  else{
    Sev_Cru[i] <- F }}

## Stop the timer
proc.time() - ptm
```

```
##    user  system elapsed
##    9.33    0.43    9.99
```

```
## Bind the unique inspection records with the Severity indicator.
inspect_work1 <- as.data.frame(cbind(inspect_unique, Sev_Cru))

head(inspect_work1)
```

```
##    ESTABLISHMENT_ID INSPECTION_ID           ESTABLISHMENT_NAME
## 1           1222579     103329697 SAI-LILA KHAMAN DHOKLA HOUSE
## 6           1222579     103420091 SAI-LILA KHAMAN DHOKLA HOUSE
## 9           1222580     103490157  OYINGBO AFRICAN SUPERMARKET
## 10          1222580     103601595  OYINGBO AFRICAN SUPERMARKET
## 11          1222807     103355310                   PHO BO TO
## 12          1222807     103472815                   PHO BO TO
##    ESTABLISHMENTTYPE ESTABLISHMENT_ADDRESS ESTABLISHMENT_STATUS
## 1      Food Take Out        870 MARKHAM RD                 Pass
## 6      Food Take Out        870 MARKHAM RD                 Pass
## 9        Supermarket          1550 JANE ST                 Pass
## 10       Supermarket          1550 JANE ST                 Pass
## 11        Restaurant   1635 LAWRENCE AVE W                 Pass
## 12        Restaurant   1635 LAWRENCE AVE W                 Pass
##    MINIMUM_INSPECTIONS_PERYEAR INSPECTION_DATE Sev_Cru
## 1                            2      1410235200    TRUE
## 6                            2      1420693200    TRUE
## 9                            1      1431403200   FALSE
## 10                           1      1446440400   FALSE
## 11                           3      1415163600   FALSE
## 12                           3      1429761600   FALSE
```

NOTE: The loop was timed to measure the efficiency of this procedure, and the feasility of applying it to a larger dataset.

Step 6: Explore the dataset (univariate analysis) to understand more about the variables.

```
head(sort(table(inspect_work1$ESTABLISHMENTTYPE),decreasing = TRUE),n=10)
```

```
##
##                          Restaurant                         Food Take Out
##                               27816                                  8369
## Food Store (Convenience / Variety)                     Food Court Vendor
##                                3360                                  2080
##                          Supermarket                  Child Care - Catered
##                                1818                                  1811
##      Child Care - Food Preparation                               Bakery
##                                1596                                  1448
##                         Butcher Shop                Food Processing Plant
##                                 620                                   591
```

```
head(sort(table(inspect_work1$ESTABLISHMENT_NAME), decreasing = TRUE),n=20)
```

```
##
##        TIM HORTONS             SUBWAY        PIZZA PIZZA
##                939                859                370
##         MCDONALD'S         PIZZA NOVA   STARBUCKS COFFEE
##                301                181                178
##       TIM HORTON'S         SECOND CUP          STARBUCKS
##                173                152                137
##      BOOSTER JUICE                KFC              METRO
##                124                119                115
##       SWISS CHALET            FRESHII     DOMINO'S PIZZA
##                114                110                108
## AROMA ESPRESSO BAR            MR. SUB       THAI EXPRESS
##                106                102                101
##          PIZZA HUT SHOPPERS DRUG MART
##                 99                 95
```

```
sort(table(inspect_work1$MINIMUM_INSPECTIONS_PERYEAR), decreasing = TRUE)
```

```
##
##     2     3     1
## 29000 19361  7058
```

```
sort(table(inspect_work1$ESTABLISHMENT_STATUS), decreasing = TRUE)
```

```
##
##             Pass Conditional Pass           Closed
##            51481             3887               51
```

```
## Percentage of inspections resulting in a Significant or Crucial Health Violation.
nrow(inspect_work1[Sev_Cru == 1,])/nrow(inspect_work1)
```

```
## [1] 0.2190404
```

Step 7: Create dummy variables for the five largest establishment types, that can be used in the logistic regression.

```
TYPE_RESTAURANT <- as.factor(grepl("Restaurant", inspect_work1$ESTABLISHMENTTYPE))
length(which(TYPE_RESTAURANT == T))
```

```
## [1] 27816
```

```
TYPE_TAKEOUT <- as.factor(grepl("Food Take Out", inspect_work1$ESTABLISHMENTTYPE))
length(which(TYPE_TAKEOUT == T))
```

```
## [1] 8369
```

```
TYPE_FOODSTORE <- as.factor(grepl("Food Store", inspect_work1$ESTABLISHMENTTYPE))
length(which(TYPE_FOODSTORE == T))
```

```
## [1] 3360
```

```
TYPE_FOODCOURT <- as.factor(grepl("Food Court Vendor", inspect_work1$ESTABLISHMENTTYPE
))
length(which(TYPE_FOODCOURT == T))
```

```
## [1] 2080
```

```
TYPE_SUPERMARKET <- as.factor(grepl("Supermarket", inspect_work1$ESTABLISHMENTTYPE))
length(which(TYPE_SUPERMARKET == T))
```

```
## [1] 1818
```

```
inspect_work1 <- cbind(inspect_work1, TYPE_RESTAURANT, TYPE_TAKEOUT, TYPE_FOODSTORE, T
YPE_FOODCOURT, TYPE_SUPERMARKET)
```

Step 8: Since there are a large number of institutional establishments with low volumes, combine these into a single variable called “Institutions”.

```
INSTITUTIONS <- c("College/University Food services","Community Kitchen Meal Program",
"Elementary School Food services","Hospitals & Health Facilities","Institutional Food
Service","Nursing Home / Home for the Aged","Other Educational Facility Food Services"
,"Rest Home","Retirement Homes(Licensed)","Retirement Homes(Un-licensed)","School Nour
ishment Program","Secondary School Food Services","Serving Kitchen")

TYPE_INSTITUTION <- vector()
for (i in 1:length(inspect_work1$ESTABLISHMENTTYPE)){
  if (inspect_work1$ESTABLISHMENTTYPE[i] %in% INSTITUTIONS){
    TYPE_INSTITUTION[i] <- T}
  else{
    TYPE_INSTITUTION[i] <- F }}

TYPE_INSTITUTION <- as.factor(TYPE_INSTITUTION)

inspect_work1 <- cbind(inspect_work1,TYPE_INSTITUTION)

length(which(TYPE_INSTITUTION == T))
```

```
## [1] 2615
```

Step 9: add the municipality from the address file, joining on the address.

```
Addr_Muni <- subset(Addresses, select = c(Num_St,MUN_NAME))
Addr_Muni$Num_St <- toupper(Addr_Muni$Num_St)

inspect_work1 <- merge(x = inspect_work1, y = Addr_Muni, by.x = "ESTABLISHMENT_ADDRESS
", by.y = "Num_St")

MUN_ETOBICOKE <- as.factor(grepl("Etobicoke",inspect_work1$MUN_NAME))
MUN_EAST_YORK <- as.factor(grepl("East York",inspect_work1$MUN_NAME))
MUN_NORTH_YORK <- as.factor(grepl("North York", inspect_work1$MUN_NAME))
MUN_SCARBOROUGH <- as.factor(grepl("Scarborough", inspect_work1$MUN_NAME))
MUN_FMR_TORONTO <- as.factor(grepl("former Toronto", inspect_work1$MUN_NAME))
MUN_YORK <- as.factor(grepl("York", inspect_work1$MUN_NAME))

inspect_work1 <- cbind(inspect_work1, MUN_FMR_TORONTO, MUN_SCARBOROUGH, MUN_NORTH_YORK
, MUN_EAST_YORK, MUN_ETOBICOKE, MUN_YORK)

table(inspect_work1$MUN_NAME)
```

```
##
##      East York      Etobicoke former Toronto     North York    Scarborough
##           1195           5674          27950           8784           8334
##           York
##           1853
```

Step 10: Create dummy variables for the frequency of inspection.

```
INSP_1 <- as.factor(grepl("1",inspect_work1$MINIMUM_INSPECTIONS_PERYEAR))
INSP_2 <- as.factor(grepl("2",inspect_work1$MINIMUM_INSPECTIONS_PERYEAR))
INSP_3 <- as.factor(grepl("3",inspect_work1$MINIMUM_INSPECTIONS_PERYEAR))

inspect_work1 <- cbind(inspect_work1, INSP_1, INSP_2, INSP_3)
```

Step 11: Check for complete cases:

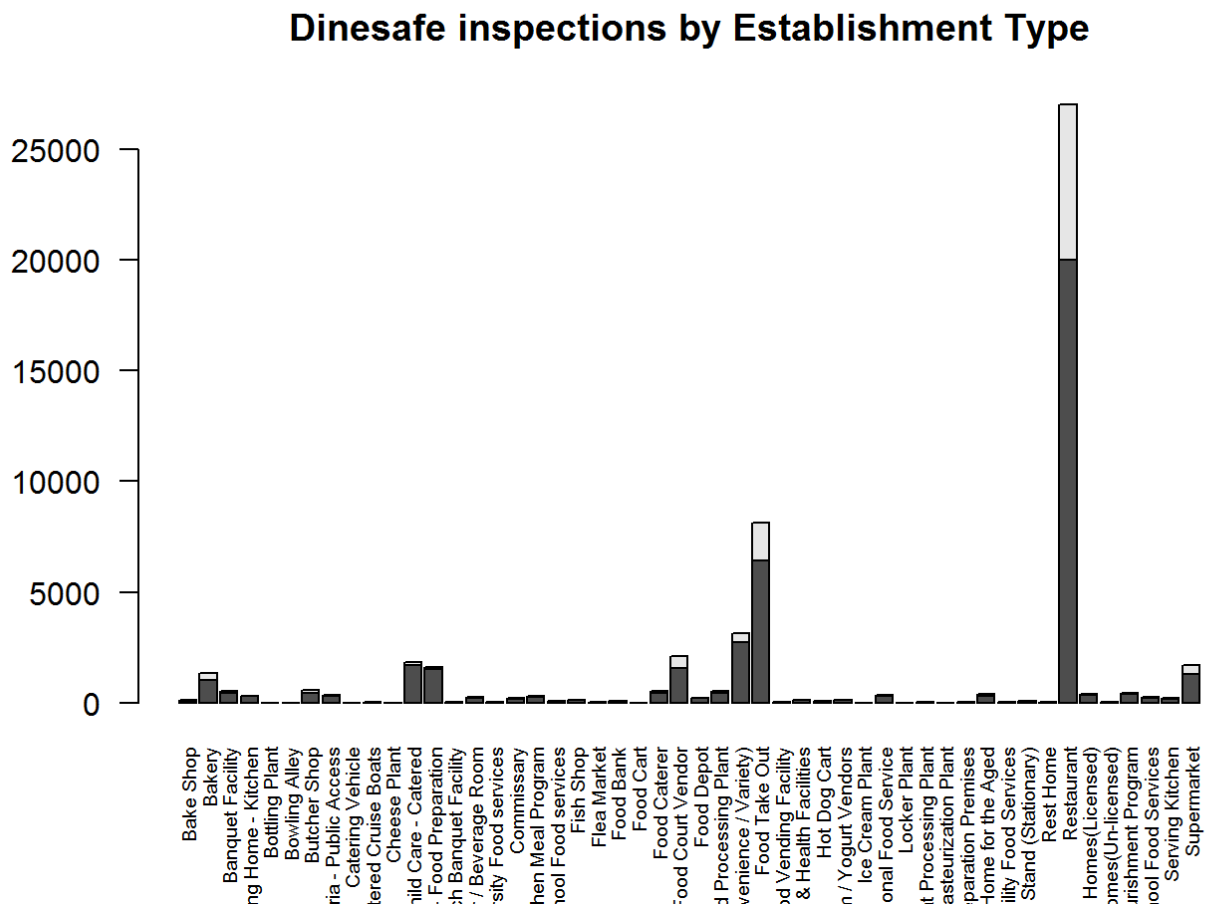
```
cc_test <- complete.cases(inspect_work1)
length(which(cc_test == F))
```

```
## [1] 0
```

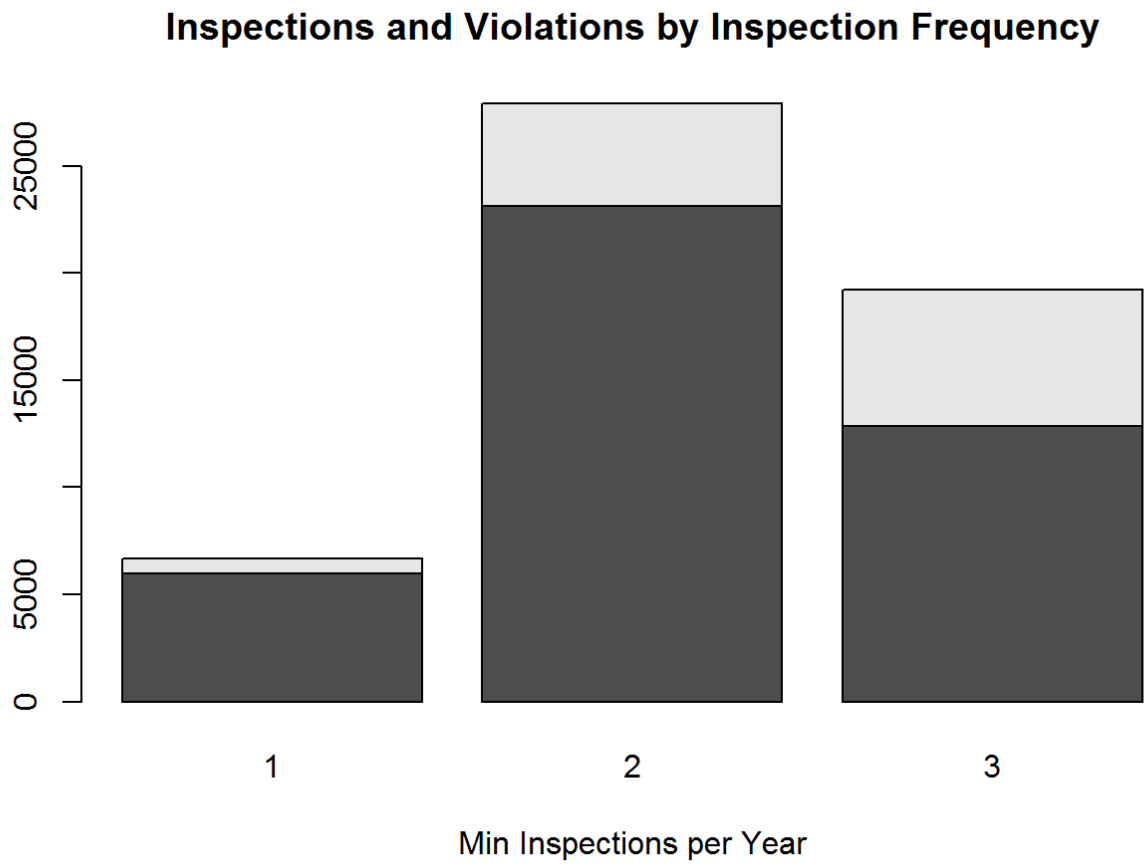
Result: It appears that there are no incomplete cases in the remaining data.

Step 12: Next, we will visualize the percentage of significant / crucial violations across the following categorical variables (Bivariate Analysis):

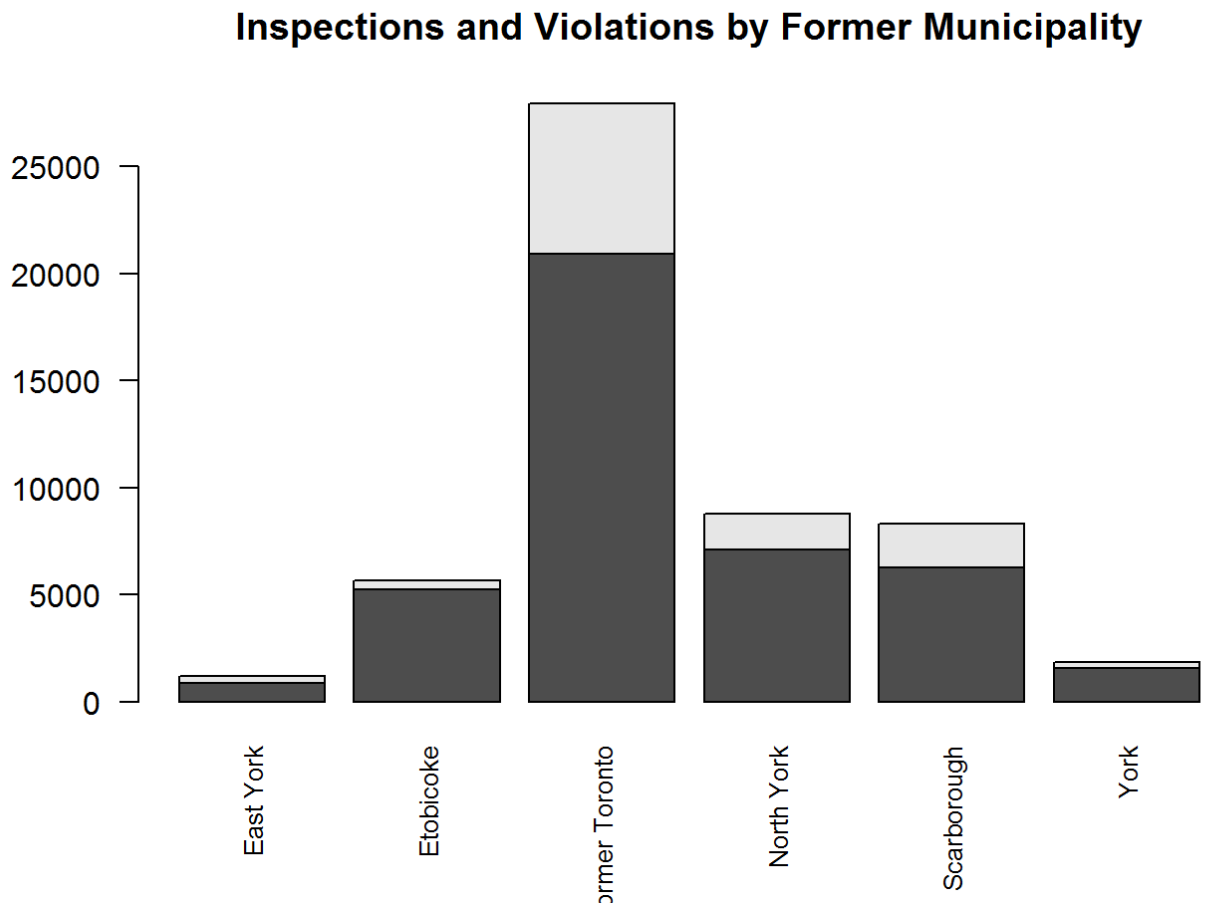
```
## (i) Establishment Type
barplot(table(inspect_work1$Sev_Cru,inspect_work1$ESTABLISHMENTTYPE), las=2, cex.names
 = 0.6)
title(main = "Dinesafe inspections by Establishment Type")
```




```
## (ii) Minimum Inspections per Year
barplot(table(inspect_work1$Sev_Cru,inspect_work1$MINIMUM_INSPECTIONS_PERYEAR))
title(main = "Inspections and Violations by Inspection Frequency", xlab = "Min Inspect
ions per Year")
```




```
## (iii) Former municipality (within current City of Toronto)
barplot(table(inspect_work1$Sev_Cru,inspect_work1$MUN_NAME), las=2, cex.names = 0.8)
title(main = "Inspections and Violations by Former Municipality")
```



Step 13: Use glmulti to choose the best fitted model for the analysis.

```
library("glmulti")
```

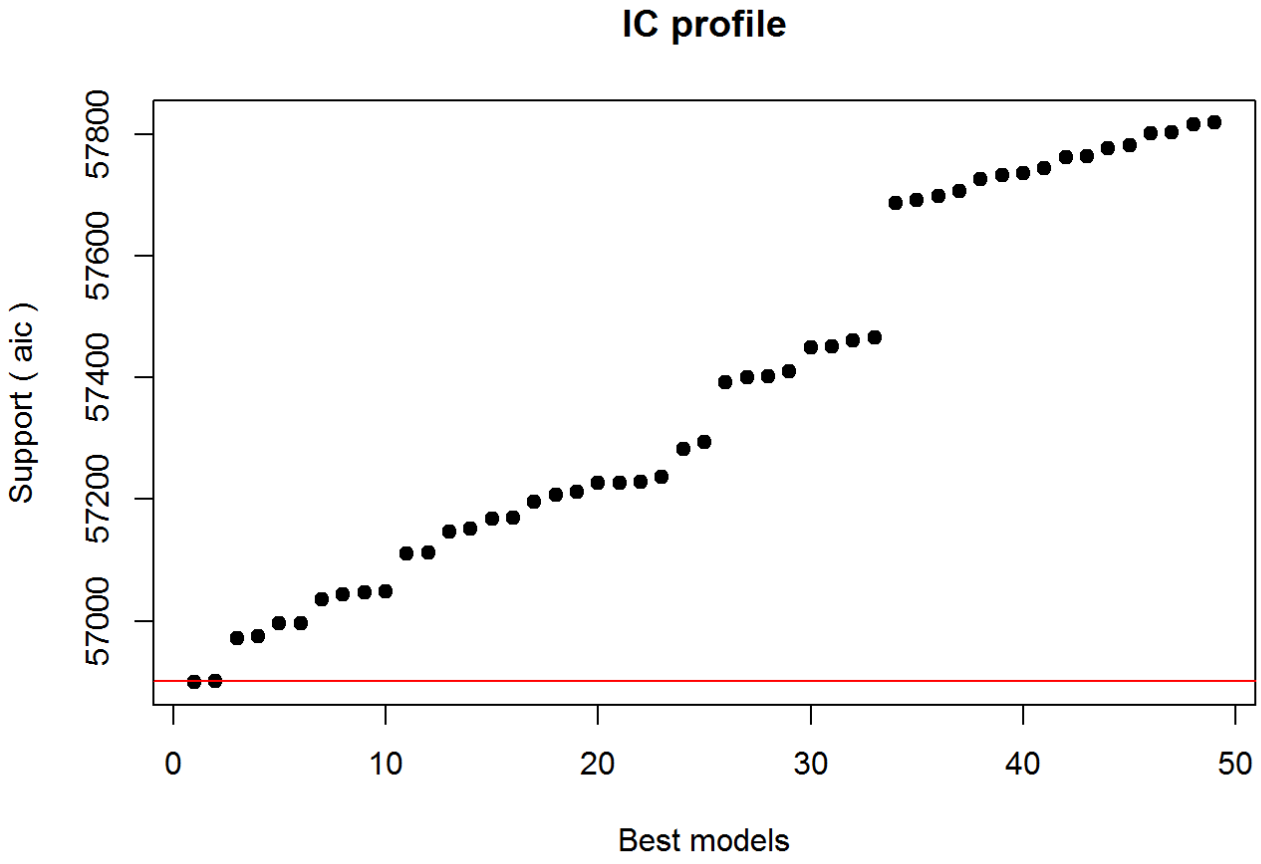
```
## Warning: package 'glmulti' was built under R version 3.3.2
```

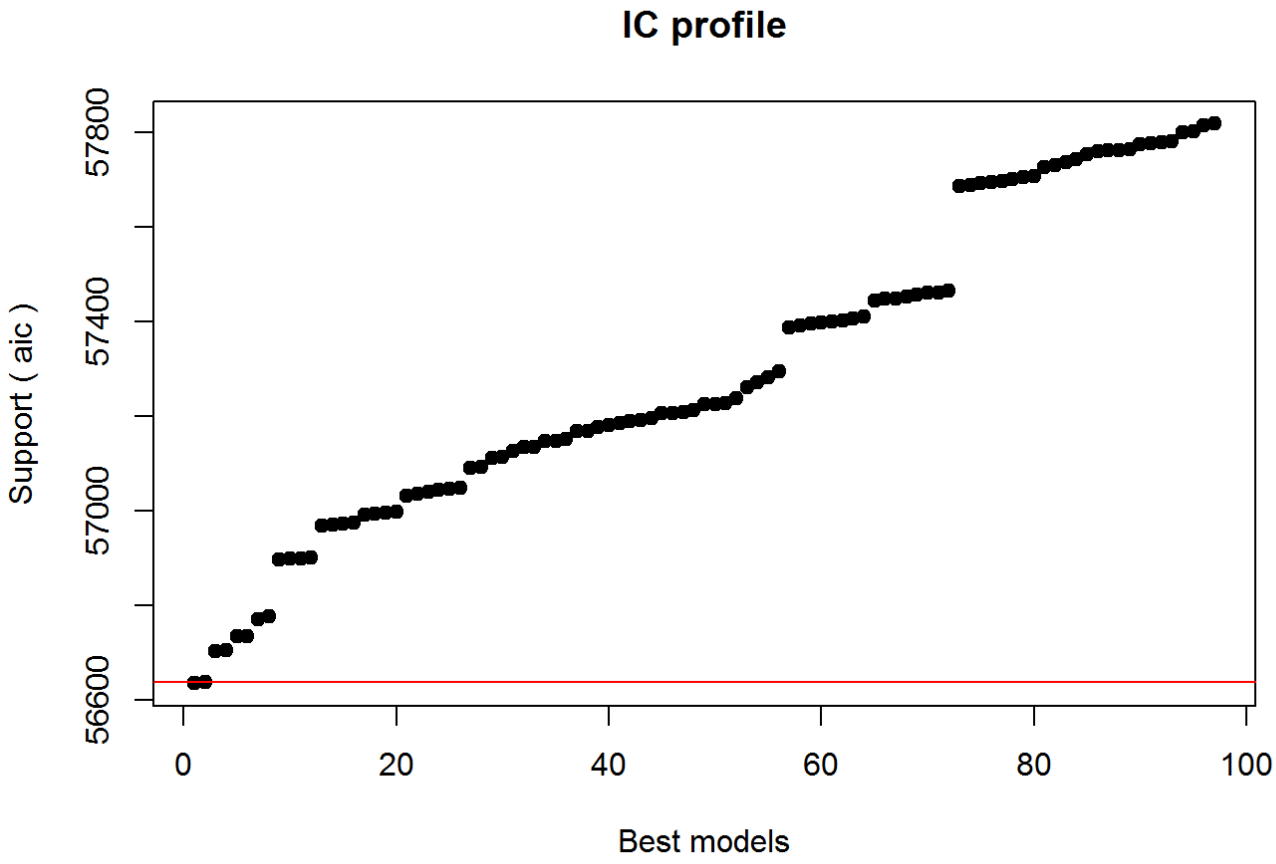
```
## Loading required package: rJava
```

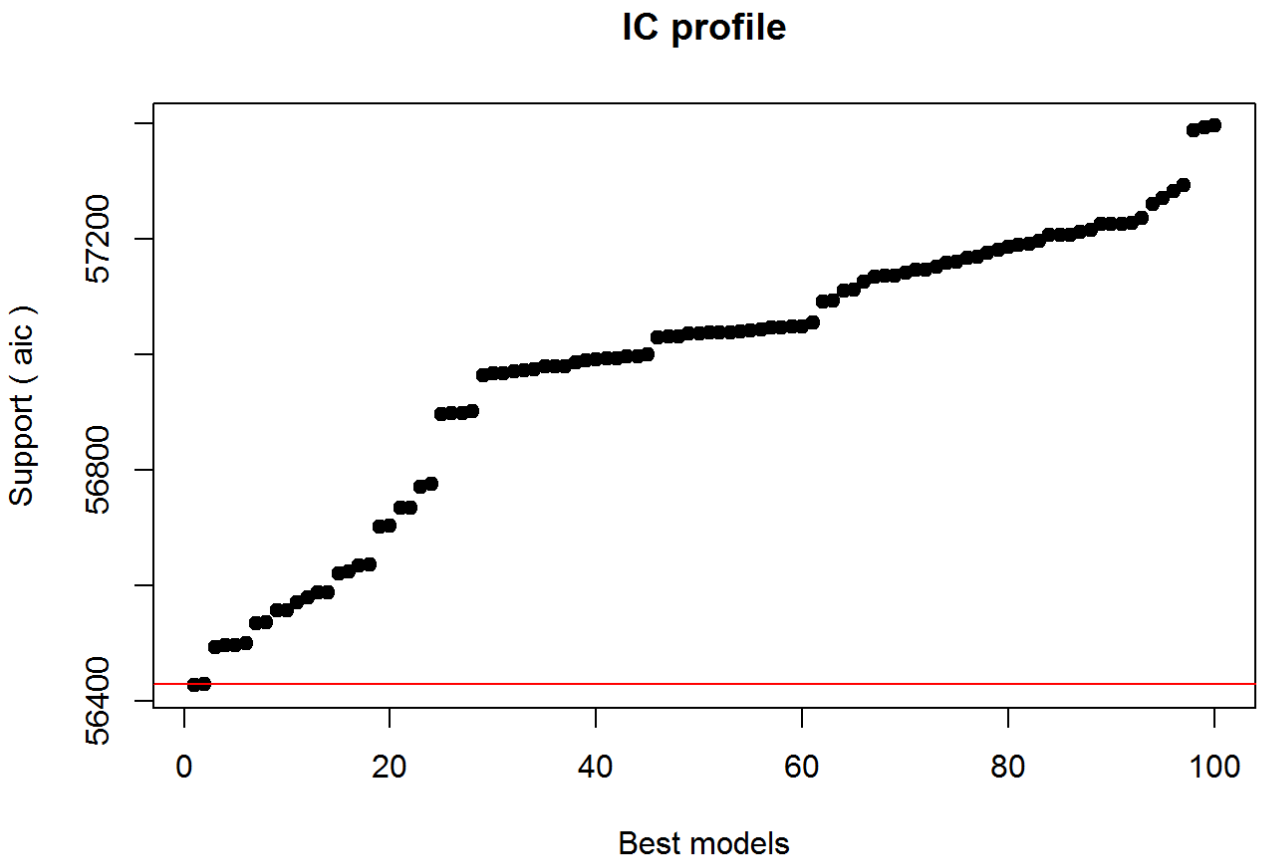
```
inspect_work1$Sev_Cru <- as.logical(inspect_work1$Sev_Cru)

glm_insp <- glm(Sev_Cru~TYPE_RESTAURANT+TYPE_INSTITUTION+TYPE_TAKEOUT+TYPE_FOODCOURT+M
UN_FMR_TORONTO+MUN_SCARBOROUGH+MUN_NORTH_YORK+MUN_ETOBICOKE+INSP_2+INSP_3, data = insp
ect_work1, family = binomial("logit"))
```

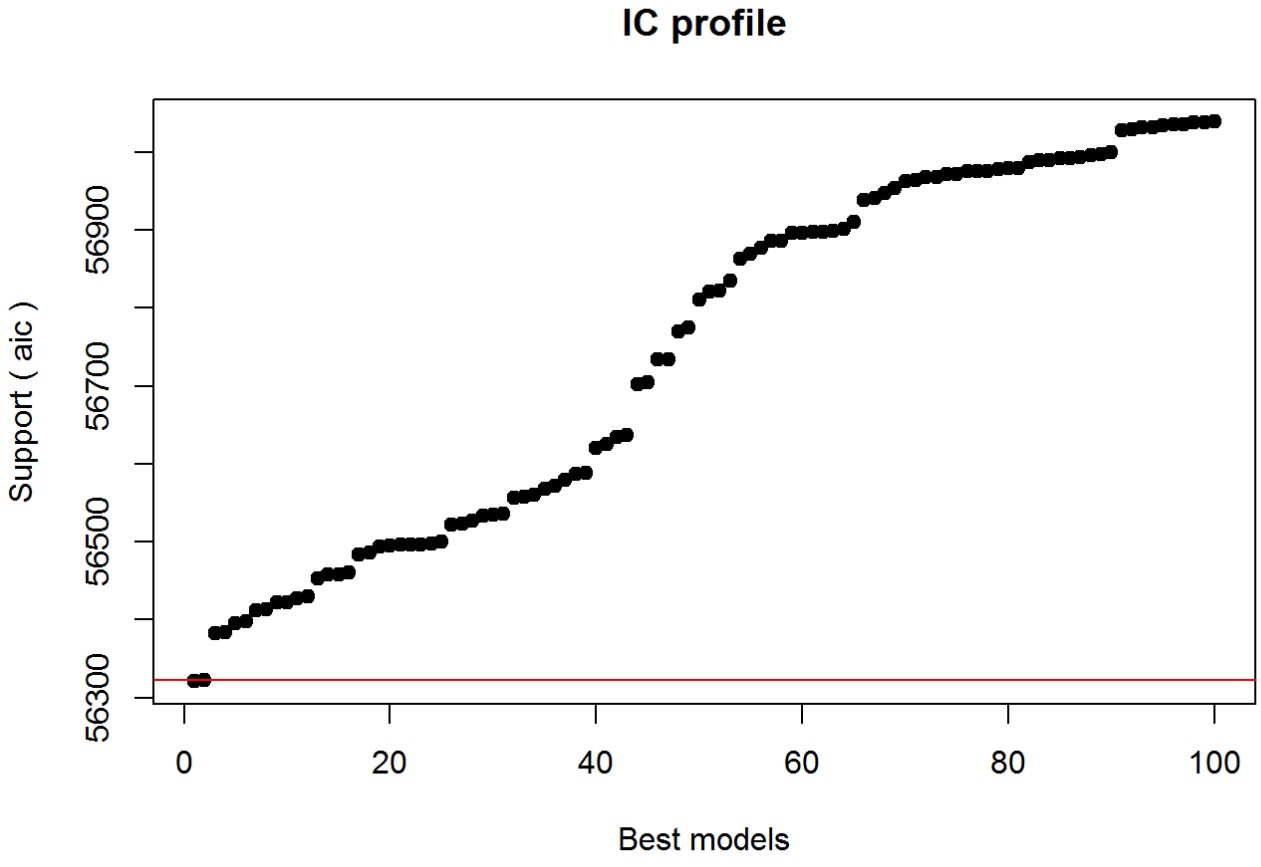
Step 13a: Because glmulti is iterative, I will hide about 20 pages of output.

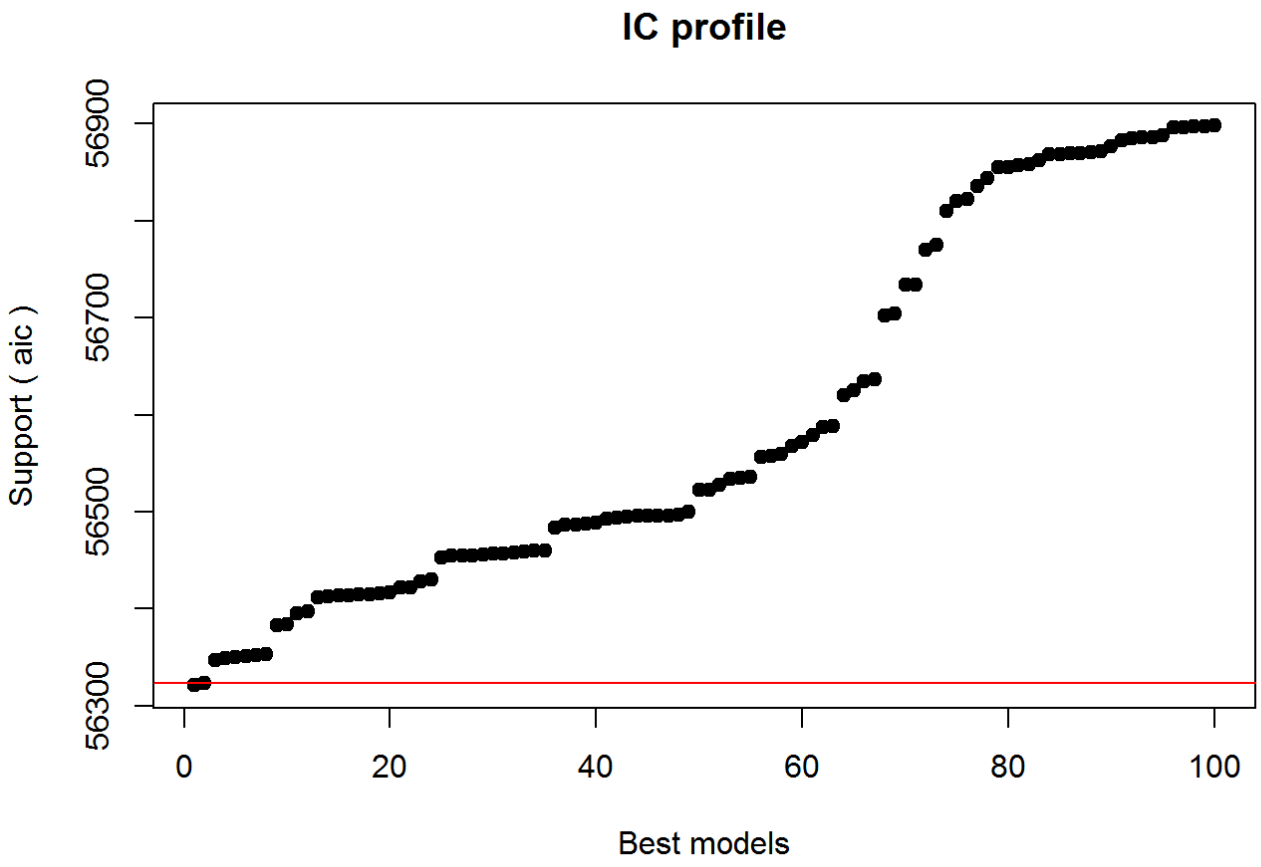
```
best.model <- glmulti(glm_insp, level = 1)
```

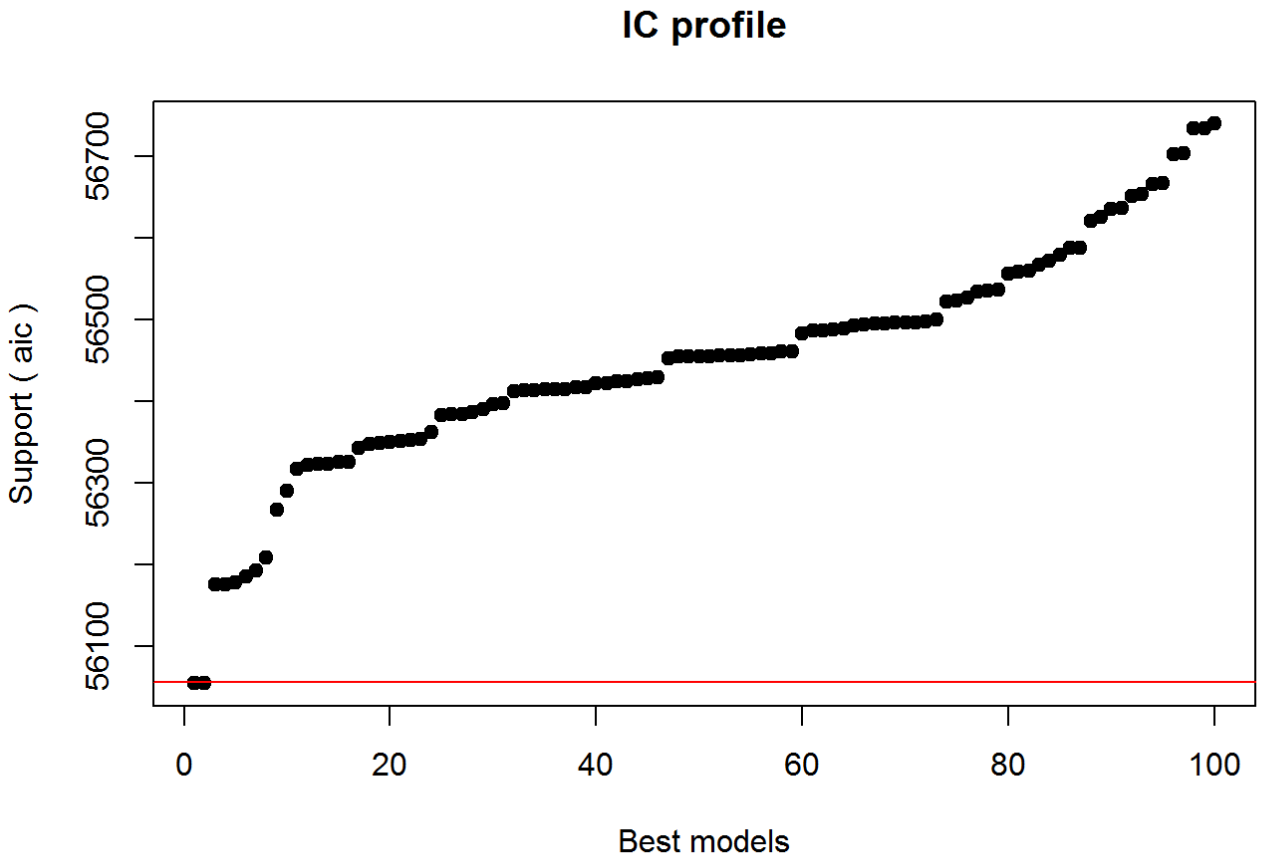

IC profile
Support (aic)
57000
57200
57400
57600
57800
0
10
20
30
40
50
Best models

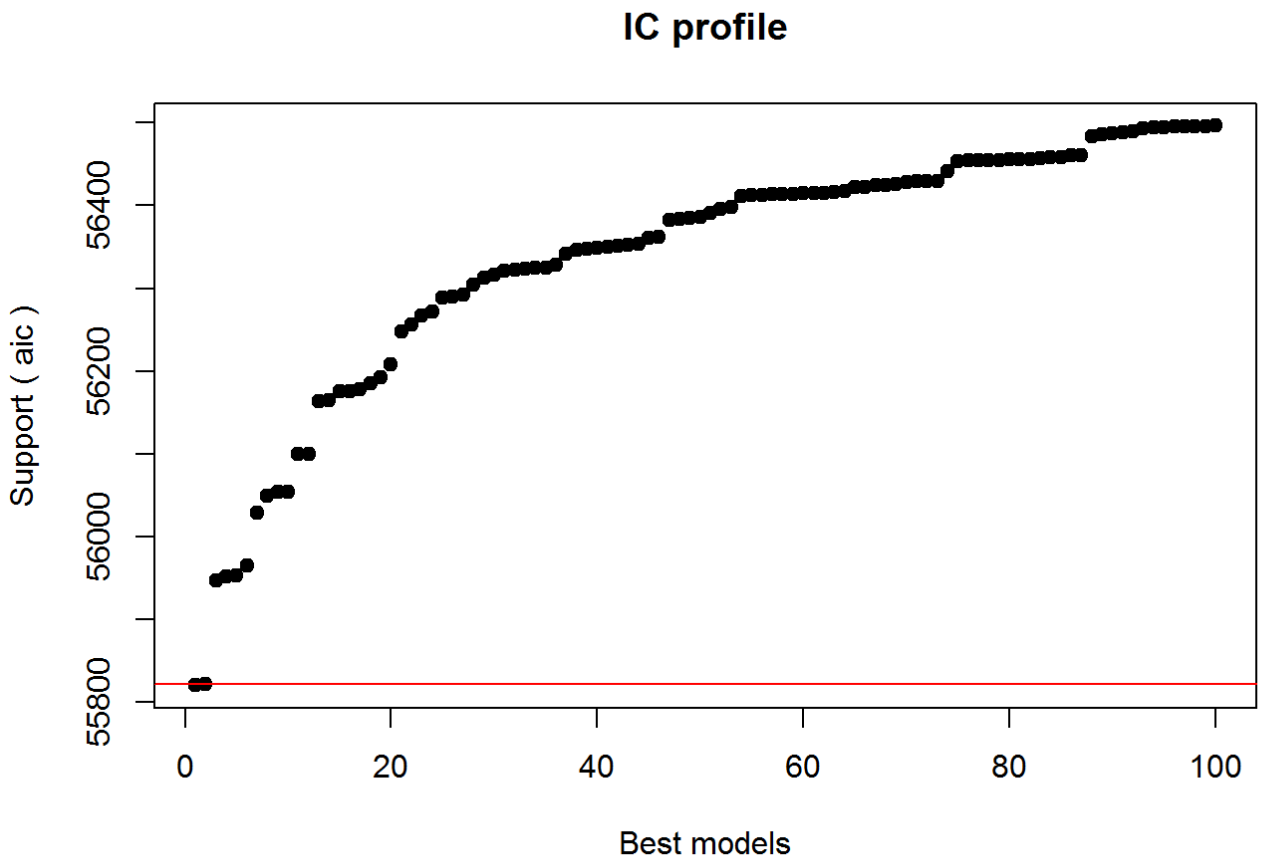

IC profile
Support (aic)
56600
57000
57400
57800
0
20
40
60
80
100
Best models

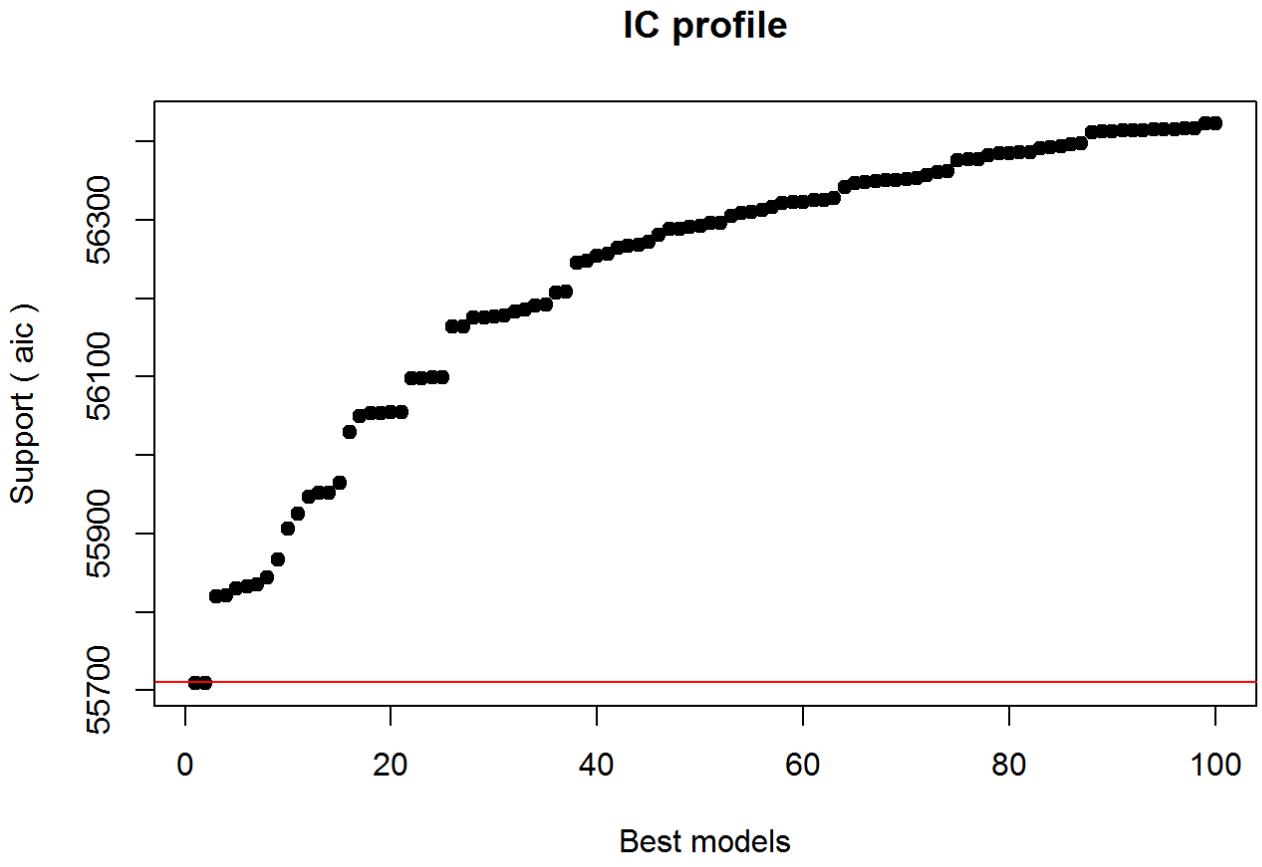

IC profile
Support (aic)
56400
56800
57200
0
20
40
60
80
100
Best models

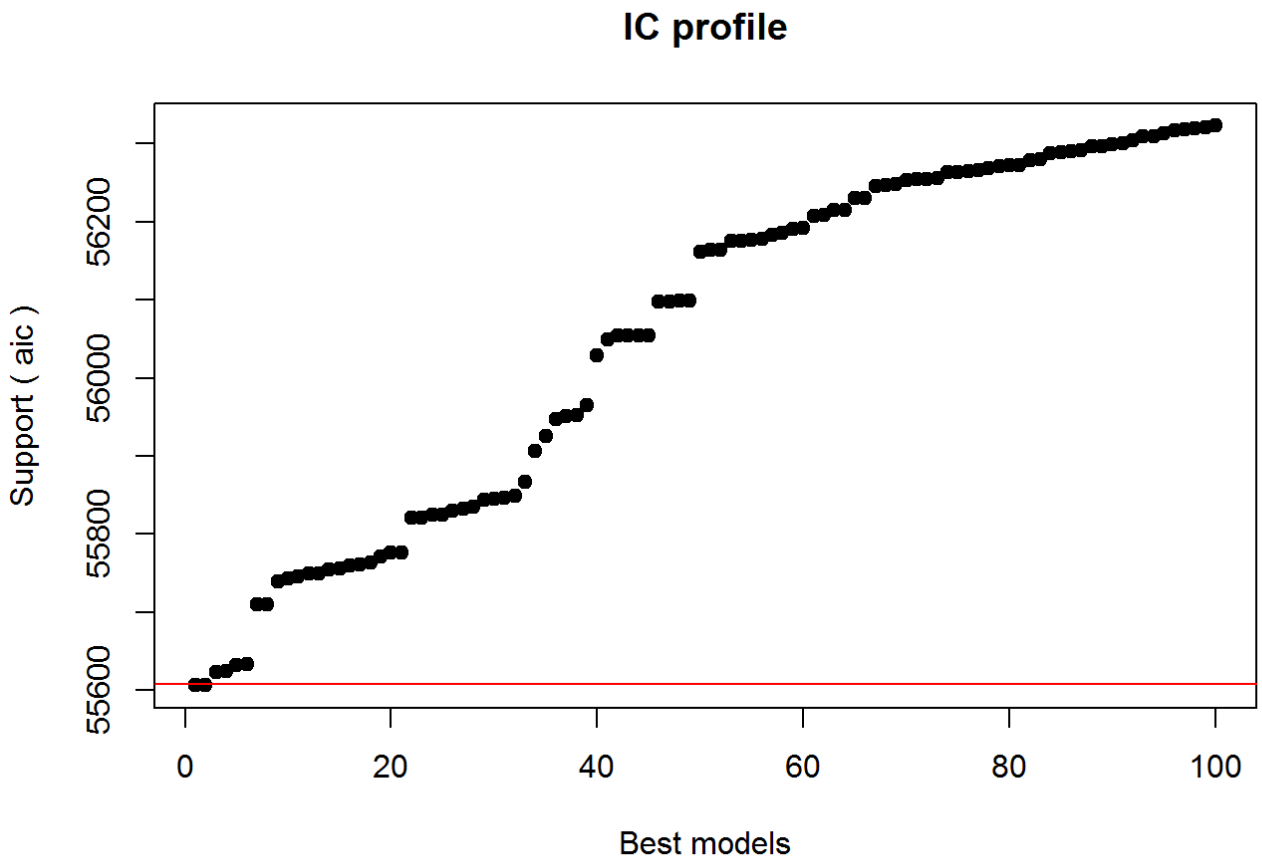

IC profile
Support (aic)
56300
56500
56700
56900
0
20
40
60
80
100
Best models

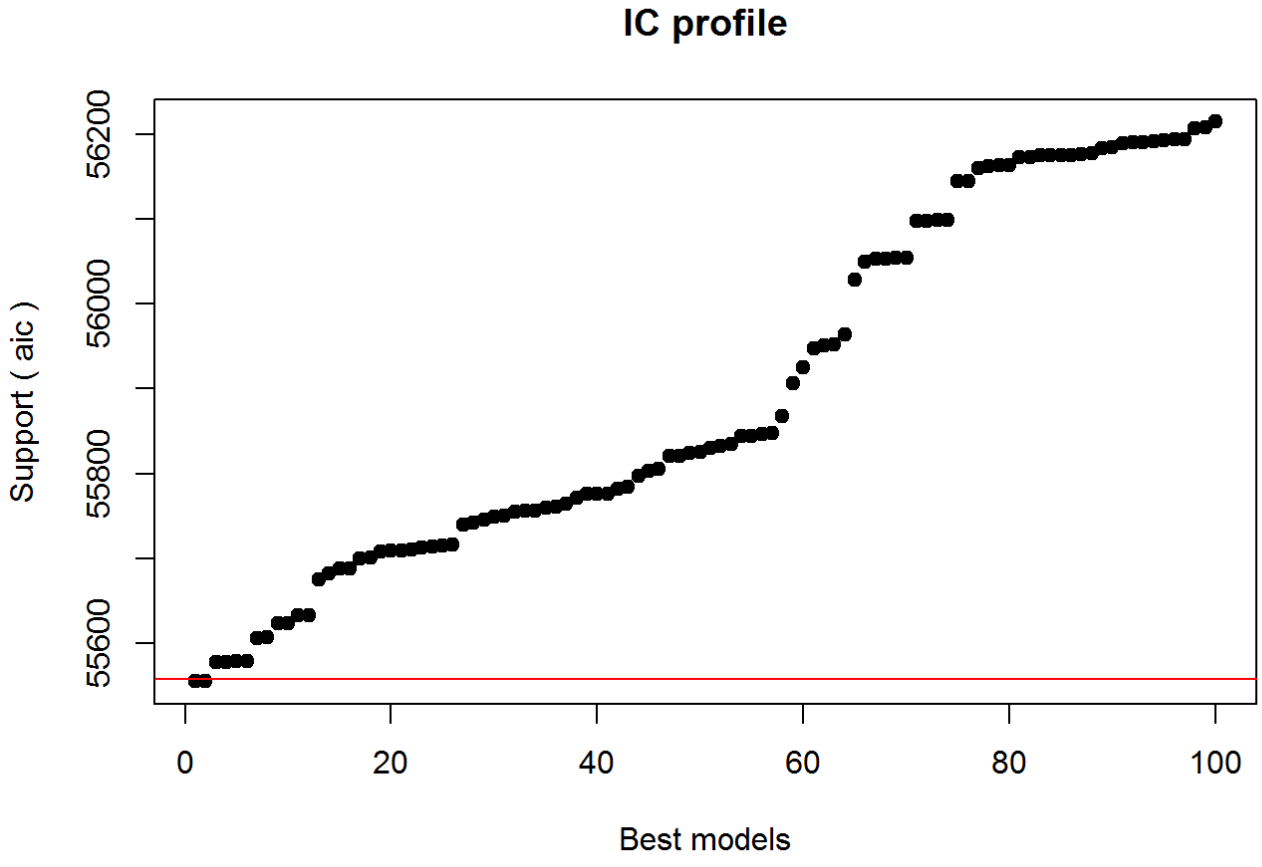

IC profile
Support (aic)
56300
56500
56700
56900
0
20
40
60
80
100
Best models

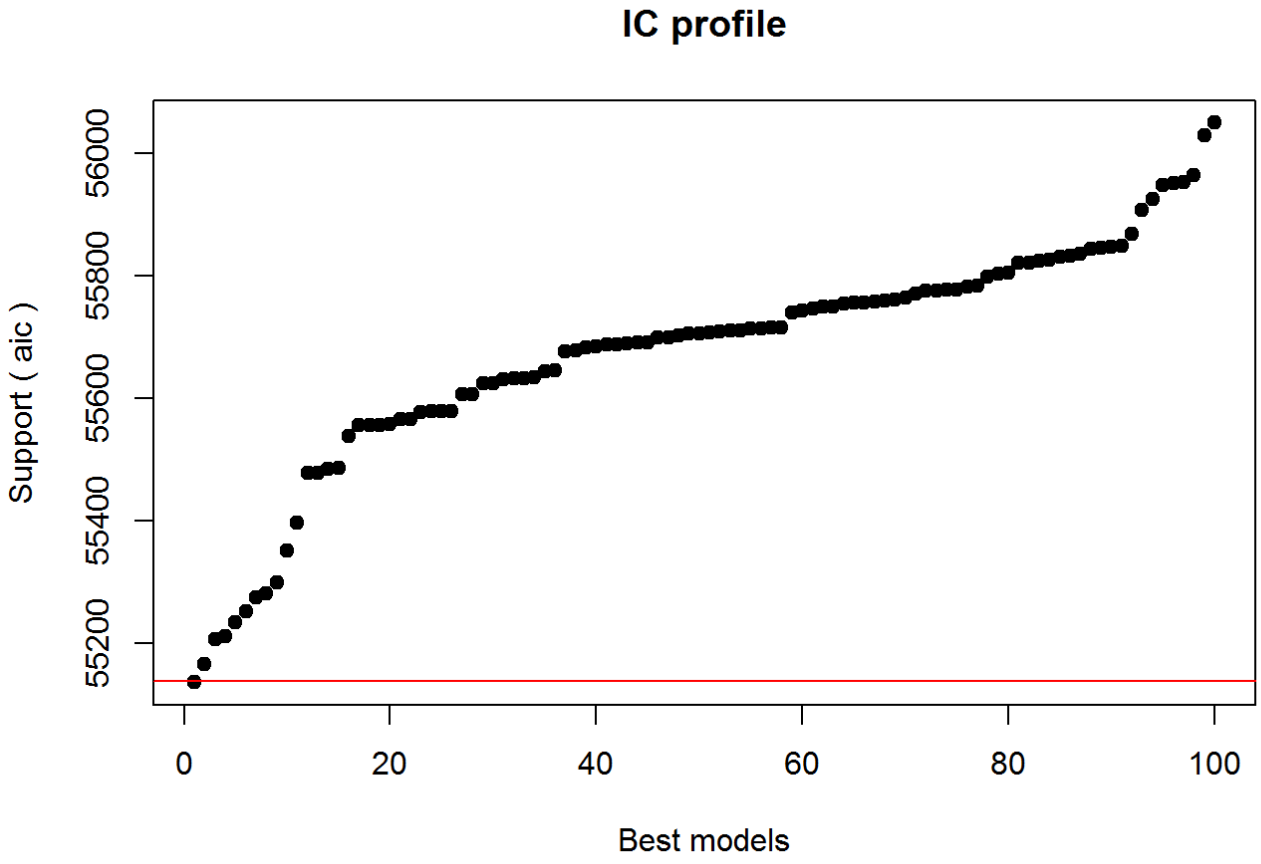

IC profile
Support (aic)
56100
56300
56500
56700
0
20
40
60
80
100
Best models

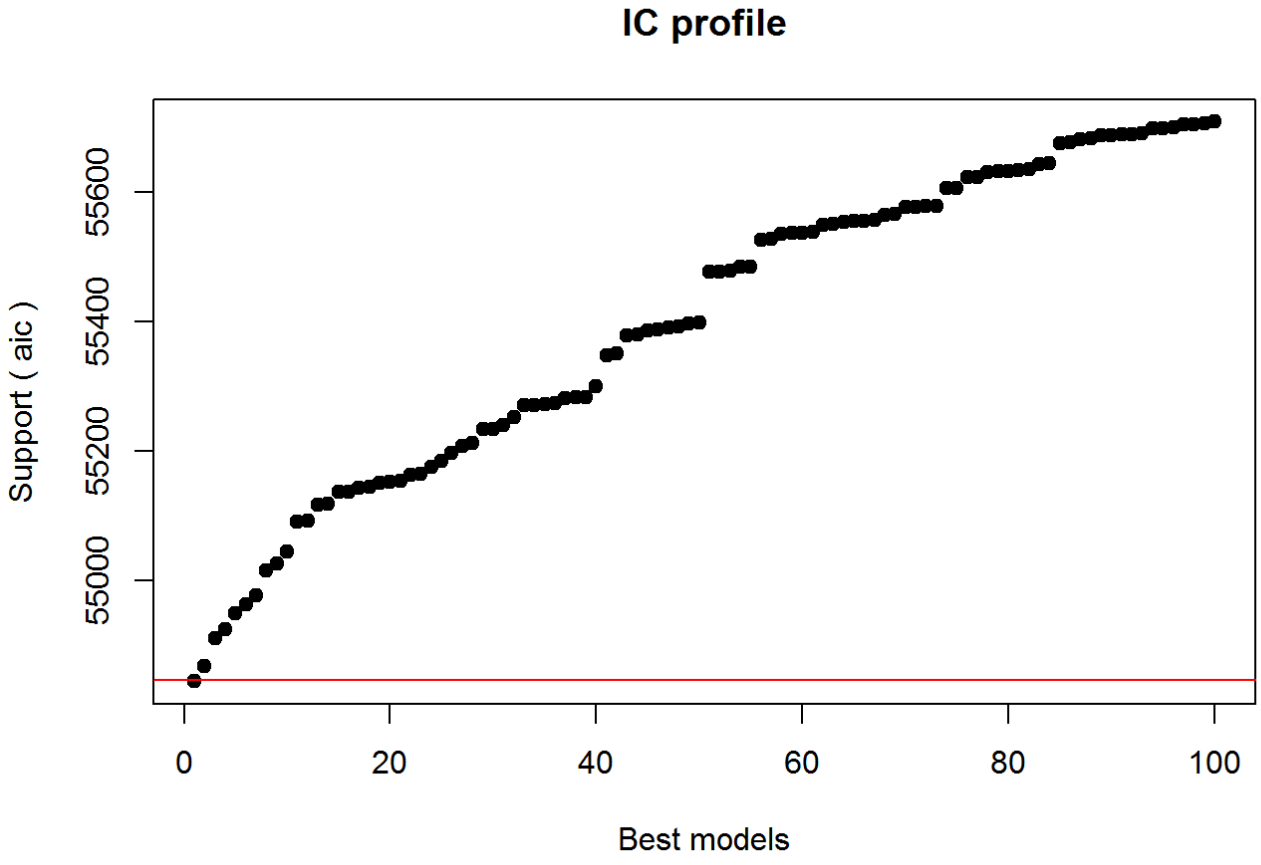

IC profile
Support (aic)
55800
56000
56200
56400
0
20
40
60
80
100
Best models

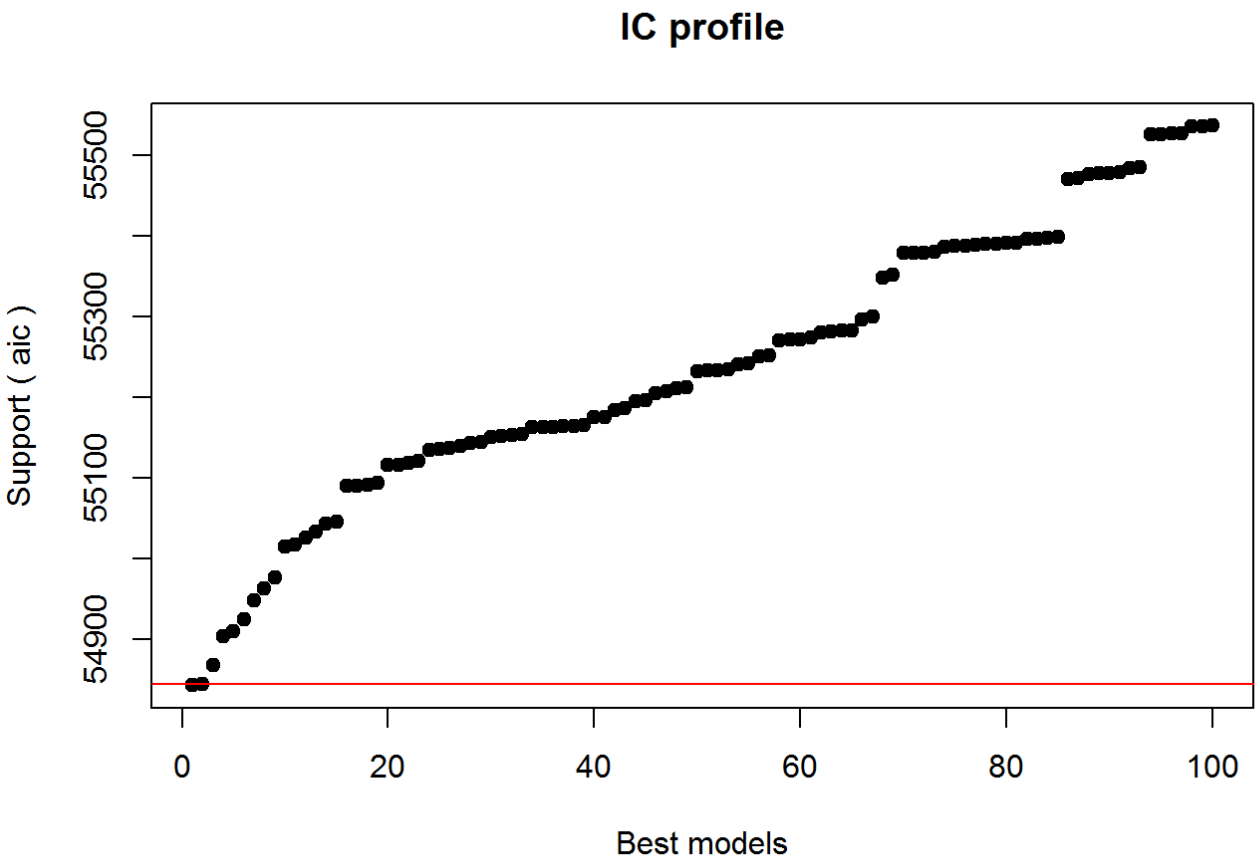

IC profile
Support (aic)
55700
55900
56100
56300
0
20
40
60
80
100
Best models

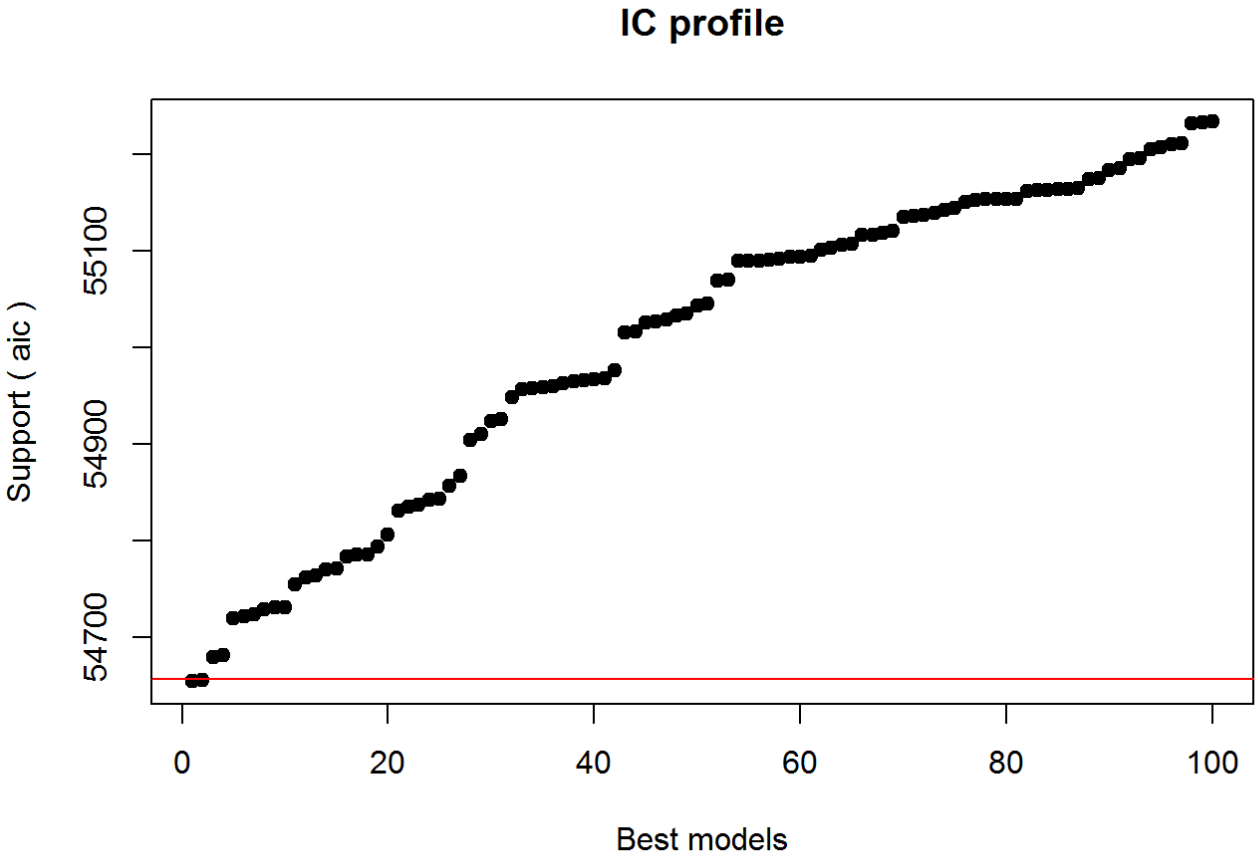

IC profile
Support (aic)
55600
55800
56000
56200
0
20
40
60
80
100
Best models

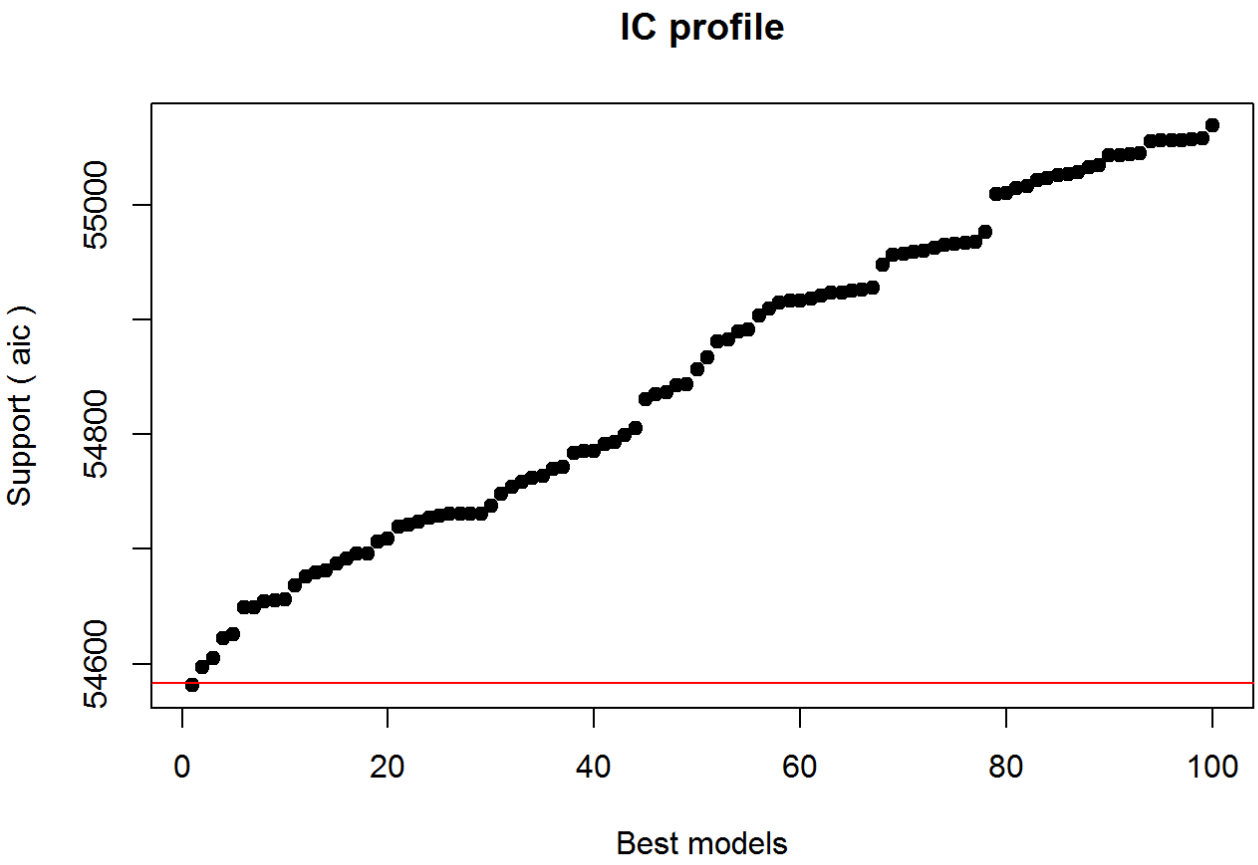

IC profile
Support (aic)
55600
55800
56000
56200
0
20
40
60
80
100
Best models

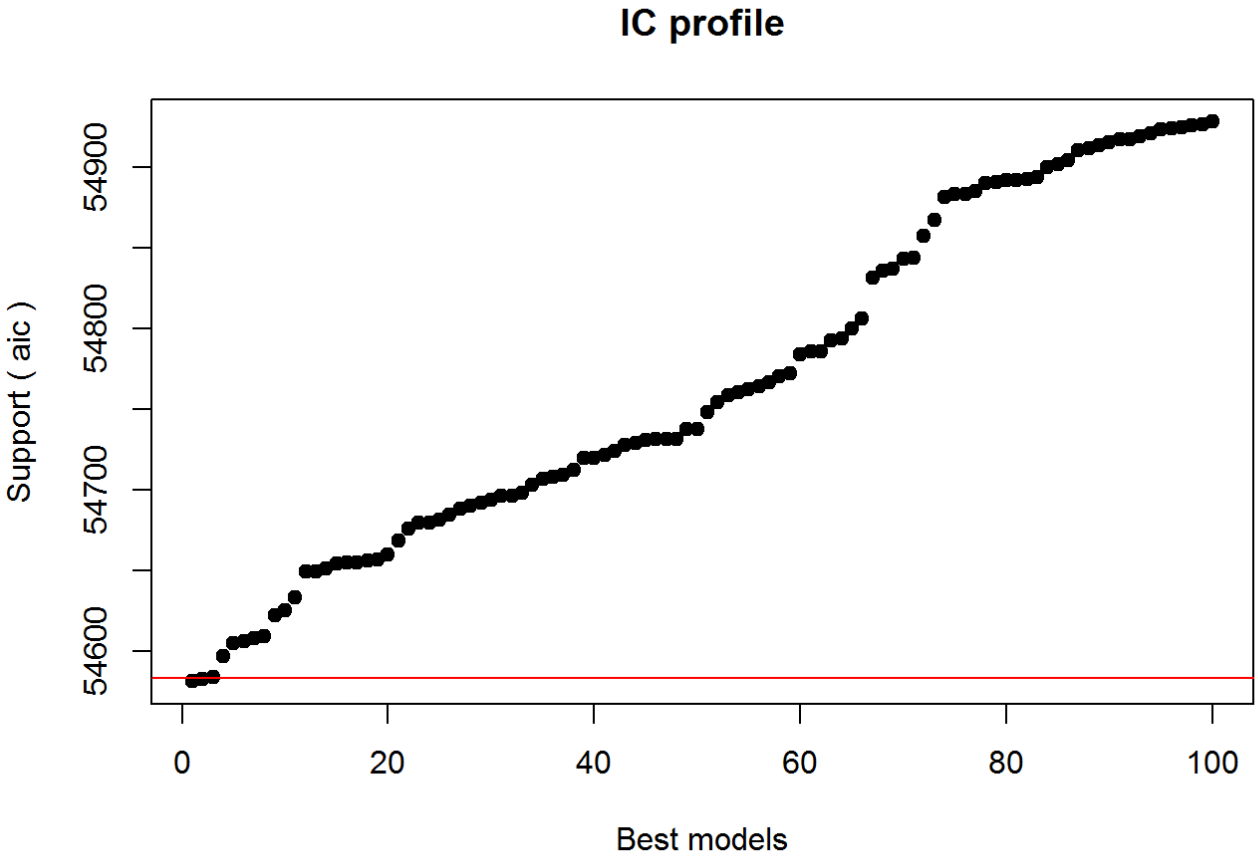

IC profile
Support (aic)
55200
55400
55600
55800
56000
0
20
40
60
80
100
Best models

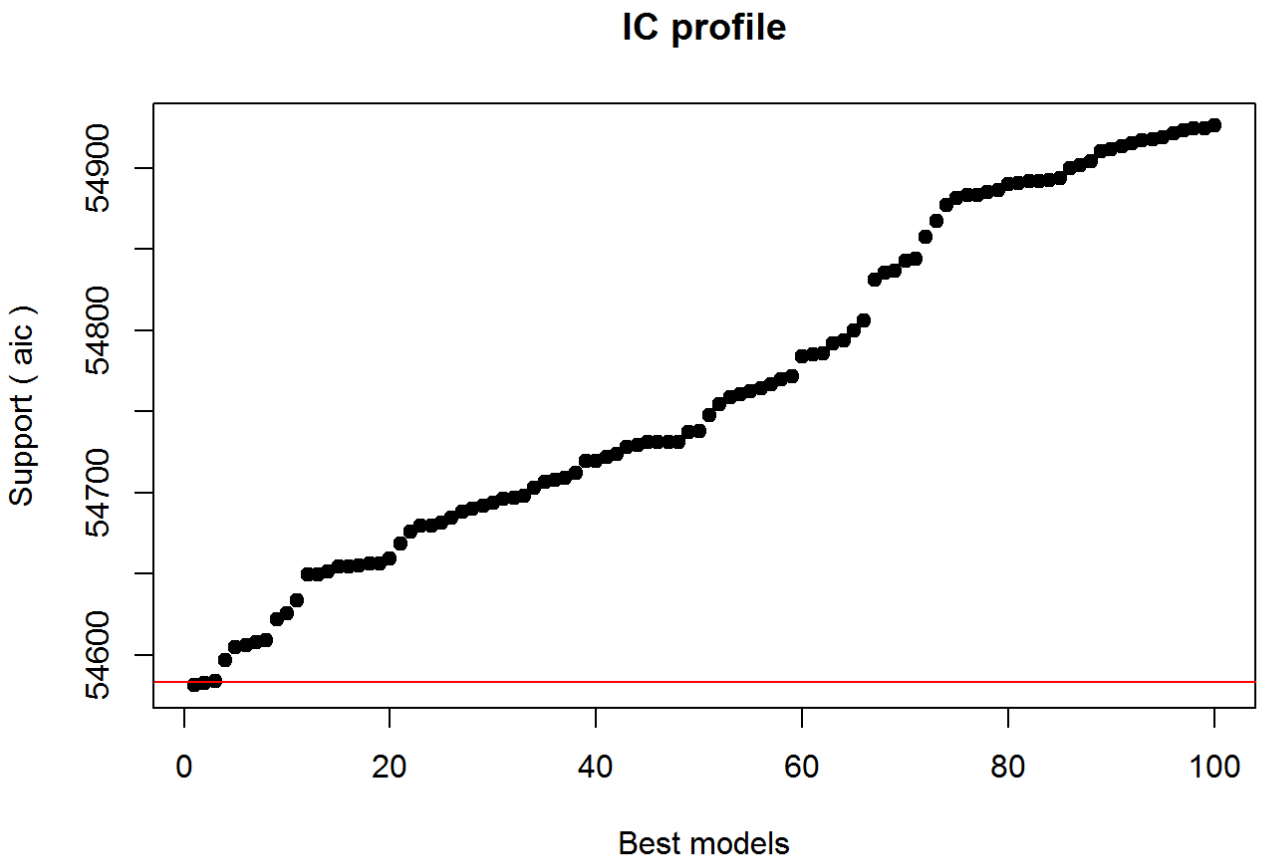

IC profile
Support (aic)
55000
55200
55400
55600
0
20
40
60
80
100
Best models

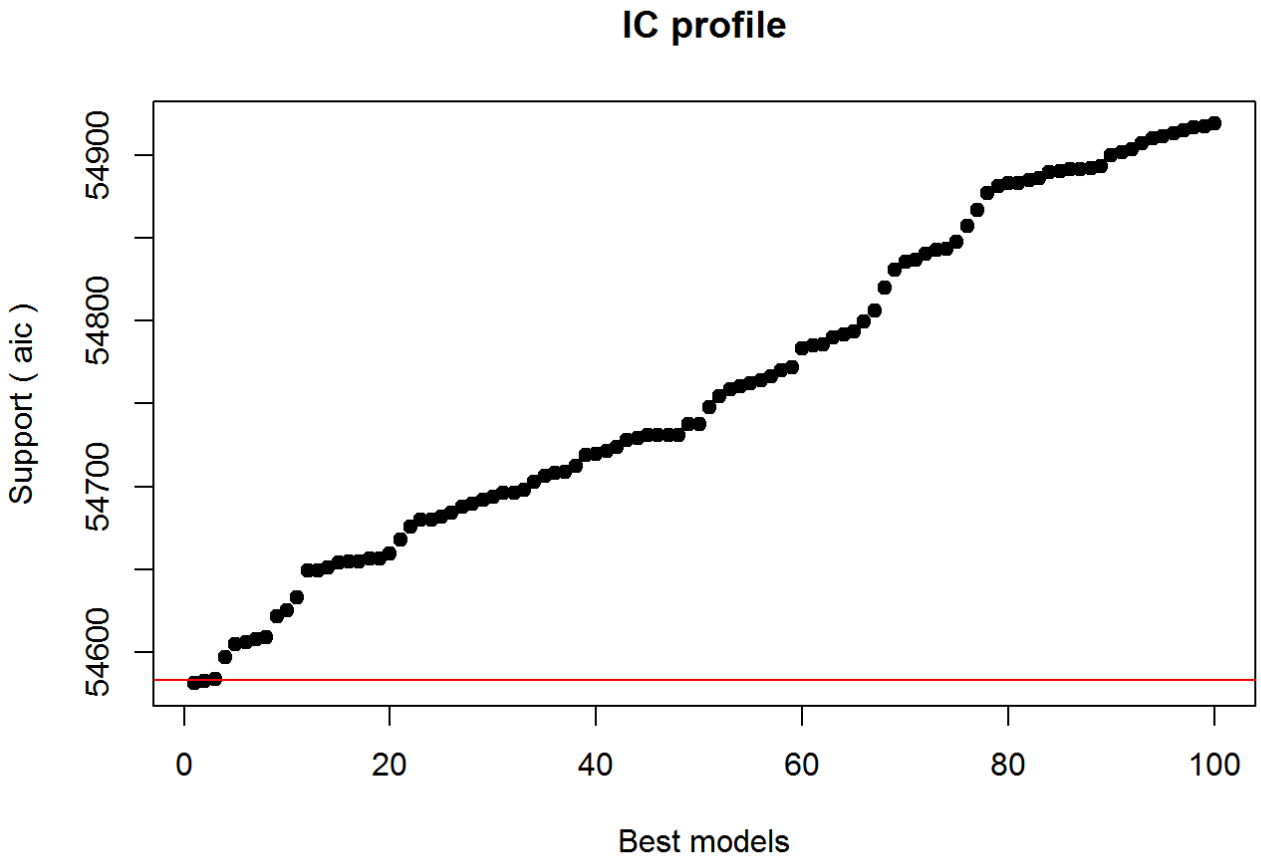

IC profile
Support (aic)
54900
55100
55300
55500
0
20
40
60
80
100
Best models

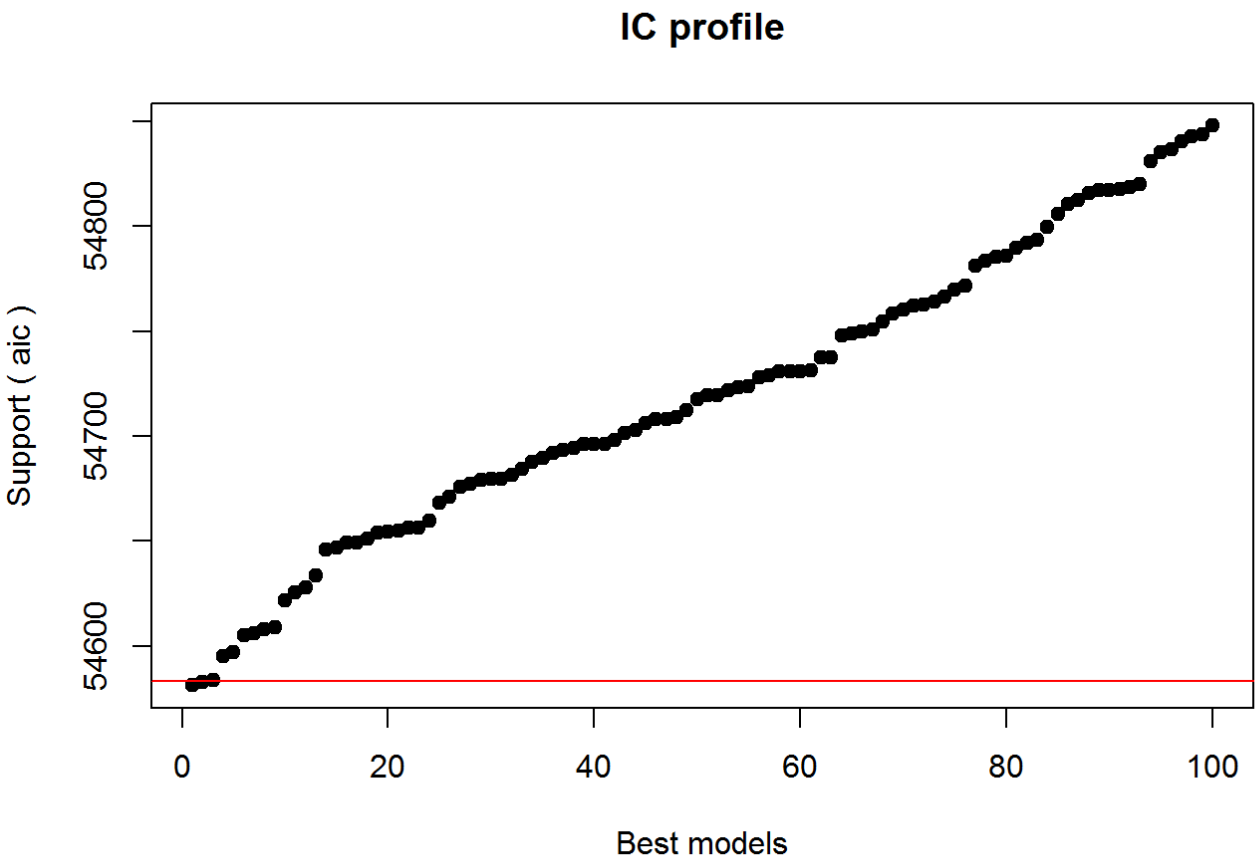

IC profile
Support (aic)
54700
54900
55100
0
20
40
60
80
100
Best models

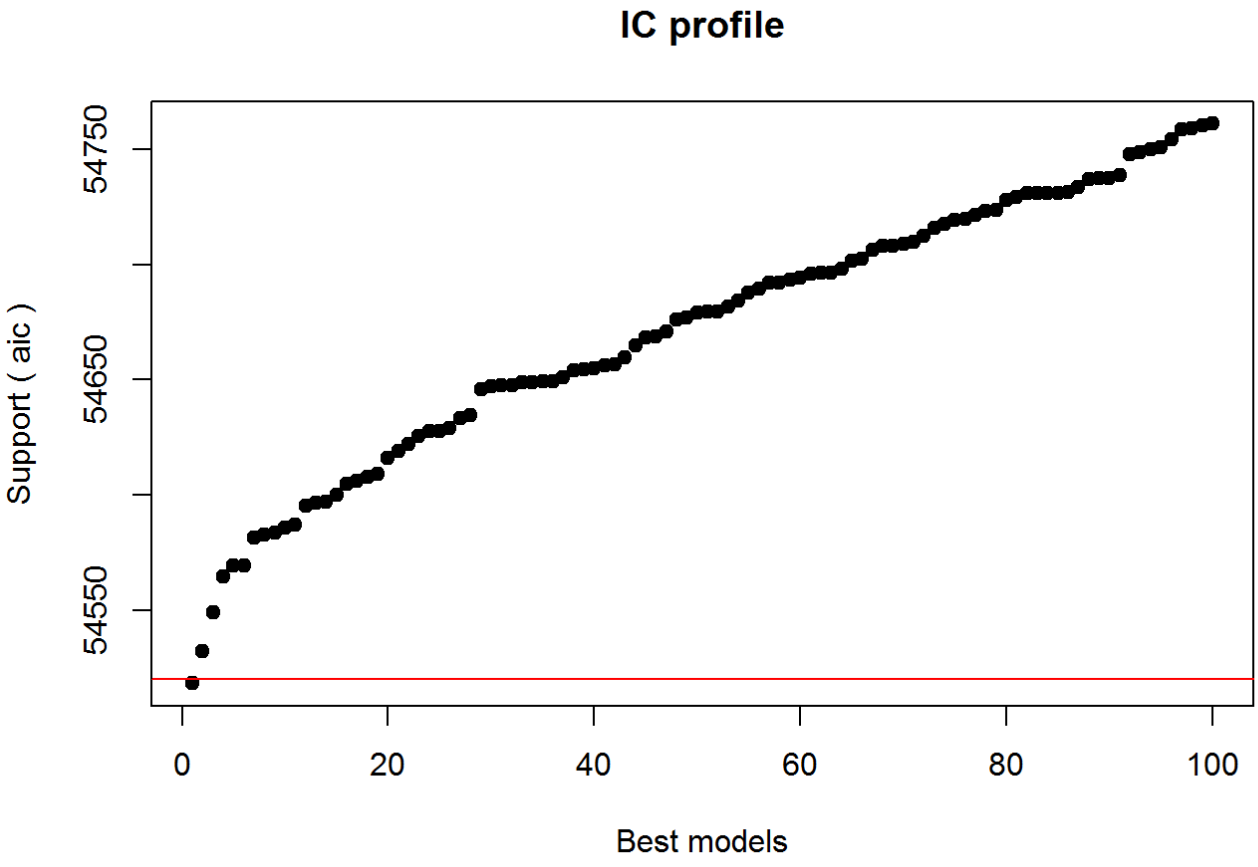

IC profile
Support (aic)
54600
54800
55000
0
20
40
60
80
100
Best models

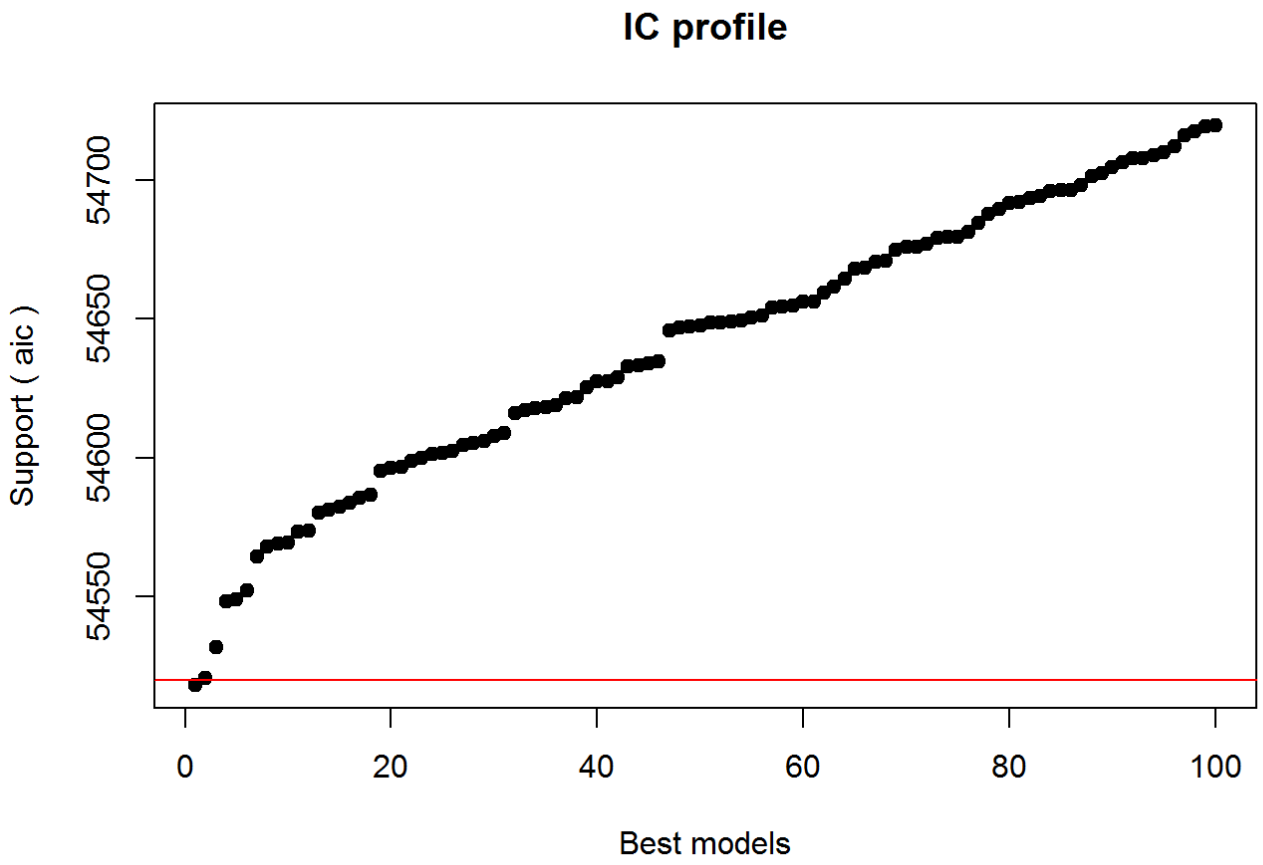

IC profile
Support (aic)
54600
54700
54800
54900
0
20
40
60
80
100
Best models


IC profile
Support (aic)
54600
54700
54800
54900
0
20
40
60
80
100
Best models


IC profile
Support (aic)
54600
54700
54800
54900
0
20
40
60
80
100
Best models


IC profile
Support (aic)
54600
54700
54800
0
20
40
60
80
100
Best models


IC profile
Support (aic)
54550
54650
54750
0
20
40
60
80
100
Best models



Step 14: Begin creating a logistic regression to evaluate the explanatory value of different variables on Significant and Crucial violations.

```
summary(best.model)
```

```
## $name
## [1] "glmulti.analysis"
##
## $method
## [1] "h"
##
## $fitting
## [1] "glm"
##
## $crit
## [1] "aic"
##
## $level
## [1] 1
##
## $marginality
## [1] FALSE
##
## $confsetsize
## [1] 100
##
## $bestic
## [1] 54518.19
##
## $icvalues
##   [1] 54518.19 54519.73 54520.83 54532.03 54548.52 54549.15 54550.70
##   [8] 54552.25 54564.58 54567.92 54569.12 54569.46 54570.83 54571.31
##  [15] 54573.43 54573.99 54580.32 54581.20 54582.53 54583.68 54585.62
##  [22] 54586.82 54595.24 54596.42 54596.90 54598.73 54599.88 54601.48
##  [29] 54601.77 54602.41 54604.79 54605.51 54606.14 54607.69 54608.87
##  [36] 54616.01 54617.13 54617.90 54618.26 54618.83 54621.53 54621.86
##  [43] 54625.34 54627.44 54627.65 54629.11 54629.22 54633.08 54633.31
##  [50] 54633.95 54634.66 54645.69 54646.98 54647.37 54647.63 54648.81
##  [57] 54648.83 54649.13 54649.36 54650.40 54651.11 54654.13 54654.52
##  [64] 54654.84 54656.23 54656.39 54659.43 54661.56 54664.65 54668.17
##  [71] 54668.52 54670.47 54671.02 54675.05 54675.90 54675.94 54677.08
##  [78] 54679.02 54679.66 54679.67 54681.50 54684.43 54687.81 54689.61
##  [85] 54691.84 54692.04 54693.52 54694.07 54695.98 54696.32 54696.37
##  [92] 54698.18 54701.37 54702.62 54704.78 54706.33 54707.96 54707.97
##  [99] 54708.91 54709.97
##
## $bestmodel
## [1] "Sev_Cru ~ 1 + TYPE_RESTAURANT + TYPE_INSTITUTION + TYPE_TAKEOUT + "
## [2] "    TYPE_FOODCOURT + MUN_FMR_TORONTO + MUN_SCARBOROUGH + MUN_ETOBICOKE + "
## [3] "    INSP_2 + INSP_3"
##
## $modelweights
##   [1] 5.776098e-01 2.674942e-01 1.543260e-01 5.696728e-04 1.494793e-07
##   [6] 1.093954e-07 5.024946e-08 2.319228e-08 4.864347e-11 9.159960e-12
##  [11] 5.046636e-12 4.240559e-12 2.138270e-12 1.682317e-12 5.824671e-13
```

```
##  [16] 4.418092e-13 1.864523e-14 1.199027e-14 6.177280e-15 3.472305e-15
##  [21] 1.314004e-15 7.236710e-16 1.070455e-17 5.931976e-18 4.682323e-18
##  [26] 1.870570e-18 1.054243e-18 4.737471e-19 4.089861e-19 2.970279e-19
##  [31] 9.060240e-20 6.318796e-20 4.598036e-20 2.122819e-20 1.178832e-20
##  [36] 3.315337e-22 1.889414e-22 1.287846e-22 1.075849e-22 8.086225e-23
##  [41] 2.094692e-23 1.777554e-23 3.119510e-24 1.089617e-24 9.853736e-25
##  [46] 4.741014e-25 4.486857e-25 6.519449e-26 5.786773e-26 4.206786e-26
##  [51] 2.955794e-26 1.186749e-28 6.226709e-29 5.129205e-29 4.513366e-29
##  [56] 2.501427e-29 2.467840e-29 2.126567e-29 1.893958e-29 1.126621e-29
##  [61] 7.911020e-30 1.744603e-30 1.438463e-30 1.225697e-30 6.108813e-31
##  [66] 5.635897e-31 1.237691e-31 4.251388e-32 9.074188e-33 1.559268e-33
##  [71] 1.309213e-33 4.948680e-34 3.760915e-34 5.013142e-35 3.272354e-35
##  [76] 3.214099e-35 1.813460e-35 6.866276e-36 4.991956e-36 4.968946e-36
##  [81] 1.992248e-36 4.590292e-37 8.501862e-38 3.452773e-38 1.130697e-38
##  [86] 1.025776e-38 4.889573e-39 3.710694e-39 1.429630e-39 1.208057e-39
##  [91] 1.176217e-39 4.763978e-40 9.640299e-41 5.161850e-41 1.749484e-41
##  [96] 8.081494e-42 3.579192e-42 3.550910e-42 2.224944e-42 1.307634e-42
##
## $includeobjects
## [1] TRUE
```

```
final.model <- glm(Sev_Cru ~ 1 + TYPE_RESTAURANT + TYPE_INSTITUTION + TYPE_TAKEOUT + T
YPE_FOODCOURT + MUN_FMR_TORONTO + MUN_SCARBOROUGH + MUN_ETOBICOKE + INSP_2 + INSP_3, d
ata = inspect_work1, family = binomial("logit"))

summary(final.model)
```

```
##
## Call:
## glm(formula = Sev_Cru ~ 1 + TYPE_RESTAURANT + TYPE_INSTITUTION +
##     TYPE_TAKEOUT + TYPE_FOODCOURT + MUN_FMR_TORONTO + MUN_SCARBOROUGH +
##     MUN_ETOBICOKE + INSP_2 + INSP_3, family = binomial("logit"),
##     data = inspect_work1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0548  -0.7055  -0.6109  -0.3271   2.7168
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -2.33474    0.04664 -50.056  < 2e-16 ***
## TYPE_RESTAURANTTRUE   0.42226    0.02896  14.580  < 2e-16 ***
## TYPE_INSTITUTIONTRUE -0.30690    0.06117  -5.018 5.23e-07 ***
## TYPE_TAKEOUTTRUE      0.29347    0.03721   7.887 3.09e-15 ***
## TYPE_FOODCOURTTRUE    0.44012    0.05727   7.685 1.52e-14 ***
## MUN_FMR_TORONTOTRUE   0.10142    0.02811   3.607  0.00031 ***
## MUN_SCARBOROUGHTRUE   0.32027    0.03530   9.072  < 2e-16 ***
## MUN_ETOBICOKETRUE    -1.02359    0.05573 -18.367  < 2e-16 ***
## INSP_2TRUE            0.45706    0.04437  10.302  < 2e-16 ***
## INSP_3TRUE            1.27890    0.04475  28.576  < 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: 56665  on 53789  degrees of freedom
## Residual deviance: 53258  on 53780  degrees of freedom
## AIC: 53278
##
## Number of Fisher Scoring iterations: 5
```

Step 15: Using the variables identified in the best model, build a classfication tree.

```
library(rpart)
library(tree)
```

```
## Warning: package 'tree' was built under R version 3.3.2
```

```
## Subset the m1 data frame into training and testing datasets.
rn_train <- sample(nrow(inspect_work1), floor(nrow(inspect_work1)*0.65))
inspect_train <- inspect_work1[rn_train,]
inspect_test <- inspect_work1[-rn_train,]

inspect_work1$Sev_Cru <- as.factor(inspect_work1$Sev_Cru)

## Attempt to build tree using 'RPART'.
rpart.fit <- rpart(formula = Sev_Cru ~ TYPE_RESTAURANT + TYPE_INSTITUTION + TYPE_TAKEO
UT + TYPE_FOODCOURT + MUN_FMR_TORONTO + MUN_SCARBOROUGH + MUN_ETOBICOKE + INSP_2 + INS
P_3, data = inspect_work1)
summary(rpart.fit)
```

```
## Call:
## rpart(formula = Sev_Cru ~ TYPE_RESTAURANT + TYPE_INSTITUTION +
##     TYPE_TAKEOUT + TYPE_FOODCOURT + MUN_FMR_TORONTO + MUN_SCARBOROUGH +
##     MUN_ETOBICOKE + INSP_2 + INSP_3, data = inspect_work1)
##   n= 53790
##
##   CP nsplit rel error xerror xstd
## 1  0      0         1      0    0
##
## Node number 1: 53790 observations
##   predicted class=FALSE  expected loss=0.219855  P(node) =1
##     class counts: 41964 11826
##    probabilities: 0.780 0.220
```

```
## Attempt to build tree using 'TREE'.
tree.fit <- tree(formula = Sev_Cru ~ TYPE_RESTAURANT + TYPE_INSTITUTION + TYPE_TAKEOUT
 + TYPE_FOODCOURT + MUN_FMR_TORONTO + MUN_SCARBOROUGH + MUN_ETOBICOKE + INSP_2 + INSP_
3, data = inspect_work1)
summary(tree.fit)
```

```
##
## Classification tree:
## tree(formula = Sev_Cru ~ TYPE_RESTAURANT + TYPE_INSTITUTION +
##     TYPE_TAKEOUT + TYPE_FOODCOURT + MUN_FMR_TORONTO + MUN_SCARBOROUGH +
##     MUN_ETOBICOKE + INSP_2 + INSP_3, data = inspect_work1)
## Variables actually used in tree construction:
## [1] "INSP_3"
## Number of terminal nodes:  2
## Residual mean deviance:  1.015 = 54600 / 53790
## Misclassification error rate: 0.2199 = 11826 / 53790
```

```
plot(tree.fit, compress = T)
```

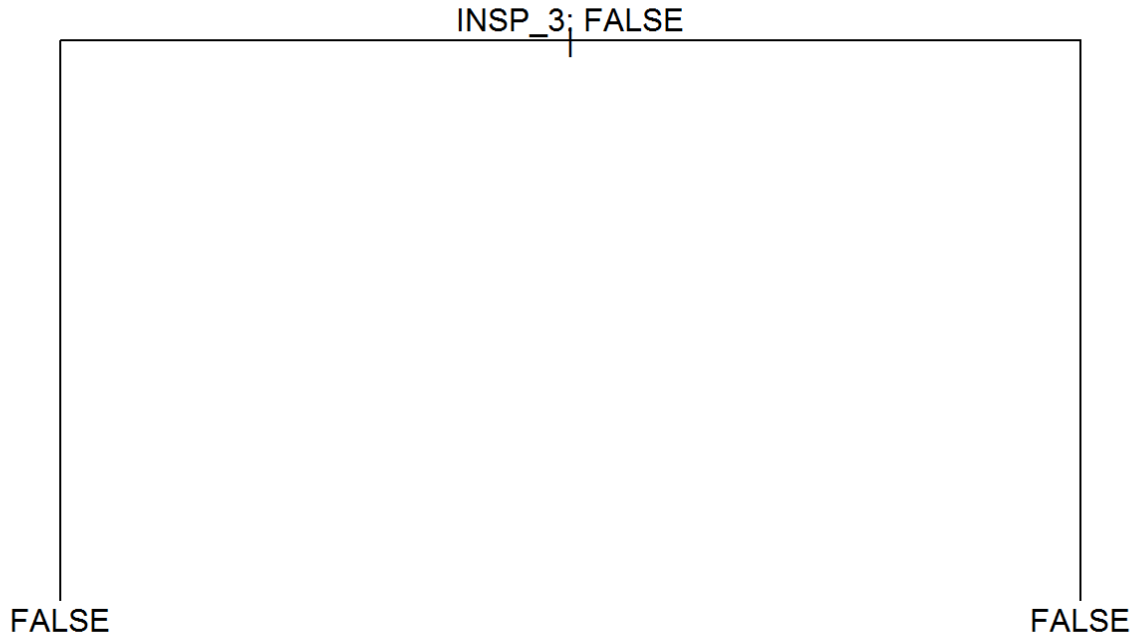
```
## Warning in text.default(x[1L], y[1L], "|", ...): "compress" is not a
## graphical parameter
```

```
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "compress" is not a
## graphical parameter
```

```
text(tree.fit, use.n = T, pretty = 0)
```

```
## Warning in text.default(xy$x[ind], xy$y[ind] + 0.5 * charht, rows[ind], :
## "use.n" is not a graphical parameter
```

```
## Warning in text.default(xy$x[leaves], xy$y[leaves] - 0.5 * charht, labels =
## stat, : "use.n" is not a graphical parameter
```



Step 16: Develop a Random Forest model.

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.3.2
```

```
## randomForest 4.6-12
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.3.2
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:randomForest':
##
##     margin
```

```
## Subset the m1 data frame into training and testing datasets.
set.seed(38)
rn_train <- sample(nrow(inspect_work1), floor(nrow(inspect_work1)*0.65))
inspect_train <- inspect_work1[rn_train,]
inspect_test <- inspect_work1[-rn_train,]

fit <- randomForest(Sev_Cru ~ TYPE_RESTAURANT + TYPE_INSTITUTION + TYPE_TAKEOUT + TYPE
_FOODCOURT + MUN_FMR_TORONTO + MUN_SCARBOROUGH + MUN_ETOBICOKE + INSP_2 + INSP_3, data
 = inspect_train, ntree = 100)
summary(fit)
```

```
##                 Length Class  Mode
## call               4  -none- call
## type               1  -none- character
## predicted      34963  factor numeric
## err.rate         300  -none- numeric
## confusion          6  -none- numeric
## votes          69926  matrix numeric
## oob.times      34963  -none- numeric
## classes            2  -none- character
## importance         9  -none- numeric
## importanceSD       0  -none- NULL
## localImportance    0  -none- NULL
## proximity          0  -none- NULL
## ntree              1  -none- numeric
## mtry               1  -none- numeric
## forest            14  -none- list
## y              34963  factor numeric
## test               0  -none- NULL
## inbag              0  -none- NULL
## terms              3  terms  call
```

```
prediction <- predict(fit, inspect_test)

##Test the random forest using a confusion matrix.
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 3.3.2
```

```
confusionMatrix(data=prediction,inspect_test$Sev_Cru)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE  TRUE
##      FALSE 14658  4169
##      TRUE      0     0
##
##                Accuracy : 0.7786
##                  95% CI : (0.7726, 0.7845)
##     No Information Rate : 0.7786
##     P-Value [Acc > NIR] : 0.5042
##
##                   Kappa : 0
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 1.0000
##             Specificity : 0.0000
##          Pos Pred Value : 0.7786
##          Neg Pred Value :    NaN
##              Prevalence : 0.7786
##          Detection Rate : 0.7786
##    Detection Prevalence : 1.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : FALSE
##
```