4.3 Case Study 2: GLMs for Binary Target Variables

Now that we have gained some hands-on experience with constructing and evaluating GLMs, in this section we look at a more involved insurance-related case study, where we develop GLMs for predicting a binary target variable and perform classifications. Compared to GLMs for numeric target variables, GLM-based classifiers enjoy some subtly unique features, which will be revealed in the course of this case study. At the completion of this case study, you should be able to:

- Combine factor levels to reduce the dimension of the data.
- Select appropriate link functions for binary target variables.
- Implement different kinds of GLMs for binary target variables in R.
- Interpret the results of a fitted logistic regression model.
- Use the results of a GLM to set cutoffs based on practical considerations.

4.3.1 Preparatory Steps

Data description. In this case study we will examine the `vehins` (meaning "vehicle insurance") dataset, which is adapted from the `dataCar` dataset in the `insuranceData` package.[viii] This dataset is based on a total of $n = 67,856$ one-year vehicle insurance policies taken out in 2004 or 2005. The variables in this dataset, described in Table 4.3, pertain to different characteristics of the policyholders and their vehicles. The target variable is the first variable `clm`, a binary variable equal to 1 if a claim occurred over the policy period and 0 otherwise. Our objective here is to construct appropriate GLMs to identify key factors leading to claim occurrence. Such factors will provide insurance companies offering vehicle insurance policies with useful information for understanding the claims-generating mechanism and, by extension, setting fair premium rates for its policyholders.

To get started, run CHUNK 1 to read in the dataset and print a summary of each variable.

```
# CHUNK 1
vehins <- read.csv("vehins.csv")
# The original version of "vehins.csv" can be accessed using the following code
#library(insuranceData)
#vehins <- data(dataCar)

summary(vehins)
```

```
##       clm             exposure          veh_value          numclaims
##  Min.   :0.00000   Min.   :0.002738   Min.   : 0.000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.219028   1st Qu.: 1.010   1st Qu.:0.00000
##  Median :0.00000   Median :0.446270   Median : 1.500   Median :0.00000
##  Mean   :0.06814   Mean   :0.468651   Mean   : 1.777   Mean   :0.07276
##  3rd Qu.:0.00000   3rd Qu.:0.709103   3rd Qu.: 2.150   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :0.999316   Max.   :34.560   Max.   :4.00000
##
```

[viii] The original dataset can also be accessed from `http://www.businessandeconomics.mq.edu.au/our_departments/Applied_Finance_and_Actuarial_Studies/research/books/GLMsforInsuranceData/data_sets`.

Variable	Description	Characteristics
`clm`	Claim occurrence	Integer 0-1 (0 = no, 1 = yes)
`veh_value`	Vehicle value (in $10,000s)	Positive value 0–34.560
`exposure`	Exposure	Positive value 0-1
`numclaims`	Number of claims	Integer 0-4
`claimcst`	Claim size	Positive value 0-55,922.1 (= 0 if no claim)
`veh_body`	Vehicle body type	Coded as `BUS`, `CONVT` (= convertible), `COUPE`, `HBACK` (= hatchback), `HDTOP` (= hardtop), `MCARA` (= motorized caravan), `MIBUS` (= minibus), `PANVN` (= panel van), `RDSTR` (= roadster), `SEDAN`, `STNWG` (= station wagon), `TRUCK`, `UTE` (= utility)
`veh_age`	Vehicle age	Integer 1 (youngest), 2, 3, 4
`gender`	Gender of policyholder	M = male, F = female
`area`	Area of residence of policyholder	A, B, C, D, E, F
`agecat`	Age band of policyholder	Integer 1 (youngest), 2, 3, 4, 5, 6

Table 4.3: Data dictionary for the `vehins` data.

```
##     claimcst          veh_body      veh_age      gender    area
##  Min.   :    0.0   SEDAN  :22233   Min.   :1.000   F:38603   A:16312
##  1st Qu.:    0.0   HBACK  :18915   1st Qu.:2.000   M:29253   B:13341
##  Median :    0.0   STNWG  :16261   Median :3.000             C:20540
##  Mean   :  137.3   UTE    : 4586   Mean   :2.674             D: 8173
##  3rd Qu.:    0.0   TRUCK  : 1750   3rd Qu.:4.000             E: 5912
##  Max.   :55922.1   HDTOP  : 1579   Max.   :4.000             F: 3578
##                    (Other): 2532
##      agecat
##  Min.   :1.000
##  1st Qu.:2.000
##  Median :3.000
##  Mean   :3.485
##  3rd Qu.:5.000
##  Max.   :6.000
##
```

We can see that the mean of `clm` (which is treated in R as a numeric variable) is 0.06814, meaning that 6.814% of the 67,856 vehicle insurance policies had at least one claim over the policy period.

TASK 1: Data pre-processing

Remove any questionable observations and variables that are not suitable as predictors for `clm`.

Because the goal of this case study is to develop GLMs that allow us to predict claim occurrence based on policyholders' characteristics, it is vitally important to make sure that the variables that serve as inputs to the GLMs are available *before claims are observed*. The values of the variables `numclaims` and `claimcst` are known at the same time as or after claims are submitted, so they cannot function as predictors of `clm`, even though they are certainly related to `clm`. In fact, if `numclaims` is 1 or more or `claimcst` is strictly positive, then `clm` is necessarily 1. In CHUNK 2, we drop these two variables from the dataset.

```
# CHUNK 2
vehins$claimcst <- NULL
vehins$numclaims <- NULL
```

The dataset also contains a small number of observations (53, to be exact) which involve a zero vehicle value. Although `veh_value` is measured in \$10,000s, a value of zero for `veh_value` means that the value of the vehicle is lower than $0.001(\$10,000) = \$10$, which does not make practical sense. For this reason, let's also drop these questionable observations.

```
# CHUNK 2 (Cont.)
nrow(vehins[vehins$veh_value == 0, ])

## [1] 53

vehins <- vehins[vehins$veh_value > 0, ]
```

The next few tasks relate to data exploration.

Data exploration.

> **TASK 2: Perform univariate exploration of the non-factor variables**
>
> Decide which numeric variables, if any, should be treated as factors. Use graphical displays and summary statistics to determine if any of the *non-factor* variables except `exposure` should be transformed and, if so, what transformation should be made. Do your recommended transformation(s), if any, and delete the original variable(s).
>
> **Note:** There is no need to analyze the `exposure` variable, which will be treated later.

Out of all the predictors for `clm`, only `veh_value`, `veh_age`, and `agecat` are numeric variables. Since `veh_age` and `agecat` are only labels of age groups, for modeling purposes it may be desirable to treat them as factors instead of numeric variables. This is because even though there is an order among the age categories (e.g., vehicles with `veh_age = 1` are the newest and the higher the value of `veh_age`, the older the vehicles), taking the two variables as numeric variables each with a single regression coefficient attached imposes the undesirable restriction that every unit increase in the two variables is associated with the same change in the linear predictor, holding all other variables fixed (e.g., the change in the linear predictor when `agecat` changes from 1 to 2 equals the change when `agecat` changes from 2 to 3). Such a restriction is lifted if `veh_age` and `agecat` are treated as categorical predictors represented by dummy variables, even though this will increase the dimension of the data (feature selection can remedy this). The conversion of `veh_age` and `agecat` from integers to factors can be achieved by the `as.factor()` function, as in CHUNK 3.

```
# CHUNK 3
# Before conversion
class(vehins$agecat)
class(vehins$veh_age)

vehins$agecat <- as.factor(vehins$agecat)
vehins$veh_age <- as.factor(vehins$veh_age)

# After conversion
class(vehins$agecat)
class(vehins$veh_age)
```

```
## [1] "integer"
## [1] "integer"
## [1] "factor"
## [1] "factor"
```

The rest of CHUNK 3 relevels all of the categorical predictors, including `veh_age` and `agecat`, so that their baseline level is the level with the most observations. The code is essentially the same as CHUNK 9 of Section 3.4.

```
# CHUNK 3 (Cont.)
# Save the names of the factor variables as a vector
# Relevel
vars.cat <- c("veh_age", "veh_body", "gender", "area", "agecat")
for (i in vars.cat){
  table <- as.data.frame(table(vehins[, i]))
  max <- which.max(table[, 2])
  level.name <- as.character(table[max, 1])
  vehins[, i] <- relevel(vehins[, i], ref = level.name)
}

summary(vehins)
```

```
##       clm             exposure          veh_value          veh_body
##  Min.   :0.00000   Min.   :0.002738   Min.   : 0.180   SEDAN  :22232
##  1st Qu.:0.00000   1st Qu.:0.219028   1st Qu.: 1.010   HBACK  :18915
##  Median :0.00000   Median :0.446270   Median : 1.500   STNWG  :16234
##  Mean   :0.06811   Mean   :0.468481   Mean   : 1.778   UTE    : 4586
##  3rd Qu.:0.00000   3rd Qu.:0.709103   3rd Qu.: 2.150   TRUCK  : 1742
##  Max.   :1.00000   Max.   :0.999316   Max.   :34.560   HDTOP  : 1579
##                                                        (Other): 2515
##  veh_age   gender    area      agecat
##  3:20060   F:38584   C:20530   4:16179
##  1:12254   M:29219   A:16302   1: 5734
##  2:16582             B:13328   2:12868
##  4:18907             D: 8161   3:15757
```

```
##                          E: 5907   5:10722
##                          F: 3575   6: 6543
##
```

For all of the categorical predictors, their baseline level is listed first and contains the greatest number of observations, as desired.

We are left with `veh_value` as the only numeric variable to analyze. It ranges from 0.180 to 34.560 and its mean (1.778) is higher than its median (1.500), suggesting that its distribution is right-skewed. This is confirmed by the histogram (see Figure 4.3.1) in CHUNK 4. For such a skewed variable, the log transformation is commonly used, so we create a log-transformed version of `veh_value` called `log_veh_value` and delete the original `veh_value` in the second part of CHUNK 4.

```
# CHUNK 4
library(ggplot2)
ggplot(vehins, aes(x = veh_value)) + geom_histogram()
```

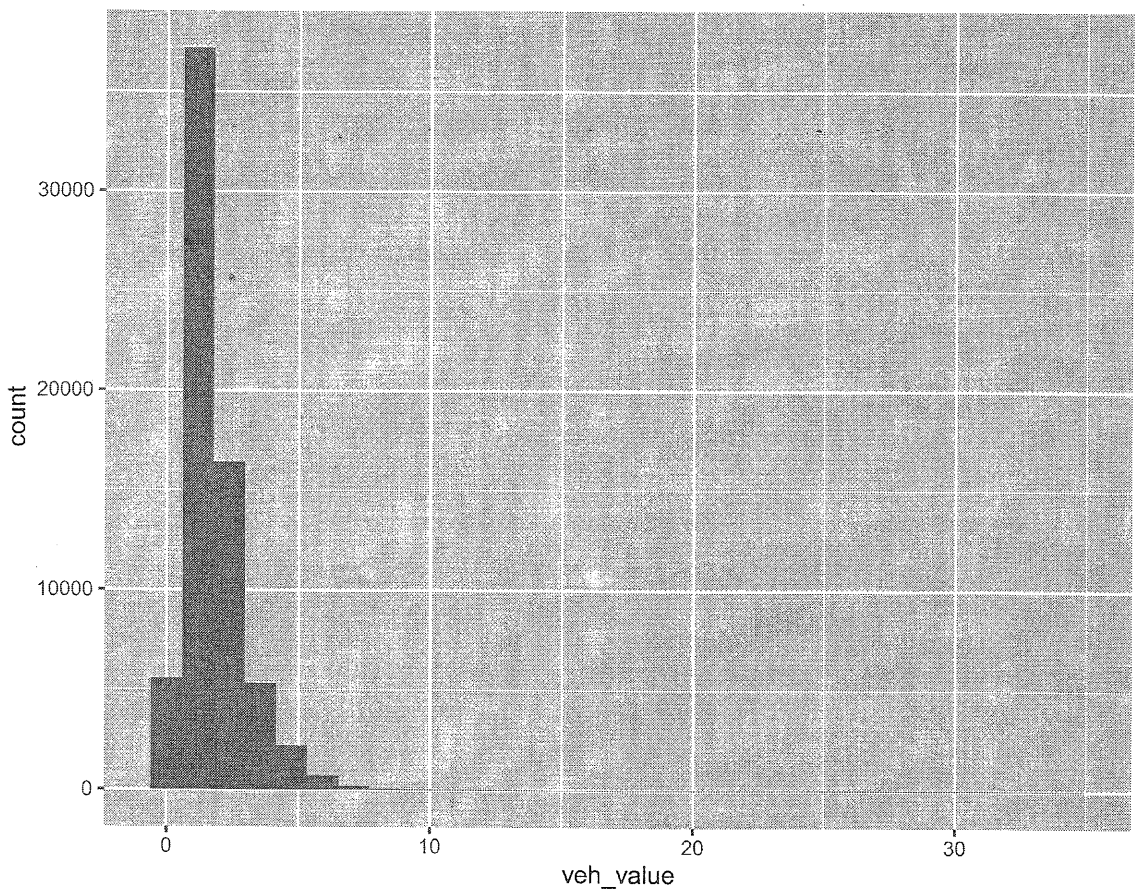

Figure 4.3.1: A histogram for `veh_value` in the `vehins` dataset.

```
# CHUNK 4 (Cont.)
vehins$log_veh_value <- log(vehins$veh_value)
vehins$veh_value <- NULL
```

TASK 3: Explore the relationship of each predictor to `clm`

Use graphical displays and summary statistics to form preliminary conclusions regarding which variables are likely to have significant predictive power.

We have explored the distribution of the target variable `clm` in CHUNK 1 and the distribution of the (only) numeric variable, vehicle value, in CHUNK 2. Now we turn to the relationship between `clm` and each predictor.

First we look at the relationship between `clm` and the only numeric variable, `log_veh_value`. Since we are dealing with a categorical variable and a numeric variable, recall from Subsection 2.2.2 that box plots for the numeric variable split by the levels of the categorical variable will be useful. CHUNK 5 produces these box plots for `log_veh_value` indexed by the two levels (`0` and `1`) of `clm`. Do remember to apply the `factor()` function to `clm` so that it is treated as a factor when making the box plots. It appears that claim occurrence (`clm = 1`) is associated with a slightly higher average value as well as a smaller variation of `log_veh_value`. This positive relationship between `clm` and `log_veh_value` will be quantified by a GLM in later tasks.

The other predictors, `veh_body`, `veh_age` (after conversion), `gender`, `area`, and `agecat` (after conversion), are all categorical variables. For categorical-categorical combinations, stacked bar charts are useful. In CHUNK 6, we use a `for` loop to construct the stacked bar charts for `clm` and each categorical predictor, with the chart for `clm` and `agecat` shown in Figure 4.3.3 as an illustration (the other bar charts can be obtained by running the R code in CHUNK 6). If a categorical variable is an important predictor for `clm`, then the proportions of 0 (in red) and 1 (in green) should vary significantly across different levels of the categorical variable. More noticeable differences[ix] were observed for the following variables:

- `veh_age`: Higher rate of claim occurrence for group 2.
- `veh_body`: Much higher rate for `BUS` and lower rate for `CONVT`, `MIBUS`, and `UTE`.
- `area`: Higher rate for area `F`.
- `agecat`: The claim occurrence rate appears to decrease with `agecat`.

[ix] Don't worry if the differences are small. This is also the case in the June 2019 PA exam!

```
# CHUNK 5
ggplot(vehins, aes(x = factor(clm), y = log_veh_value)) +
  geom_boxplot(fill = "red")
```

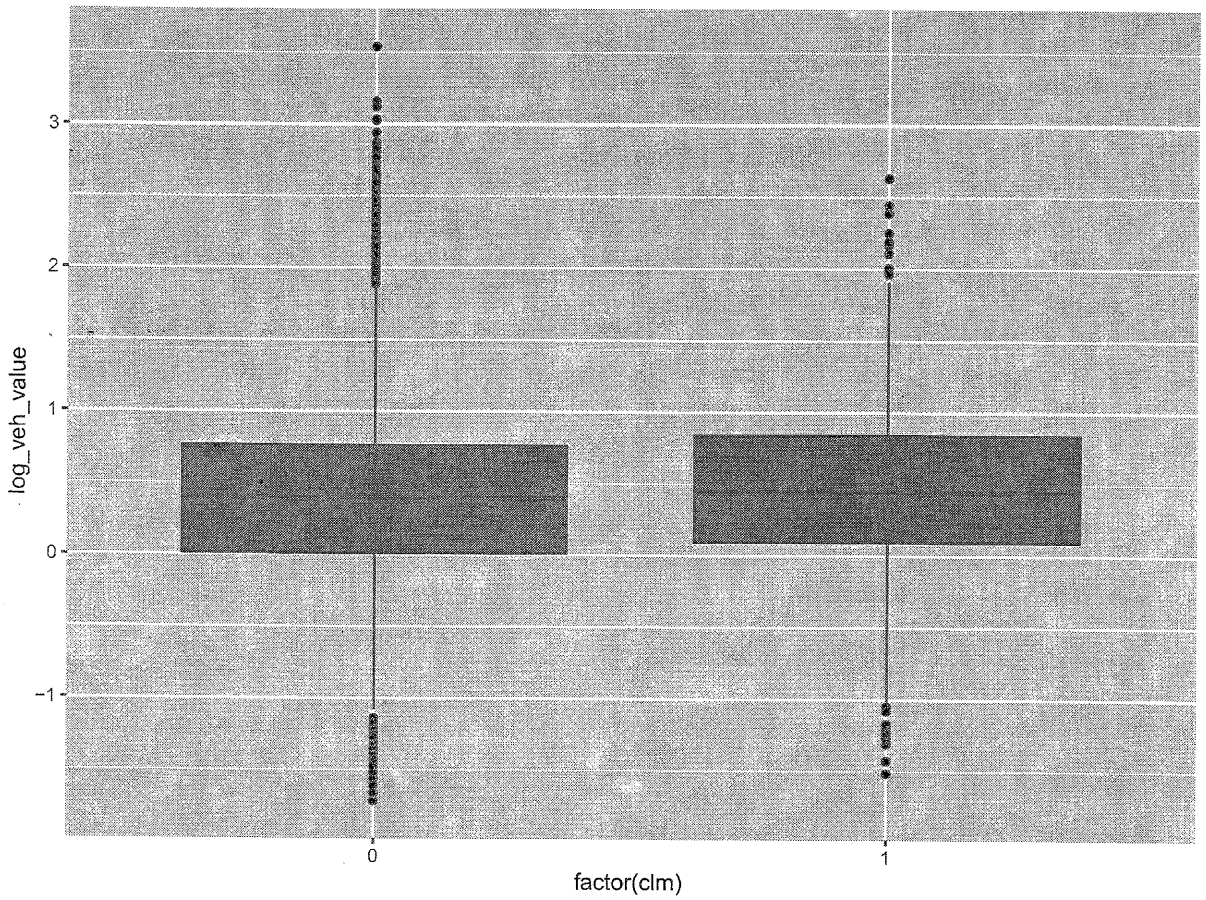

Figure 4.3.2: A split box plot for `log_veh_value` by `clm` in the `vehins` dataset.

```
# CHUNK 6
# Save the names of the factor variables as a vector
for (i in vars.cat) {
  plot <- ggplot(vehins, aes(x = vehins[, i], fill = factor(clm))) +
    geom_bar(position = "fill") +
    labs(x = i, y = "percent") +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))
  print(plot)
}
```

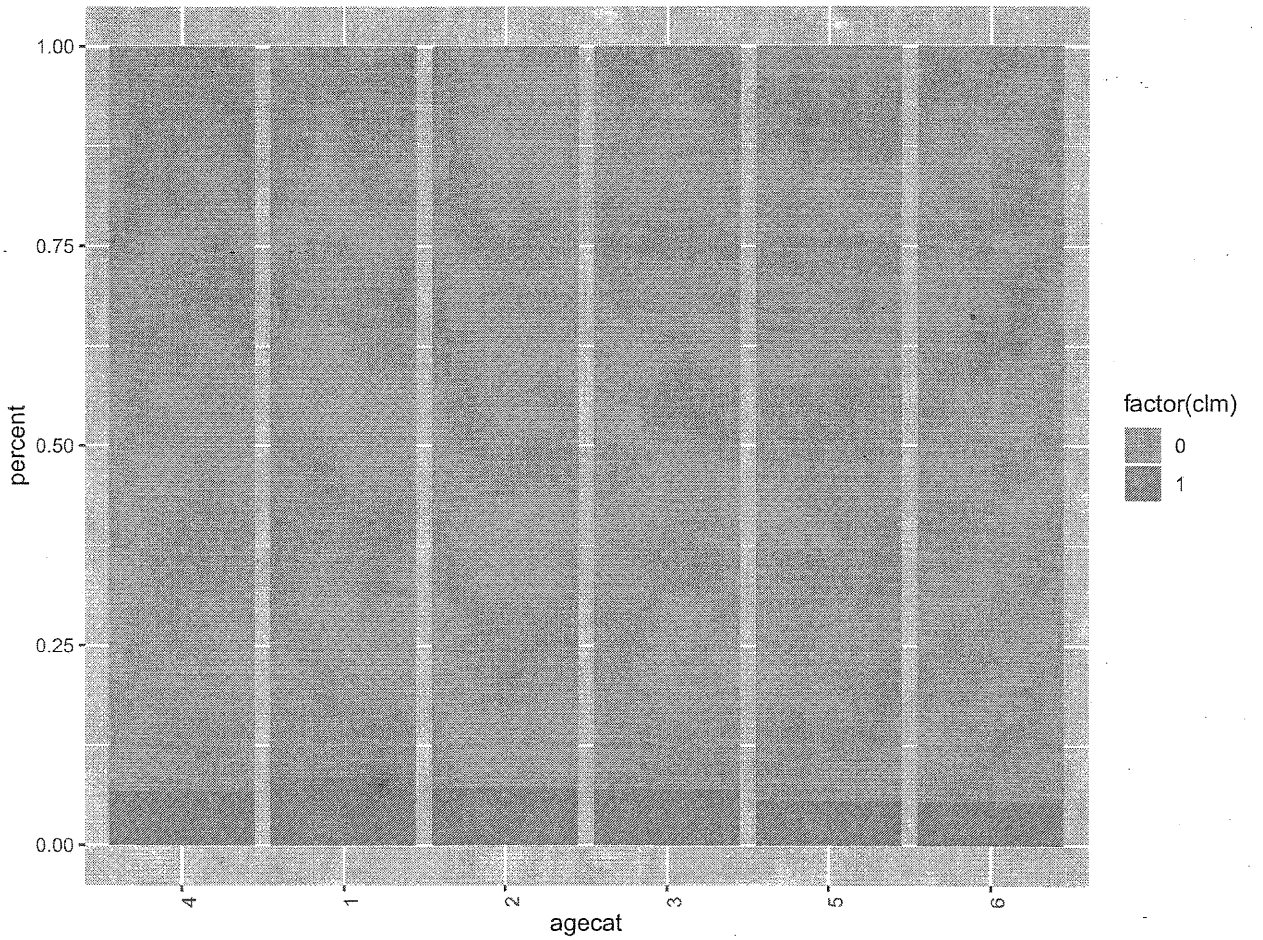

Figure 4.3.3: A bar plot for agecat by clm in the vehins dataset.

We can also obtain a more accurate understanding of how the categorical predictors affect `clm` by looking at the mean of `clm` by different levels of the predictors, as in CHUNK 7. The code, which makes use of the `plyr` and `dplyr` packages, is adapted from the code directly provided on the June 2019 PA exam, so don't worry if the code sounds unfamiliar to you. The summary statistics in CHUNK 7 are consistent with what we observe from the stacked bar charts in CHUNK 6.

```
# CHUNK 7
library(plyr)
library(dplyr)
for (i in vars.cat) {
  print(i)
  x <- vehins %>% group_by_(i)%>%summarise(mean = mean(clm),
                                           n = n())
  print(x)
}
```

```
## [1] "veh_age"
## # A tibble: 4 x 3
##   veh_age   mean     n
##   <fct>    <dbl> <int>
## 1 3       0.0679 20060
## 2 1       0.0672 12254
## 3 2       0.0759 16582
## 4 4       0.0620 18907
## [1] "veh_body"
## # A tibble: 13 x 3
##    veh_body   mean     n
##    <fct>     <dbl> <int>
##  1 SEDAN    0.0664 22232
##  2 BUS      0.184     38
##  3 CONVT    0.0370    81
##  4 COUPE    0.0872   780
##  5 HBACK    0.0668 18915
##  6 HDTOP    0.0823  1579
##  7 MCARA    0.116    121
##  8 MIBUS    0.0601   716
##  9 PANVN    0.0824   752
## 10 RDSTR    0.0741    27
## 11 STNWG    0.0721 16234
## 12 TRUCK    0.0683  1742
## 13 UTE      0.0567  4586
## [1] "gender"
## # A tibble: 2 x 3
##   gender   mean     n
##   <fct>   <dbl> <int>
## 1 F      0.0686 38584
## 2 M      0.0675 29219
```

```
## [1] "area"
## # A tibble: 6 x 3
##   area    mean     n
##   <fct>  <dbl> <int>
## 1 C     0.0688 20530
## 2 A     0.0664 16302
## 3 B     0.0723 13328
## 4 D     0.0607  8161
## 5 E     0.0652  5907
## 6 F     0.0783  3575
## [1] "agecat"
## # A tibble: 6 x 3
##   agecat   mean     n
##   <fct>   <dbl> <int>
## 1 4      0.0682 16179
## 2 1      0.0863  5734
## 3 2      0.0724 12868
## 4 3      0.0705 15757
## 5 5      0.0572 10722
## 6 6      0.0556  6543
```

TASK 4: Reduce the number of factor levels where appropriate

Several of the factor variables have a small number of observations at some of the levels. Consider using knowledge of the factor levels as well as evidence from the previous task to combine some of them into factor levels with more observations.

The `vehins` dataset features a number of categorical predictors, some of which like `veh_body` and `area` have a large number of levels. This significantly increases the dimension of the data and may dilute the predictive power of the GLMs we construct. To reduce the dimension of the data, we can combine factor levels with either similar claim occurrence rates or very few observations to form more populous levels. Based on the mean claim occurrence rates grouped by factor levels shown in CHUNK 7, the following combinations can be made: (there are many possible combinations; the combinations you choose are not as important as your justification!)

- `veh_body`: Combine all levels other than `BUS`, `CONVT`, `MCARA` as one big level called `HYBRID` due to their similar claim occurrence rates. Although each of `BUS`, `CONVT`, `MCARA` has only a few observations, their claim occurrence rates are so different from the claim occurrence rates of other levels that they should better be considered separate levels.
- `area`: Combine A, C, D, and E as one level called `BASE` due to their similar claim occurrence rates and B and F as another level called `OTHER`.
- `agecat`: Combine groups 2, 3, and 4 as one level called group 2, and groups 5 and 6 as another level called group 3.

EXAM NOTE

The June 2019 exam model solutions says that when doing combinations,

> "[t]he best candidates found multiple cases where factor levels could be combined, considered the *similarity of means/medians*, and provided adequate rationale for combining levels. It was *not* sufficient to only combine those levels with *extremely low counts*."

These combinations are done in CHUNK 8, where the code was adapted from the code provided on the June 2019 PA exam. You are not responsible for knowing the details of the `mapvalues()` function, which is from the `plyr` package. If you are interested in knowing, the first argument of the function is the factor variable to work on, and the second and third arguments are the original levels and new levels of the factor variable. To combine the factor levels, we merely have to spell out the third argument appropriately. The last part of CHUNK 8 relevels the new factor variables to ensure that the baseline level is the most populous level.

```
# CHUNK 8
# veh_body
var <- "veh_body"
var.levels <- levels(vehins[, var])
vehins[,var] <- mapvalues(vehins[, var], var.levels,
                          c("HYBRID", "BUS", "CONVT", "HYBRID", "HYBRID",
                            "HYBRID", "MCARA", "HYBRID", "HYBRID", "HYBRID",
                            "HYBRID", "HYBRID", "HYBRID"))

# area
var <- "area"
var.levels <- levels(vehins[, var])
vehins[,var] <- mapvalues(vehins[, var], var.levels,
                          c("BASE", "BASE", "OTHER", "BASE", "BASE", "OTHER"))

# agecat
var <- "agecat"
var.levels <- levels(vehins[, var])
vehins[,var] <- mapvalues(vehins[, var], var.levels,
                          c("2", "1", "2", "2", "3", "3"))

# Relevel
for (i in vars.cat){
  table <- as.data.frame(table(vehins[, i]))
  max <- which.max(table[, 2])
  level.name <- as.character(table[max, 1])
  vehins[, i] <- relevel(vehins[, i], ref = level.name)
}

summary(vehins)
```

```
##       clm              exposure          veh_body     veh_age   gender
##  Min.   :0.00000   Min.   :0.002738   HYBRID:67563   3:20060   F:38584
##  1st Qu.:0.00000   1st Qu.:0.219028   BUS    :   38   1:12254   M:29219
##  Median :0.00000   Median :0.446270   CONVT  :   81   2:16582
##  Mean   :0.06811   Mean   :0.468481   MCARA  :  121   4:18907
##  3rd Qu.:0.00000   3rd Qu.:0.709103
##  Max.   :1.00000   Max.   :0.999316
##     area        agecat     log_veh_value
##  BASE :50900   2:44804   Min.   :-1.71480
##  OTHER:16903   1: 5734   1st Qu.: 0.00995
##                3:17265   Median : 0.40547
##                          Mean   : 0.38675
##                          3rd Qu.: 0.76547
##                          Max.   : 3.54270
```

As the summary output shows, all the combinations have been done as intended.

TASK 5: Select an interaction

Select one pair of features that may be included as an interaction variable in your GLM. Do this by first proposing two variables that are likely to interact and then using appropriate graphical displays to confirm the existence of an interaction. Continue until a promising interaction has been identified. Include your selected interaction variables when constructing a GLM in subsequent tasks.

Recall that an interaction is indicated when changing the level of one variable alters the relationship between the target variable and another variable. In CHUNK 9, we explore the possible interaction between `log_veh_value` and `veh_body` via the use of faceted box plots (see Figure 4.3.4). It is possible that the effect of `log_veh_value` on claim occurrence differs for different kinds of vehicles. We can see that `log_veh_value` is positively associated with claim occurrence for `HYBRID`, `CONVT`, and `MCARA` vehicles, but the reverse relationship is observed for `BUS`. We will include the interaction between these two variables when constructing GLMs in later tasks.

```
# CHUNK 9
ggplot(vehins, aes(x = factor(clm), y = log_veh_value)) +
  geom_boxplot() +
  facet_wrap(~ veh_body) +
  ylim(-1, 1)
```

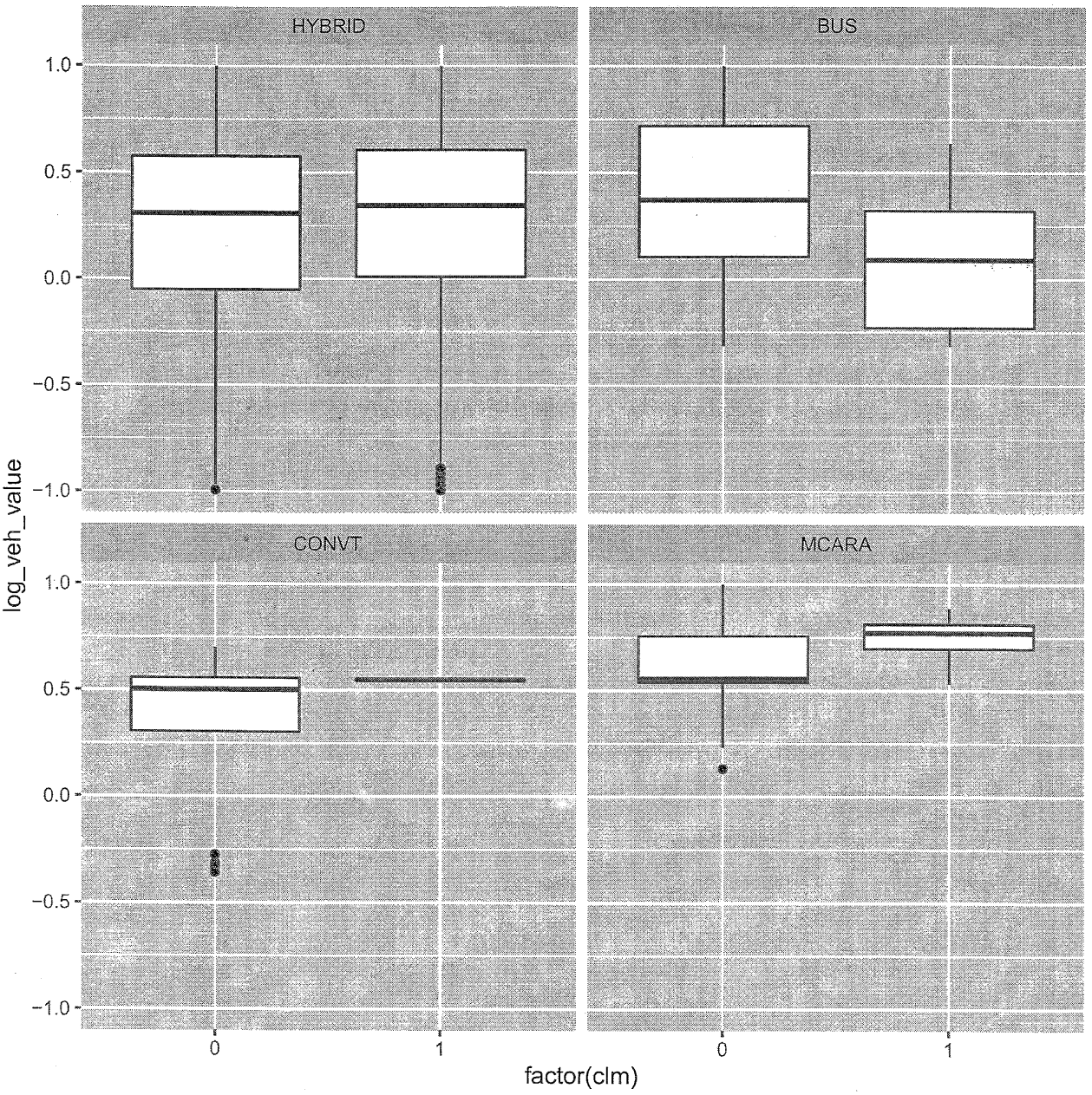

Figure 4.3.4: Faceted split box plots for `log_veh_value` by `clm` and `veh_body` in the `vehins` data.

4.3.2 Model Construction and Selection

Having explored the data and done some preliminary work, we now get to the meat of this case study and proceed to construct GLMs for explaining the claim occurrence rate based on other risk factors in the vehicle insurance dataset.

TASK 6: Select a link function

With the target variable being only 0 or 1, the binomial distribution is the only reasonable choice. For the glm package in R there are five link functions that can be used with the binomial distribution. They are shown below, where η is the linear predictor and p is the probability that the target variable equals 1.

- Logit (`link = "logit"`): $\ln\frac{p}{1-p} = \eta$ or $p = \frac{e^\eta}{1+e^\eta}$
- Probit (`link = "probit"`): $\Phi^{-1}(p) = \eta$ or $p = \Phi(\eta)$, where Φ is the standard normal cumulative distribution function
- Cauchit (`link = "cauchit"`): $p = \frac{1}{\pi}\arctan(\eta) + \frac{1}{2}$ (this is the cumulative distribution function of the standard Cauchy distribution, which is a t distribution with one degree of freedom)
- Log (`link = "log"`): $\ln p = \eta$ or $p = e^\eta$
- Complementary log-log (`link = "cloglog"`): $\ln[-\ln(1-p)] = \eta$ or $p = 1 - e^{-e^\eta}$

Evaluate two potential link functions for applying a GLM to the training set. Explain, prior to fitting the models, why your choices are reasonable for this problem. Fit both models using the features developed in Task 5 and select the better link function, justifying your choice based on the performance of the models on the *training set*. Use only that link function in subsequent tasks.

Note:

(i) Do not compare the models based on their predictive performance on the test set. Predictions, with an exposure adjustment, will be performed in a later task.

(ii) Do not use the `exposure` variable when fitting your GLMs. This variable will be dealt with in a later task.

Since the target variable `clm` is binary, the binomial distribution is the only reasonable candidate for the distribution of the target variable. It remains to specify an appropriate link function.

Among the five link functions above, the log link can be safely ruled out as the mean of the binary target variable, given by $p = e^\eta$ in this case, is not necessarily lying between 0 and 1 (predictions, although non-negative, may be greater than 1, which is impossible for a probability). While the four remaining link functions all ensure a unit-valued mean for the binary target, the logit link is most commonly used and most interpretable due to its intimate connections with the odds of an event. The probit, cauchit, and complementary log-log link functions all produce results which are much more difficult to interpret because of the rather complex relationship between p and η. In fact, the logit link is the default choice in R and is the canonical link function for the binomial

response distribution. In what follows, we will explore the logit and probit links (you can replace probit by the cauchit or complementary log-log links).

Creation of training and test sets. Before building any GLMs, let's use the `createDataPartition()` function again to split the data into the training (75%) and test (25%) sets, as we do in CHUNK 10.

```
# CHUNK 10
library(caret)
set.seed(4769)
partition <- createDataPartition(y = vehins$clm, p = .75, list = FALSE)
train <- vehins[partition, ]
test <- vehins[-partition, ]
mean(train$clm)
mean(test$clm)

## [1] 0.06870784
## [1] 0.06631268
```

To check that the two sets are representative, we observe that the claim occurrence rate in the training set is 6.87% and that in the test set is 6.63%. That the two rates are very similar shows that the built-in stratification of the target variable works well.

We then fit separately the logistic regression model and probit regression model to the training set. The default choice of the link function for the binomial distribution is the logit link, so it does not have to be specified, but for the probit model the command `link = "probit"` has to be added as an extra argument. Because we are specifically asked not to use the `exposure` variable, we use the formula `clm ~ . - exposure + log_veh_value:veh_body`, meaning to regress `clm` on all variables in the training set except for `exposure`, with the interaction terms between `log_veh_value` and `veh_body` added (don't miss the interaction found in Task 5!).

```
# CHUNK 11
logit.full <- glm(clm ~ . - exposure + log_veh_value:veh_body,
                  data = train, family = binomial)
probit.full <- glm(clm ~ . - exposure + log_veh_value:veh_body,
                   data = train, family = binomial(link = "probit"))
```

Which link function to use? The most natural way to compare these two GLMs is to look at their predictive performance on the *test* set. As we are explicitly told not to do so in the statement of the task, we instead look at measures computed on the *training* set that assesses not only the goodness of fit of a GLM, but also the complexity of the model and strikes a balance between the two sorts of model quality. The deviance is generally not a good measure as it focuses solely on the goodness of fit of the two GLMs on the training set. A good metric is the AIC,[x] which penalizes a

[x]In fact, because the logistic and probit models share the same number of parameters, the AIC is comparing their goodness of fit to the training data captured by the (training) loglikelihood l. You can also compare the two models with respect to their deviance, as long as you note that the two models have the same degree of model complexity.

GLM by the number of parameters it carries (you can also use the BIC). Recall that the lower the AIC, the better the model.

In the second part of CHUNK 11, we use the `AIC()` function to return the AIC of each of the two GLMs.

```
# CHUNK 11 (Cont.)
AIC(logit.full)
AIC(probit.full)

## [1] 25348.85
## [1] 25348.12
```

The two AIC values are very close to each other, but the logistic regression model is much more interpretable than the probit model, so we will use the logit link in subsequent tasks.

Constructing a confusion matrix. To get a feel for how well the logistic regression model is doing, we can construct a confusion matrix (on the training set) corresponding to a certain cutoff. In R, a confusion matrix can be aptly created by the `confusionMatrix()` function from the `caret` package. This function takes a vector of predicted classes and a vector of observed classes as the first two arguments (if you are doubtful about which vector should come first, type `?confusionMatrix`) and returns the confusion matrix. In CHUNK 12, we generate the confusion matrix of the logistic regression model on the training set using a cutoff of 0.1 (which is arbitrarily set; feel free to change its value).

```
# CHUNK 12
cutoff <- 0.1  # you can try other values
# Generate predicted probabilities
pred <- predict(logit.full, type = "response")
# Turn predicted probabilities into predicted classes
class <- 1*(pred > cutoff) # OR ifelse(pred > cutoff, 1, 0)
confusionMatrix(factor(class), factor(train$clm), positive = "1")

## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 46709  3405
##          1   650    89
##
##                Accuracy : 0.9203
##                  95% CI : (0.9179, 0.9226)
##     No Information Rate : 0.9313
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.0185
##
##  Mcnemar's Test P-Value : <2e-16
```

```
##
##             Sensitivity : 0.02547
##             Specificity : 0.98628
##          Pos Pred Value : 0.12043
##          Neg Pred Value : 0.93205
##              Prevalence : 0.06871
##          Detection Rate : 0.00175
##    Detection Prevalence : 0.01453
##       Balanced Accuracy : 0.50587
##
##       'Positive' Class : 1
##
```

It is important to note that the first two arguments of the `confusionMatrix()` function must be factors, hence the use of the `factor()` function. It is also a good idea to explicitly specify the factor level that corresponds to a "positive" result (which is `"1"` in this case) via the `positive` argument. Otherwise, R will take the alpha-numerically first level (which is `"0"`) as the level of interest, in which case the sensitivity and specificity returned by R are, in fact, the specificity and sensitivity we are interested in.

We can see that while the logistic regression model is good at identifying policies without a claim (indicated by a high specificity), it is not sensitive enough to those with a claim (indicated by a low sensitivity). This is partly a result of the fact that most policies do not lead to a claim.

TASK 7: Select features

Some of the features may lack predictive power and result in overfitting. Determine which features should be retained. Use the stepAIC function (from the MASS package) to make this determination. For factor variables this function either retains the variable with its existing levels or removes the variable entirely. It does not allow for the possibility that individual factor levels may be insignificant with regard to the base level (and hence could be combined with it) or insignificantly different from other level(s) (in which case they could be combined into a new level).

Simplify the model by combining factor levels as appropriate. Use an approach that relies on hypothesis tests or information criteria. Do not use a regularization method. Be sure to explain your methodology and why it is a reasonable approach.

When finished with this task, this will be your final model form.

Now that we have decided on the type of GLM (logistic regression model) to use, it is time to simplify the model by retaining only those with strong predictive power and prevent overfitting. In CHUNK 13, we produce a summary of the fitted logistic regression model.

```
# CHUNK 13
summary(logit.full)

##
```

```
## Call:
## glm(formula = clm ~ . - exposure + log_veh_value:veh_body, family = binomial,
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1054  -0.3962  -0.3710  -0.3442   2.5928
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -2.666221   0.040753 -65.425  < 2e-16 ***
## veh_bodyBUS                    1.893682   0.614643   3.081  0.00206 **
## veh_bodyCONVT                  0.344542   1.097079   0.314  0.75348
## veh_bodyMCARA                 -0.676286   1.037468  -0.652  0.51449
## veh_age1                      -0.126600   0.056819  -2.228  0.02587 *
## veh_age2                       0.063206   0.048253   1.310  0.19023
## veh_age4                      -0.008732   0.052120  -0.168  0.86695
## genderM                       -0.036598   0.036311  -1.008  0.31350
## areaOTHER                      0.140962   0.039342   3.583  0.00034 ***
## agecat1                        0.240163   0.057624   4.168 3.08e-05 ***
## agecat3                       -0.236353   0.044210  -5.346 8.98e-08 ***
## log_veh_value                  0.184019   0.038525   4.777 1.78e-06 ***
## veh_bodyBUS:log_veh_value     -2.120656   1.262239  -1.680  0.09294 .
## veh_bodyCONVT:log_veh_value   -0.580723   0.642281  -0.904  0.36591
## veh_bodyMCARA:log_veh_value    1.135352   0.860314   1.320  0.18694
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 25455  on 50852  degrees of freedom
## Residual deviance: 25319  on 50838  degrees of freedom
## AIC: 25349
##
## Number of Fisher Scoring iterations: 5
```

The model currently has a total of 14 features (excluding the intercept), many of which are dummy variables representing the categorical predictors. The model summary shows that only a few features are statistically significant and so feature selection will likely be meaningful.

Before we use a function like `stepAIC()` (which works well on GLMs, not just on linear models) to perform stepwise selection on the fitted logistic regression model, it is a desirable activity to explicitly binarize the categorical predictors with three or more levels, as noted in Subsection 3.4.2. This allows the feature selection function to view each level of a multi-level categorical predictor separately and remove one level at a time, effectively combining that level with the baseline level, instead of removing all of the dummy variables of the categorical predictor entirely and losing too much information. As we can see from the summary output in CHUNK 13, only one level of each of `veh_body` (`"BUS"`) and `veh_age` (`"1"`) is significant with respect to the baseline, so it may be

sensible to combine the insignificant levels with the baseline and use only the significant levels for prediction. Explicit binarization will make this possible.

In CHUNK 14, we use the `dummyVars()` function from the `caret` package we first saw in Subsection 3.4.2 to binarize the categorical predictors with three or more levels, which include `veh_body`, `agecat`, `veh_age`. Because we plan to factor in the interaction between `log_veh_value` and `veh_body`, we include the term `log_veh_value * veh_body` in the formula argument of the `dummyVars()` function.

```
# CHUNK 14
library(caret)
binarizer <- dummyVars(~ log_veh_value * veh_body + agecat + veh_age,
                       data = vehins,
                       fullRank = TRUE)
binarized.vars <- data.frame(predict(binarizer, newdata = vehins))
head(binarized.vars)

##   log_veh_value veh_body.BUS veh_body.CONVT veh_body.MCARA agecat.1
## 1    0.05826891            0              0              0         0
## 2    0.02955880            0              0              0         0
## 3    1.18172720            0              0              0         0
## 4    1.42069579            0              0              0         0
## 5   -0.32850407            0              0              0         0
## 6    0.69813472            0              0              0         0
##   agecat.3 veh_age.1 veh_age.2 veh_age.4 log_veh_value.veh_bodyBUS
## 1        0         0         0         0                         0
## 2        0         0         1         0                         0
## 3        0         0         1         0                         0
## 4        0         0         1         0                         0
## 5        0         0         0         1                         0
## 6        0         0         0         0                         0
##   log_veh_value.veh_bodyCONVT log_veh_value.veh_bodyMCARA
## 1                           0                           0
## 2                           0                           0
## 3                           0                           0
## 4                           0                           0
## 5                           0                           0
## 6                           0                           0
```

Then we attach the binarized variables to the `vehins` dataset (with a `.bin` suffix) and delete the original categorical variables. We also drop one of the two `log_veh_value` variables—an extra copy is created when the interaction terms are produced by the `dummyVars()` function.

```
# CHUNK 14 (Cont.)
vehins.bin <- cbind(vehins, binarized.vars)  # attach the binarized variables
# remove the original factor variables
vehins.bin$veh_age <- NULL
vehins.bin$agecat <- NULL
```

```
vehins.bin$veh_body <- NULL
vehins.bin$log_veh_value <- NULL
head(vehins.bin)
```

```
##   clm  exposure gender area log_veh_value veh_body.BUS veh_body.CONVT
## 1   0 0.3039014      F BASE    0.05826891            0              0
## 2   0 0.6488706      F BASE    0.02955880            0              0
## 3   0 0.5694730      F BASE    1.18172720            0              0
## 4   0 0.3175907      F BASE    1.42069579            0              0
## 5   0 0.6488706      F BASE   -0.32850407            0              0
## 6   0 0.8542094      M BASE    0.69813472            0              0
##   veh_body.MCARA agecat.1 agecat.3 veh_age.1 veh_age.2 veh_age.4
## 1              0        0        0         0         0         0
## 2              0        0        0         0         1         0
## 3              0        0        0         0         1         0
## 4              0        0        0         0         1         0
## 5              0        0        0         0         0         1
## 6              0        0        0         0         0         0
##   log_veh_value.veh_bodyBUS log_veh_value.veh_bodyCONVT
## 1                         0                           0
## 2                         0                           0
## 3                         0                           0
## 4                         0                           0
## 5                         0                           0
## 6                         0                           0
##   log_veh_value.veh_bodyMCARA
## 1                           0
## 2                           0
## 3                           0
## 4                           0
## 5                           0
## 6                           0
```

Given the binarized dataset, we reuse the `partition` vector generated in CHUNK 10 to partition it into the training and test sets. Then we fit the logistic regression model to the binarized training set and print a model summary.

```
# CHUNK 15
# Set up the binarized training and test sets
train <- vehins.bin[partition, ]
test <- vehins.bin[-partition, ]

# Fit the logistic regression model to the binarized training set
logit.full <- glm(clm ~ . - exposure, data = train, family = binomial)
summary(logit.full)

##
```

```
## Call:
## glm(formula = clm ~ . - exposure, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1054  -0.3962  -0.3710  -0.3442   2.5928
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -2.666221   0.040753 -65.425  < 2e-16 ***
## genderM                       -0.036598   0.036311  -1.008  0.31350
## areaOTHER                      0.140962   0.039342   3.583  0.00034 ***
## log_veh_value                  0.184019   0.038525   4.777 1.78e-06 ***
## veh_body.BUS                   1.893682   0.614643   3.081  0.00206 **
## veh_body.CONVT                 0.344542   1.097079   0.314  0.75348
## veh_body.MCARA                -0.676286   1.037468  -0.652  0.51449
## agecat.1                       0.240163   0.057624   4.168 3.08e-05 ***
## agecat.3                      -0.236353   0.044210  -5.346 8.98e-08 ***
## veh_age.1                     -0.126600   0.056819  -2.228  0.02587 *
## veh_age.2                      0.063206   0.048253   1.310  0.19023
## veh_age.4                     -0.008732   0.052120  -0.168  0.86695
## log_veh_value.veh_bodyBUS     -2.120656   1.262239  -1.680  0.09294 .
## log_veh_value.veh_bodyCONVT   -0.580723   0.642281  -0.904  0.36591
## log_veh_value.veh_bodyMCARA    1.135352   0.860314   1.320  0.18694
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 25455  on 50852  degrees of freedom
## Residual deviance: 25319  on 50838  degrees of freedom
## AIC: 25349
##
## Number of Fisher Scoring iterations: 5
```

The summary output is identical to that in CHUNK 13, showing that the binarization is done properly and does not have any effect on the model, but the automatic stepwise selection procedure we are going to carry out will recognize each level of the categorical predictors as a separate feature.

In CHUNK 16, we run the `stepAIC()` function on the full logistic regression model. The task does not say anything about the options of the `stepAIC()` function you can set (the focus of this task, like Task 7 of the Hospital Readmissions sample project, is on binarization, not the `stepAIC()` function), so you are at liberty to perform either the backward selection (default) or forward selection, using the AIC (default) or BIC as the selection criterion. On the exam, you will probably be asked to do only one round of stepwise model selection using a particular selection process and selection criterion, and justify your choices with reference to the business problem (e.g., the BIC represents a more conservative approach to feature selection and allows the insurance company to identify key factors affecting claim occurrence rates). Considerations discussed on page

206 will be of use. Here for the purposes of illustration, we will do two rounds of backward selection, one using the AIC and the other using the BIC, and compare the two reduced models. It may take a minute or two to execute the stepwise selection algorithm due to the rather large number of dummy variables, each of which is treated as a separate feature to remove.

```
# CHUNK 16
library(MASS)
logit.AIC <- stepAIC(logit.full)

## Start:  AIC=25348.85
## clm ~ (exposure + gender + area + log_veh_value + veh_body.BUS +
##     veh_body.CONVT + veh_body.MCARA + agecat.1 + agecat.3 + veh_age.1 +
##     veh_age.2 + veh_age.4 + log_veh_value.veh_bodyBUS +
##     log_veh_value.veh_bodyCONVT + log_veh_value.veh_bodyMCARA) - exposure
##
##                                Df Deviance   AIC
## - veh_age.4                     1    25319 25347
## - veh_body.CONVT                1    25319 25347
## - veh_body.MCARA                1    25319 25347
## - log_veh_value.veh_bodyCONVT   1    25320 25348
## - gender                        1    25320 25348
## - veh_age.2                     1    25321 25349
## - log_veh_value.veh_bodyMCARA   1    25321 25349
## <none>                               25319 25349
## - log_veh_value.veh_bodyBUS     1    25322 25350
## - veh_age.1                     1    25324 25352
## - veh_body.BUS                  1    25326 25354
## - area                          1    25332 25360
## - agecat.1                      1    25335 25363
## - log_veh_value                 1    25342 25370
## - agecat.3                      1    25348 25376
##
## Step:  AIC=25346.87
## clm ~ gender + area + log_veh_value + veh_body.BUS + veh_body.CONVT +
##     veh_body.MCARA + agecat.1 + agecat.3 + veh_age.1 + veh_age.2 +
##     log_veh_value.veh_bodyBUS + log_veh_value.veh_bodyCONVT +
##     log_veh_value.veh_bodyMCARA
##
##                                Df Deviance   AIC
## - veh_body.CONVT                1    25319 25345
## - veh_body.MCARA                1    25319 25345
## - log_veh_value.veh_bodyCONVT   1    25320 25346
## - gender                        1    25320 25346
## - log_veh_value.veh_bodyMCARA   1    25321 25347
## <none>                               25319 25347
## - veh_age.2                     1    25321 25347
## - log_veh_value.veh_bodyBUS     1    25322 25348
```

```
## - veh_age.1                        1    25324 25350
## - veh_body.BUS                     1    25326 25352
## - area                             1    25332 25358
## - agecat.1                         1    25336 25362
## - log_veh_value                    1    25347 25373
## - agecat.3                         1    25348 25374
##
## Step:  AIC=25344.97
## clm ~ gender + area + log_veh_value + veh_body.BUS + veh_body.MCARA +
##     agecat.1 + agecat.3 + veh_age.1 + veh_age.2 + log_veh_value.veh_bodyBUS +
##     log_veh_value.veh_bodyCONVT + log_veh_value.veh_bodyMCARA
##
##                                Df Deviance   AIC
## - veh_body.MCARA                1    25319 25343
## - gender                        1    25320 25344
## - log_veh_value.veh_bodyMCARA   1    25321 25345
## - log_veh_value.veh_bodyCONVT   1    25321 25345
## <none>                               25319 25345
## - veh_age.2                     1    25321 25345
## - log_veh_value.veh_bodyBUS     1    25322 25346
## - veh_age.1                     1    25324 25348
## - veh_body.BUS                  1    25326 25350
## - area                          1    25332 25356
## - agecat.1                      1    25336 25360
## - log_veh_value                 1    25347 25371
## - agecat.3                      1    25349 25373
##
## Step:  AIC=25343.43
## clm ~ gender + area + log_veh_value + veh_body.BUS + agecat.1 +
##     agecat.3 + veh_age.1 + veh_age.2 + log_veh_value.veh_bodyBUS +
##     log_veh_value.veh_bodyCONVT + log_veh_value.veh_bodyMCARA
##
##                                Df Deviance   AIC
## - gender                        1    25321 25343
## - log_veh_value.veh_bodyCONVT   1    25321 25343
## <none>                               25319 25343
## - veh_age.2                     1    25322 25344
## - log_veh_value.veh_bodyBUS     1    25323 25345
## - log_veh_value.veh_bodyMCARA   1    25323 25345
## - veh_age.1                     1    25325 25347
## - veh_body.BUS                  1    25327 25349
## - area                          1    25332 25354
## - agecat.1                      1    25336 25358
## - log_veh_value                 1    25348 25370
## - agecat.3                      1    25349 25371
##
## Step:  AIC=25342.52
```

```
## clm ~ area + log_veh_value + veh_body.BUS + agecat.1 + agecat.3 +
##     veh_age.1 + veh_age.2 + log_veh_value.veh_bodyBUS +
##     log_veh_value.veh_bodyCONVT +
##     log_veh_value.veh_bodyMCARA
##
##                                Df Deviance   AIC
## - log_veh_value.veh_bodyCONVT  1    25322 25342
## <none>                              25321 25343
## - veh_age.2                    1    25323 25343
## - log_veh_value.veh_bodyBUS    1    25324 25344
## - log_veh_value.veh_bodyMCARA  1    25324 25344
## - veh_age.1                    1    25325 25345
## - veh_body.BUS                 1    25328 25348
## - area                         1    25333 25353
## - agecat.1                     1    25337 25357
## - log_veh_value                1    25348 25368
## - agecat.3                     1    25351 25371
##
## Step:  AIC=25342.41
## clm ~ area + log_veh_value + veh_body.BUS + agecat.1 + agecat.3 +
##     veh_age.1 + veh_age.2 + log_veh_value.veh_bodyBUS +
##     log_veh_value.veh_bodyMCARA
##                                Df Deviance   AIC
## <none>                              25322 25342
## - veh_age.2                    1    25325 25343
## - log_veh_value.veh_bodyBUS    1    25326 25344
## - log_veh_value.veh_bodyMCARA  1    25326 25344
## - veh_age.1                    1    25327 25345
## - veh_body.BUS                 1    25330 25348
## - area                         1    25335 25353
## - agecat.1                     1    25339 25357
## - log_veh_value                1    25348 25366
## - agecat.3                     1    25353 25371

summary(logit.AIC)

##
## Call:
## glm(formula = clm ~ area + log_veh_value + veh_body.BUS + agecat.1 +
##     agecat.3 + veh_age.1 + veh_age.2 + log_veh_value.veh_bodyBUS +
##     log_veh_value.veh_bodyMCARA, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0936  -0.3957  -0.3710  -0.3448   2.5219
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)                  -2.68557    0.02922 -91.906  < 2e-16 ***
## areaOTHER                     0.14069    0.03933   3.577 0.000348 ***
## log_veh_value                 0.17631    0.03457   5.101 3.38e-07 ***
## veh_body.BUS                  1.87999    0.61420   3.061 0.002207 **
## agecat.1                      0.23894    0.05758   4.150 3.33e-05 ***
## agecat.3                     -0.23955    0.04409  -5.433 5.54e-08 ***
## veh_age.1                    -0.11617    0.05539  -2.097 0.035953 *
## veh_age.2                     0.07383    0.04569   1.616 0.106107
## log_veh_value.veh_bodyBUS    -2.09924    1.26072  -1.665 0.095889 .
## log_veh_value.veh_bodyMCARA   0.59658    0.28817   2.070 0.038430 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 25455  on 50852  degrees of freedom
## Residual deviance: 25322  on 50843  degrees of freedom
## AIC: 25342
##
## Number of Fisher Scoring iterations: 5

logit.BIC <- stepAIC(logit.full, k = log(nrow(train)))

## Start:  AIC=25481.4
## clm ~ (exposure + gender + area + log_veh_value + veh_body.BUS +
##     veh_body.CONVT + veh_body.MCARA + agecat.1 + agecat.3 + veh_age.1 +
##     veh_age.2 + veh_age.4 + log_veh_value.veh_bodyBUS +
##     log_veh_value.veh_bodyCONVT + log_veh_value.veh_bodyMCARA) - exposure
##
##                                Df Deviance   AIC
## - veh_age.4                     1    25319 25471
## - veh_body.CONVT                1    25319 25471
## - veh_body.MCARA                1    25319 25471
## - log_veh_value.veh_bodyCONVT   1    25320 25471
## - gender                        1    25320 25472
## - veh_age.2                     1    25321 25472
## - log_veh_value.veh_bodyMCARA   1    25321 25472
## - log_veh_value.veh_bodyBUS     1    25322 25474
## - veh_age.1                     1    25324 25476
## - veh_body.BUS                  1    25326 25478
## <none>                               25319 25481
## - area                          1    25332 25483
## - agecat.1                      1    25335 25487
## - log_veh_value                 1    25342 25493
## - agecat.3                      1    25348 25500
##
## Step:  AIC=25470.59
```

```
## clm ~ gender + area + log_veh_value + veh_body.BUS + veh_body.CONVT +
##     veh_body.MCARA + agecat.1 + agecat.3 + veh_age.1 + veh_age.2 +
##     log_veh_value.veh_bodyBUS + log_veh_value.veh_bodyCONVT +
##     log_veh_value.veh_bodyMCARA
##
##                                Df Deviance   AIC
## - veh_body.CONVT                1    25319 25460
## - veh_body.MCARA                1    25319 25460
## - log_veh_value.veh_bodyCONVT   1    25320 25461
## - gender                        1    25320 25461
## - log_veh_value.veh_bodyMCARA   1    25321 25462
## - veh_age.2                     1    25321 25462
## - log_veh_value.veh_bodyBUS     1    25322 25463
## - veh_age.1                     1    25324 25465
## - veh_body.BUS                  1    25326 25467
## <none>                               25319 25471
## - area                          1    25332 25472
## - agecat.1                      1    25336 25476
## - log_veh_value                 1    25347 25488
## - agecat.3                      1    25348 25489
##
## Step:  AIC=25459.84
## clm ~ gender + area + log_veh_value + veh_body.BUS + veh_body.MCARA +
##     agecat.1 + agecat.3 + veh_age.1 + veh_age.2 + log_veh_value.veh_bodyBUS +
##     log_veh_value.veh_bodyCONVT + log_veh_value.veh_bodyMCARA
##
##                                Df Deviance   AIC
## - veh_body.MCARA                1    25319 25450
## - gender                        1    25320 25450
## - log_veh_value.veh_bodyMCARA   1    25321 25451
## - log_veh_value.veh_bodyCONVT   1    25321 25451
## - veh_age.2                     1    25321 25451
## - log_veh_value.veh_bodyBUS     1    25322 25453
## - veh_age.1                     1    25324 25454
## - veh_body.BUS                  1    25326 25456
## <none>                               25319 25460
## - area                          1    25332 25462
## - agecat.1                      1    25336 25466
## - log_veh_value                 1    25347 25477
## - agecat.3                      1    25349 25479
##
## Step:  AIC=25449.47
## clm ~ gender + area + log_veh_value + veh_body.BUS + agecat.1 +
##     agecat.3 + veh_age.1 + veh_age.2 + log_veh_value.veh_bodyBUS +
##     log_veh_value.veh_bodyCONVT + log_veh_value.veh_bodyMCARA
##
##                                Df Deviance   AIC
```

```
## - gender                          1    25321 25440
## - log_veh_value.veh_bodyCONVT     1    25321 25441
## - veh_age.2                       1    25322 25441
## - log_veh_value.veh_bodyBUS       1    25323 25442
## - log_veh_value.veh_bodyMCARA     1    25323 25442
## - veh_age.1                       1    25325 25444
## - veh_body.BUS                    1    25327 25446
## <none>                                 25319 25450
## - area                            1    25332 25451
## - agecat.1                        1    25336 25455
## - log_veh_value                   1    25348 25467
## - agecat.3                        1    25349 25468
##
## Step:  AIC=25439.72
## clm ~ area + log_veh_value + veh_body.BUS + agecat.1 + agecat.3 +
##     veh_age.1 + veh_age.2 + log_veh_value.veh_bodyBUS +
##     log_veh_value.veh_bodyCONVT +
##     log_veh_value.veh_bodyMCARA
##
##                                Df Deviance   AIC
## - log_veh_value.veh_bodyCONVT   1    25322 25431
## - veh_age.2                     1    25323 25431
## - log_veh_value.veh_bodyBUS     1    25324 25432
## - log_veh_value.veh_bodyMCARA   1    25324 25432
## - veh_age.1                     1    25325 25434
## - veh_body.BUS                  1    25328 25436
## <none>                               25321 25440
## - area                          1    25333 25441
## - agecat.1                      1    25337 25445
## - log_veh_value                 1    25348 25456
## - agecat.3                      1    25351 25459
##
## Step:  AIC=25430.77
## clm ~ area + log_veh_value + veh_body.BUS + agecat.1 + agecat.3 +
##     veh_age.1 + veh_age.2 + log_veh_value.veh_bodyBUS +
##     log_veh_value.veh_bodyMCARA
##
##                                Df Deviance   AIC
## - veh_age.2                     1    25325 25423
## - log_veh_value.veh_bodyBUS     1    25326 25423
## - log_veh_value.veh_bodyMCARA   1    25326 25424
## - veh_age.1                     1    25327 25424
## - veh_body.BUS                  1    25330 25427
## <none>                               25322 25431
## - area                          1    25335 25433
## - agecat.1                      1    25339 25436
## - log_veh_value                 1    25348 25446
## - agecat.3                      1    25353 25451
##
```

```
## Step:  AIC=25422.54
## clm ~ area + log_veh_value + veh_body.BUS + agecat.1 + agecat.3 +
##     veh_age.1 + log_veh_value.veh_bodyBUS + log_veh_value.veh_bodyMCARA
##
##                                Df Deviance   AIC
## - log_veh_value.veh_bodyMCARA  1    25328 25415
## - log_veh_value.veh_bodyBUS    1    25328 25415
## - veh_body.BUS                 1    25332 25419
## - veh_age.1                    1    25335 25422
## <none>                              25325 25423
## - area                         1    25337 25424
## - agecat.1                     1    25342 25429
## - agecat.3                     1    25355 25442
## - log_veh_value                1    25367 25454
##
## Step:  AIC=25415.04
## clm ~ area + log_veh_value + veh_body.BUS + agecat.1 + agecat.3 +
##     veh_age.1 + log_veh_value.veh_bodyBUS
##
##                              Df Deviance   AIC
## - log_veh_value.veh_bodyBUS  1    25332 25408
## - veh_body.BUS               1    25335 25411
## - veh_age.1                  1    25339 25415
## <none>                            25328 25415
## - area                       1    25341 25417
## - agecat.1                   1    25345 25421
## - agecat.3                   1    25358 25434
## - log_veh_value              1    25372 25448
##
## Step:  AIC=25407.62
## clm ~ area + log_veh_value + veh_body.BUS + agecat.1 + agecat.3 +
##     veh_age.1
##
##                 Df Deviance   AIC
## - veh_body.BUS   1    25336 25401
## - veh_age.1      1    25342 25407
## <none>                25332 25408
## - area           1    25344 25409
## - agecat.1       1    25349 25414
## - agecat.3       1    25361 25426
## - log_veh_value  1    25375 25440
##
## Step:  AIC=25400.48
## clm ~ area + log_veh_value + agecat.1 + agecat.3 + veh_age.1
##
##                 Df Deviance   AIC
## - veh_age.1      1    25346 25400
```

```
## <none>                  25336 25401
## - area           1      25348 25402
## - agecat.1       1      25353 25407
## - agecat.3       1      25365 25419
## - log_veh_value  1      25379 25433
##
## Step:  AIC=25400.09
## clm ~ area + log_veh_value + agecat.1 + agecat.3
##
##                 Df Deviance   AIC
## <none>                  25346 25400
## - area           1      25359 25402
## - agecat.1       1      25362 25405
## - agecat.3       1      25377 25420
## - log_veh_value  1      25379 25423

summary(logit.BIC)

##
## Call:
## glm(formula = clm ~ area + log_veh_value + agecat.1 + agecat.3,
##     family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.5332  -0.3957  -0.3730  -0.3442   2.5141
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.67961    0.02730 -98.165  < 2e-16 ***
## areaOTHER     0.14087    0.03931   3.584 0.000339 ***
## log_veh_value 0.16440    0.02859   5.750 8.90e-09 ***
## agecat.1      0.23614    0.05747   4.109 3.98e-05 ***
## agecat.3     -0.23941    0.04398  -5.443 5.24e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 25455  on 50852  degrees of freedom
## Residual deviance: 25346  on 50848  degrees of freedom
## AIC: 25356
##
## Number of Fisher Scoring iterations: 5
```

The final model based on the AIC uses nine features:

areaOTHER, log_veh_value, veh_body.BUS, agecat.1, agecat.3, veh_age.1, veh_age.2,

`log_veh_value.veh_bodyBUS`, `log_veh_value.veh_bodyMCARA`.

In contrast, the final model based on the BIC carries only four features:

`areaOTHER`, `log_veh_value`, `agecat.1`, `agecat.3`.

In particular, the interaction variables are dropped out of the BIC model. The fact that the BIC model is smaller in size comes as no surprise: The penalty term of the BIC, which is $\ln 50,853 = 10.8367$, is much larger than that of the AIC, which is 2. The BIC therefore favors simpler models and retains only extremely significant features. As we can see in the summary output of the BIC model, all of the four retained features are statistically significant, whereas a few of the retained features in the AIC model are only mildly significant. In the remaining tasks, we will adopt the BIC model.

TASK 8: Exposure-adjusted predictions

The exposure variable measures the effective duration of each vehicle insurance policy over the one-year policy period.

Use the model selected in TASK 7 to make predictions on the test set assuming that exposure affects claim occurrence probability multiplicatively. Calculate the AUC of the model on the test set.

So far all of our GLMs ignore the `exposure` variable (as requested) and implicitly assume that each vehicle insurance policyholder is covered for the entire period (i.e., one year). In reality, some of the policies came into force partly into the year and some were canceled by the end of the policy year. The `exposure` variable records the effective duration of each vehicle insurance policy in the dataset and captures the amount of risk each policy is exposed to. One way to take advantage of this variable is to assume that the claim occurrence probability is proportionally reduced by the amount of `exposure`.[xi] Mathematically, we assume that the exposure-adjusted claim occurrence probability π^* is related to the one-year claim occurrence probability π via $\pi^* = t\pi$ and we use the logistic regression model above to get the predicted one-year claim occurrence probability $\hat{\pi}$. Then the exposure-adjusted predicted claim occurrence probability is given by $\widehat{\pi^*} = t\hat{\pi}$.

In CHUNK 17, we first re-scale the predictions of the BIC-based logistic regression model by multiplying the unadjusted predictions by the amount of exposure of different test observations. Then we use the `roc()` and `auc()` functions from the `pROC` package (available on the exam) to compute the AUC of the model on the test set. The `roc()` function takes the observed levels of the target variable and the predicted probabilities supplied in order and produces a list, which can be plotted or passed to the `auc()` function to return the test AUC of the model. For completeness, we also compute the test AUC of the full logistic regression model and the reduced model based on the AIC produced in CHUNK 16.

```
# CHUNK 17
library(pROC)
pred.full <- test$exposure * predict(logit.full, newdata = test, type = "response")
roc.full <- roc(test$clm, pred.full)
auc(roc.full)
```

[xi] This way of doing exposure adjustment differs from the use of offset and is motivated by Section 7.4 of *Generalized Linear Models for Insurance Data*, by P. de Jong and G. Z. Heller.

```
## Area under the curve: 0.6589

pred.AIC <- test$exposure * predict(logit.AIC, newdata = test, type = "response")
roc.AIC <- roc(test$clm, pred.AIC)
auc(roc.AIC)

## Area under the curve: 0.6587

pred.BIC <- test$exposure * predict(logit.BIC, newdata = test, type = "response")
roc.BIC <- roc(test$clm, pred.BIC)
auc(roc.BIC)

## Area under the curve: 0.6592
```

It is reassuring to see that the AUC of the BIC-based model is the highest among the three models despite having far fewer features and therefore using much less information.[xii] By the way, if you omit the exposure adjustment, then the AUC of all of the three models will be only marginally higher than 0.5 (Try it!). This shows how important the exposure adjustment is.

TASK 9: Interpret the model

Run the selected model from Task 7 on the full dataset and provide the output. Interpret the results in a manner that will provide useful information to the insurance company.

To round up our model development procedure, we refit the BIC-based logistic regression model to the full dataset and save it as a `glm` object named `logit.final`. This refitting is done because in a real-world application we plan to eventually use the logistic regression model on completely new data that will arrive in the future, making it desirable to estimate the parameters using all the available data. For moderately sized datasets, this refitting may give rise to a meaningful improvement in the accuracy of the parameter estimates with the use of extra observations. The results are in CHUNK 18 (note the argument `data = vehins.bin`, not `data = train`).

```
# CHUNK 18
logit.final <- glm(clm ~ area + log_veh_value + agecat.1 + agecat.3,
                   data = vehins.bin, family = binomial)
summary(logit.final)

##
## Call:
## glm(formula = clm ~ area + log_veh_value + agecat.1 + agecat.3,
##     family = binomial, data = vehins.bin)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.5173  -0.3923  -0.3724  -0.3461   2.5019
```

[xii] If you change the random seed in CHUNK 10, you may run into a situation where the AUC of the BIC-based model is lower than that of the full model, but the difference is likely negligible.

```
##
## Coefficients:
##                Estimate Std. Error  z value Pr(>|z|)
## (Intercept)   -2.67942    0.02367 -113.200  < 2e-16 ***
## areaOTHER      0.09974    0.03449    2.892  0.00383 **
## log_veh_value  0.15962    0.02481    6.433 1.25e-10 ***
## agecat.1       0.22451    0.05055    4.441 8.93e-06 ***
## agecat.3      -0.21333    0.03786   -5.635 1.76e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 33728  on 67802  degrees of freedom
## Residual deviance: 33608  on 67798  degrees of freedom
## AIC: 33618
##
## Number of Fisher Scoring iterations: 5
```

The results of a fitted logistic regression model can be interpreted in different ways.

Odds-based interpretation. Table 4.4 provides the odds-based interpretation of the final model using the value of each estimated coefficient. The statements there are in terms of multiplicative changes in the odds of claim occurrence. Equivalently, we can also state the findings in terms of percentage changes. For example, the odds of claim occurrence is estimated to increase by $e^{0.15962} - 1 = 0.1731$ for every unit increase in the log of vehicle value, holding all other features fixed.

Probability-based interpretation. Translating the results of a logistic regression model into statements for the probability of the event of interest is harder. This is because the probability $p = e^{\eta}/(1 + e^{\eta}) = 1/(1 + e^{-\eta})$ is a nonlinear, rather complex function of the linear predictor. One interesting way to make probability-based statements, as suggested in the Hospital Readmissions sample project, is to use as the baseline the "average policyholder," who has the mean feature values or median feature levels, and examine the isolated impact of each feature on the claim occurrence probability.

In CHUNK 19, we create a data frame hosting the feature combinations we want to explore. The first row of the data frame contains the feature values of the "average policyholder," who has `area = "BASE"`, `log_veh_value = 0.38675`, and `agecat = "2"` (these choices come from the summary output in CHUNK 8). In each of the four remaining rows, we alter the value or level of one and only one feature (`area`, `log_veh_value`, `agecat`, in that order) to look at the effect of that feature on the predicted claim occurrence probability. Table 4.5 summarizes the results.

```
# CHUNK 19
new.data <- data.frame(area = c("BASE", "OTHER", "BASE", "BASE", "BASE"),
                       log_veh_value = c(0.38675, 0.38675, 0.42543,
                                         0.38675, 0.38675),
```

Feature	Coefficient Estimate	Interpretation
`area = OTHER`	0.09974	The odds of claim occurrence for policyholders living in `OTHER` areas is $e^{0.09974} = 1.1049$ times of that for those living in the baseline area. (Not much is known about the type of residence area, so it is difficult to interpret this result using common knowledge.)
`log_veh_value`	0.15962	A unit increase in the log of vehicle value is associated with a multiplicative increase of $e^{0.15962} = 1.1731$ in the odds of claim occurrence, holding all other features fixed. This makes intuitive sense because the more valuable the vehicle, the more likely a driver will submit insurance claims in the case of accidental damage.
`agecat.1`	0.22451	The odds of claim occurrence for policyholders in age category 1 is $e^{0.22451} = 1.2517$ times of that for those in age category 2 (baseline). It is possible that younger drivers are more reckless and tend to have higher accident rates.
`agecat.3`	−0.21333	The odds of claim occurrence for policyholders in age category 3 is $e^{-0.21333} = 0.8079$ times of that for those in age category 2 (baseline). This also aligns with intuition as more mature drivers are more careful and tend to have lower accident rates.

Table 4.4: Odds-based interpretation of the BIC-based logistic regression model fitted to the full dataset.

area	log_veh_value	agecat.1	agecat.3	Predicted Probability (without Exposure Adjustment)
BASE	0.38675	0	0	0.0680
OTHER	0.38675	0	0	0.0746
BASE	0.42543	0	0	0.0684
BASE	0.38675	1	0	0.0837
BASE	0.38675	0	1	0.0557

Table 4.5: Probability-based interpretation of the BIC-based logistic regression model fitted to the full dataset.

```
                    agecat.1 = c(0, 0, 0, 1, 0),
                    agecat.3 = c(0, 0, 0, 0, 1))
new.data

##    area log_veh_value agecat.1 agecat.3
## 1  BASE       0.38675        0        0
## 2 OTHER       0.38675        0        0
## 3  BASE       0.42543        0        0
## 4  BASE       0.38675        1        0
## 5  BASE       0.38675        0        1

predict(logit.final, newdata = new.data, type = "response")

##          1          2          3          4          5
## 0.06800846 0.07460946 0.06840083 0.08369446 0.05567071
```

We can see that:

- Living in `OTHER` areas compared to `BASE` areas increases the claim occurrence probability by $0.0746 - 0.0680 = 0.66\%$.
- A 10% increase (in proportion) in `log_veh_value` increases the claim occurrence probability by $0.0684 - 0.0680 = 0.04\%$.
- Being in age category 1 increases the claim occurrence probability by $0.0837 - 0.0680 = 1.57\%$ compared to being in age category 2.
- Being in age category 3 increases the claim occurrence probability by $0.0557 - 0.0680 = -1.23\%$, i.e., a decrease of 1.23%, compared to being in age category 2.

Regardless of whether our statements are based on odds or probability, overall our results quantify the directional impact of the key risk factors affecting the claim-generating mechanism and inform the insurance company as to what characteristics of its policyholders and their vehicles should enter its ratemaking plan.