Midterm Project

Jingwen Xu

2020/12/10

Abstract

In this project, I mainly set up a model to predict the possibility of customers from an insurance company that are interested in vehicle insurance offered by the company. In kaggle website, some data scientists have made EDA, feature engineering and modeling about the data. I also make EDA using stacked bar to show the proportion of binary outcome in variables, and construct a multilevel logistic regression model to further explore the relationship between predictors and outcome with model check and inference. I am pretty satisfied with the results, because the coefficients of the model line up with the data visualization about the data. In summary, most of the customers are not interested in the vehicle insurance, but young male customers with car aged 1-2 years and no other vehicle insurance can be the potential client of selling the vehicle insurance offered by the company.

Introduction

An insurance company had provided health insurance for customers, and now the company wants to predict the responses of its customers to vehicle insurance. So they need the help of data scientists to do data analysis which will be beneficial to their decision about the insurance policy, sale channel and so on.

The data is collected in customers by the company, divided into train data (more than 380,000 observations) and test data (more than 127,000 observations). The data includes the demographics information, vehicle information and insurance policy of these customers. Based on the requirement of the company, I will do exploratory data analysis and modeling (here I choose multilevel logistic regression) using train data to explore the relationship between the customers' information and their interest to vehicle insurance. And then I will make prediction with test data and compare the outcome with true values.

Method

Exploratory Data Analysis

From the data table, we can see that the data has already been tidy so I don't need to make preliminary data cleaning. Before modeling, we usually use EDA to investigate the relationship between possible predictors and outcome to make some comparison.

In this project, the outcome is binary - whether the customers are interested in the vehicle insurance. Besides, the data includes categorical variables and continuous variables as possible predictors. Here I use different functions in *ggplot* package to do data visualizations about different types of variables.

At first, according to the summary of the data, *Region_Code*, *Vintage* and *Policy_Sales_Channel* are all categorical variables with more than 50 categories. And the distribution of these categories are extremely unbalanced. So in order to make the plot more aesthetic and readable, I arrange the counts for each category and only show the binary responses' proportions in top 5 categories with stacked bar plot.

From the plot, we can obtain following information. Firstly, there are the most observations from region 28 and sales channel 152. Secondly, obviously most of the customers are not interested in the vehicle insurance for each category. Thirdly, for region 28 and sales channel 152, the customers that are interested take the

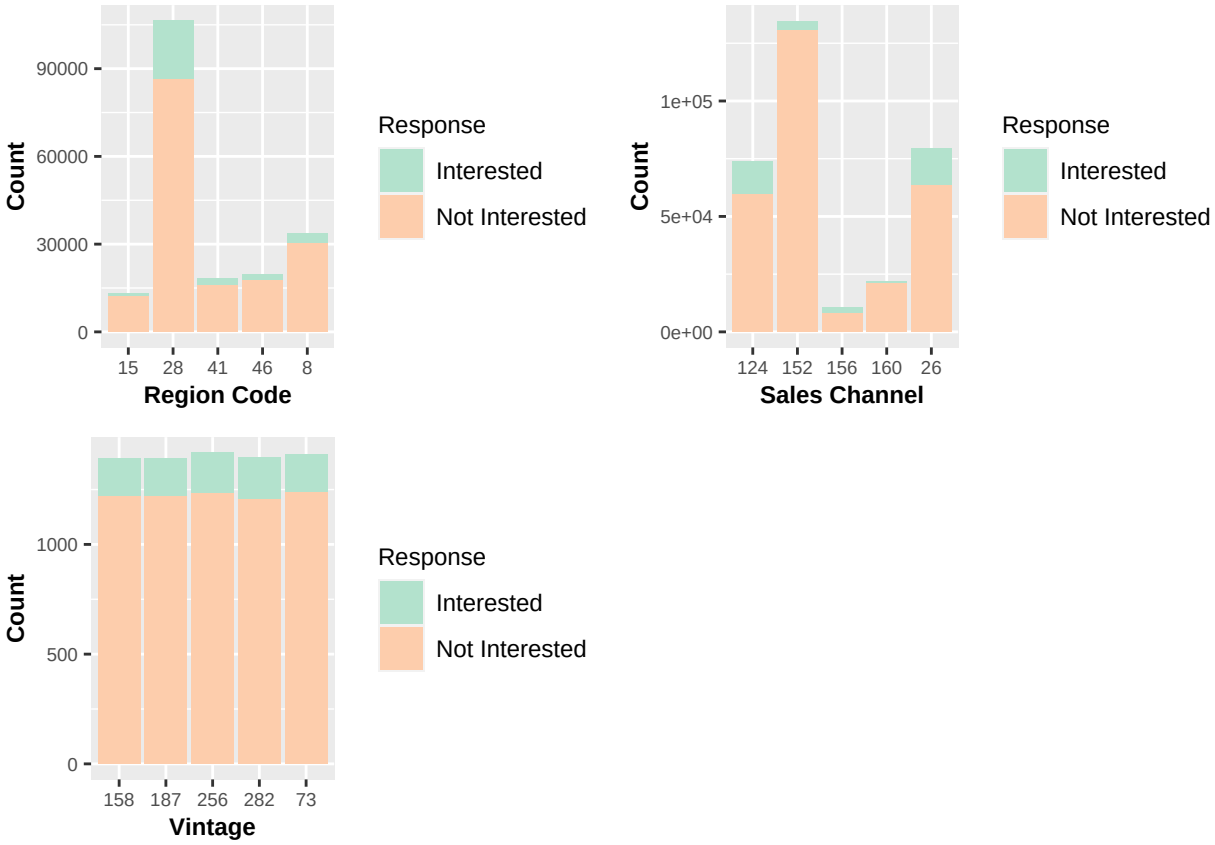


Figure 1: Proportion of responses in categorical variable(more than 50 categories)

largest proportion compared with other categories. Last, there are trivial differences between the proportion of interested for different vintage categories.

Then, for binary or ternary categorical variables *Gender*, *Driving_License*, *Previously_Insured*, *Vehicle_Age* and *Vehicle_Damage*, I also make stacked bar plot which is appropriate to display the proportions of binary responses in different categories of certain variable. This method is corresponded to the outcome (probability of interested or not interested) of multilevel logistics regression model. And last, for continuous variables *Age* and *Annual_Premium*, I can just make a scatter plot. But to show the distribution of responses and the continuous variables, I add the marginal density plots to it.

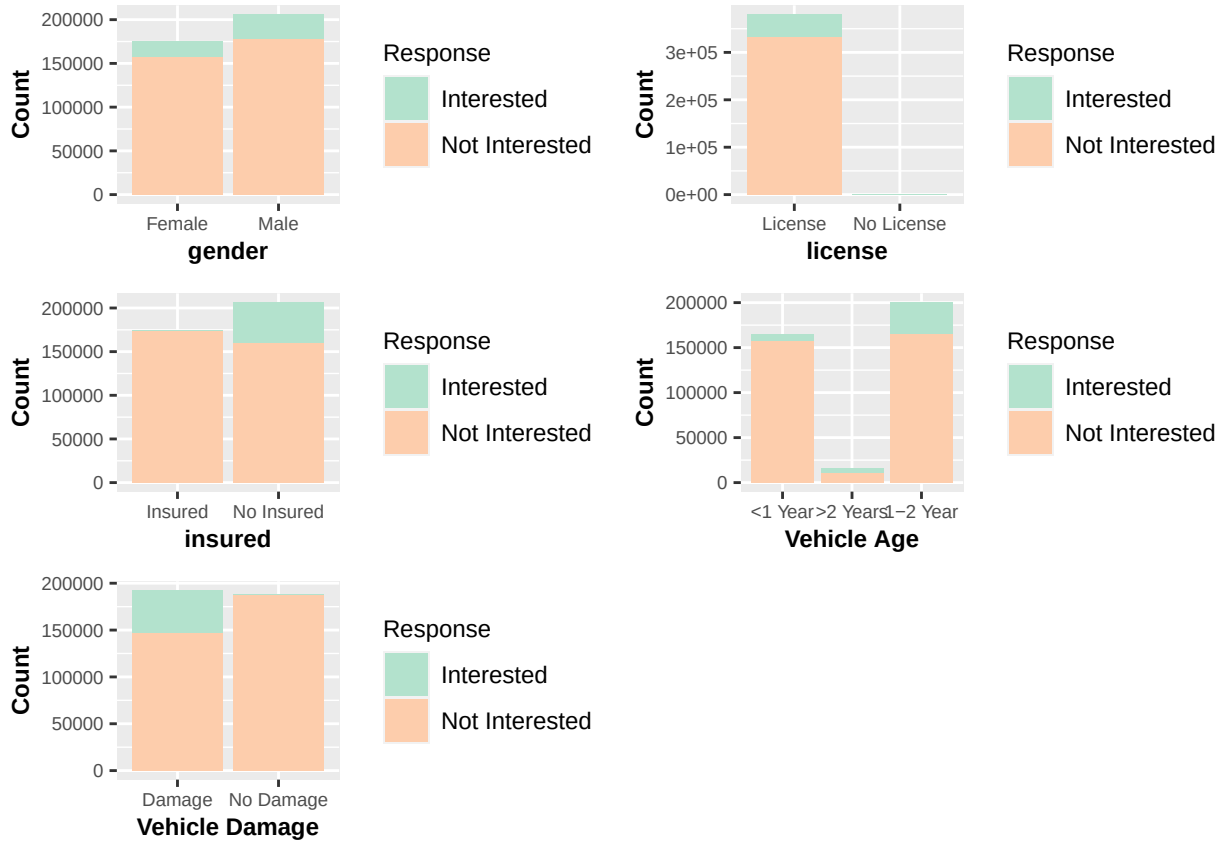


Figure 2: Proportion of responses in binary or ternary categorical variable

In the plot, we can see that customers who are not interested in vehicle insurance take a very large proportion in each category of all the variables. Respectively, for *Gender* variable, male will be more possible to give positive response to vehicle insurance which is reasonable because male usually more care about their cars. For *Driving_License* variable, the number of people without license is too small to clearly compare the proportion of response for the two categories. For *Previously_Insured* variable, people who have had vehicle insurance hardly consider to buy another one. For *Vehicle_Age* variable, people with the car aged 1-2 years will be more likely to be interested in vehicle insurance. Because the cars aged less than 1 year are nearly new that do not need to too care, and the cars aged more than 2 years may be replaced. Lastly for *Vehicle_Damage* variable, cars without damage usually don't need a vehicle insurance.

About the plot of continuous variables(it's in Appendix because the plot always shows behind a blank plot), it's obvious that zero response always take the most proportion. Besides, customers investigated are mostly young and pay a fairly low annual premium. As for the probability of response along with these two continuous variables, we need to observe through the model.

Data Processing

Before modeling, we also need to do some operations on the data such as data transformation. First of all, I transform the character categories of some categorical variables to "0, 1" or "1, 2, 3" - factor variables. And then, because of the extremely large range of annual premium, I use log transformation to re-scale it. Also for more convenience to interpret, I normalize the age variable so that we can use the mean age as the baseline. At last, I make a data table to show the transformed variables.

Gender	c_Age	Vehicle_Age	Vehicle_Damage	log_Annual_Premium
1	0.3337768	3	1	10.607921
1	2.3967476	2	0	10.420375
1	0.5271803	3	1	10.553048
1	-1.1489834	1	0	10.261826
0	-0.6332407	1	0	10.221796
0	-0.9555799	1	1	7.874739

Model

- Predictors: Referring to the data visualization in EDA part, nearly all the variables except for *Vintage* variable have obvious correspondence with the binary outcome so I choose all the variables except for *Vintage* as predictors.
- Model select: Firstly, this is undoubtedly a logistic model due to the binary outcome. Secondly, I choose *Region_Code* and *Policy_Sales_Channel* as the two group levels of multilevel model because they will be collinear with intercept and cause no pooling situation in just logistic regression.(Using *arm* package)
- Data in model: In order to make prediction using test data, I must filter the data to make sure that the train data and test data has totally the same group level.

Results

Model coefficients

Except that age and previously insured are negatively correspond to the outcome, other predictors are all positively related to the outcome. And all the fixed coefficients are significant because the estimates are more than to two standard error from zero. Besides, the negative or positive random effects are nearly equal for each of the two group levels(see the histogram in Appendix). I also calculated the confidence interval of fixed effects as following:

	X2.5..	X97.5..
.sig01	NA	NA
.sig02	NA	NA
(Intercept)	-5.4949428	-4.7516183
Gender	0.0456049	0.0899359
c_Age	-0.4016529	-0.3688140
Driving_License	0.8590931	1.5009863
Previously_Insured	-4.1079644	-3.7834934
Vehicle_Age	0.2380015	0.2966105
Vehicle_Damage	1.8874791	2.0231898
log_Annual_Premium	0.0130897	0.0376288

Model Checking/Predictions

- Binned residual plot:

From the binned residual plot, we can see that most of the observations are between the two boundary which indicates theoretical 95% error bounds that would be appropriate if the model were true. And average residuals - positive or negative ones - are pretty evenly distributed for each point of expected values. But for the expected values near to 0.5, the average residuals are abnormally low. So the model are fitted good with

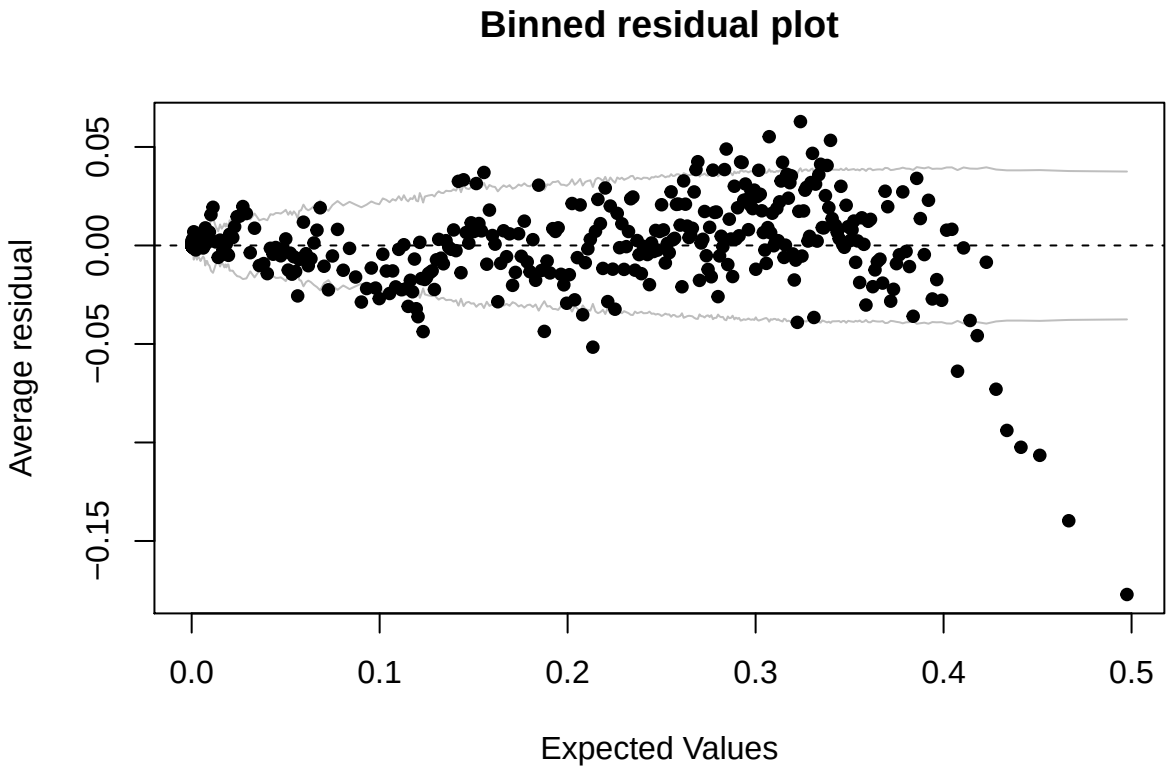


Figure 3: Binned residual plot

just several tricky problems such as the outliers. Besides, the probabilities of interested response are nearly all less than 0.5 or just 0.

- Predict using test data:

According to the distribution of predicted values in test data(the histogram of distribution is in Appendix), the possibilities of positive responses are nearly all less than 0.5 which are consistent with all zero response in the test data. So the prediction using the model is pretty good.

Discussion

The results fairly line up with what I expect that the great majority of customers will not be interested in the vehicle insurance. I recommend the insurance company to focus on young male customers with car aged 1-2 years to sell the vehicle insurance.

As for the limitation of the data, the most outstanding problem is that some variables are extremely unevenly distributed. For example, there are only near 1000 customers without driving license. To make the data analysis more precise, I need to learn how to deal with unbalanced variables in the future to improve the model. Besides, there are several hard code in coding process. I hope to fix them using certain function if I can find such a proper function or calculation. Lastly, I think that the model should be simplified for next step because of its large AIC value.

Bibliography

(1)H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.

(2)Yihui Xie (2020). knitr: A General-Purpose Package for Dynamic Report Generation in R. R package version 1.29.

(3)Stefan Milton Bache and Hadley Wickham (2014). magrittr: A Forward-Pipe Operator for R. R package version 1.5. https://CRAN.R-project.org/package=magrittr.

(4)Hao Zhu (2020). kableExtra: Construct Complex Table with 'kable' and Pipe Syntax. R package version 1.2.1. https://CRAN.R-project.org/package=kableExtra.

(5)Hadley Wickham, Romain Fran?ois, Lionel Henry and Kirill Müller (2020). dplyr: A Grammar of Data Manipulation. R package version 1.0.2 https://CRAN.R-project.org/package=dplyr

(6)Andrew Gelman and Yu-Sung Su (2020). arm: Data Analysis Using Regression and Multilevel/Hierarchical Models. R package version 1.11-2. https://CRAN.R-project.org/package=arm

(7)Dean Attali and Christopher Baker (2019). ggExtra: Add Marginal Histograms to 'ggplot2', and More 'ggplot2' Enhancements. R package version 0.9. https://CRAN.R-project.org/package=ggExtra

(8)Kaggle. (2020). Health Insurance Cross Sell Prediction. Available from https://www.kaggle.com/anmolkumar/health-insurance-cross-sell-prediction?select=train.csv

Appendix

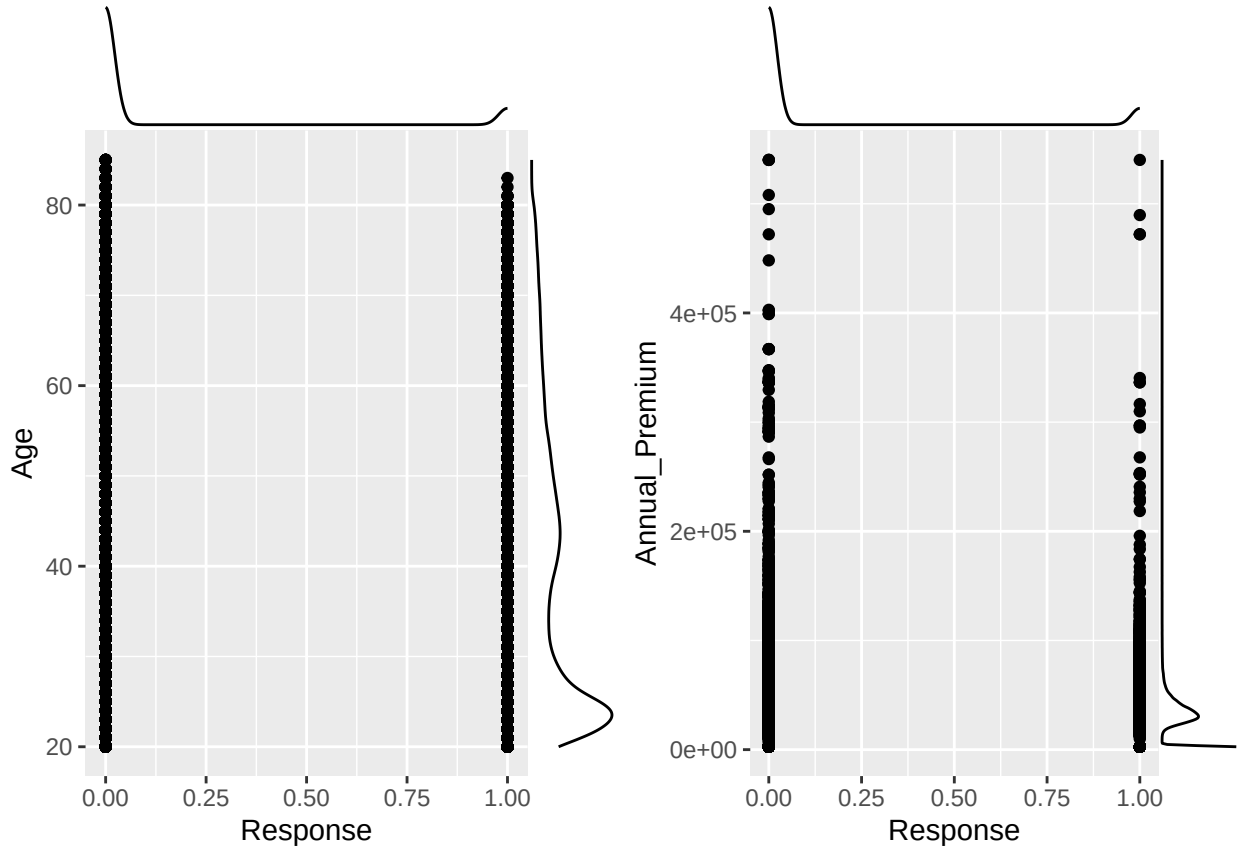
The code of EDA part

```
test <- read.csv("test.csv")
train <- read.csv("train.csv")
submission <- read.csv("sample_submission.csv")
test <- left_join(test, submission, join="id")

## For categorical predictors with more than 10 categories

train_region <- train %>% group_by(Region_Code,Response) %>% summarise(Count=n())
##summary(train_region$Count)
train_region_1 <- train_region %>% group_by(Region_Code) %>% summarise(Sum=sum(Count)) %>% arrange(desc
train_region %<>% filter(Region_Code==28|Region_Code==8|Region_Code==46|Region_Code==41|Region_Code==15
region <- train_region$Region_Code
response_6 <- rep(c("Not Interested","Interested"),5)
value_6 <- train_region$Count
data_6 <- data.frame(region,response_6,value_6)
p6 <- ggplot(data_6,aes(fill=response_6,y=value_6,x=as.character(region)))+
    geom_bar(position="stack",stat="identity")+theme(
    axis.text = element_text(size = 7),
        axis.title = element_text(size = 9, face = "bold"),
    legend.title = element_text(size=9),
    legend.text = element_text(size = 9))+scale_fill_brewer(palette = "Pastel2")+labs(fill="Response")+

train_channel <- train %>% group_by(Policy_Sales_Channel,Response) %>% summarise(Count=n())
##summary(train_channel$Count)
train_channel_1 <- train_channel %>% group_by(Policy_Sales_Channel) %>% summarise(Sum=sum(Count)) %>% a
train_channel %<>% filter(Policy_Sales_Channel==152|Policy_Sales_Channel==26|Policy_Sales_Channel==124|
channel <- train_channel$Policy_Sales_Channel
response_7 <- rep(c("Not Interested","Interested"),5)
value_7 <- train_channel$Count
data_7 <- data.frame(channel,response_7,value_7)
p7 <- ggplot(data_7,aes(fill=response_7,y=value_7,x=as.character(channel)))+
    geom_bar(position="stack",stat="identity")+theme(
    axis.text = element_text(size = 7),
        axis.title = element_text(size = 9, face = "bold"),
    legend.title = element_text(size=9),
    legend.text = element_text(size = 9))+scale_fill_brewer(palette = "Pastel2")+labs(fill="Response")+

train_vintage <- train %>% group_by(Vintage,Response) %>% summarise(Count=n())
##summary(train_vintage$Count)
train_vintage_1 <- train_vintage %>% group_by(Vintage) %>% summarise(Sum=sum(Count)) %>% arrange(desc(S
train_vintage %<>% filter(Vintage==256|Vintage==73|Vintage==282|Vintage==158|Vintage==187)
vintage <- train_vintage$Vintage
response_8 <- rep(c("Not Interested","Interested"),5)
value_8 <- train_vintage$Count
data_8 <- data.frame(vintage,response_8,value_8)
p8 <- ggplot(data_8,aes(fill=response_8,y=value_8,x=as.character(vintage)))+
    geom_bar(position="stack",stat="identity")+theme(
    axis.text = element_text(size = 7),
        axis.title = element_text(size = 9, face = "bold"),
    legend.title = element_text(size=9),
```

```
    legend.text = element_text(size = 9))+scale_fill_brewer(palette = "Pastel2")+labs(fill="Response")+
```

```
## Data visualization
## For categorical predictors
train_gender <- train %>% group_by(Gender,Response) %>% summarise(Count=n())
gender <- c(rep("Female",2),rep("Male",2))
response_1 <- rep(c("Not Interested","Interested"),2)
value_1 <- train_gender$Count
data_1 <- data.frame(gender,response_1,value_1)
p1 <- ggplot(data_1, aes(fill=response_1, y=value_1, x=gender)) +
    geom_bar(position="stack", stat="identity")+theme(
    axis.text = element_text(size = 7),
        axis.title = element_text(size = 9, face = "bold"),
    legend.title = element_text(size=9),
    legend.text = element_text(size = 9))+scale_fill_brewer(palette = "Pastel2") + labs(fill="Response")

train_license <- train %>% group_by(Driving_License,Response) %>% summarise(Count=n())
license <- c(rep("No License",2),rep("License",2))
response_2 <- rep(c("Not Interested","Interested"),2)
value_2 <- train_license$Count
data_2 <- data.frame(license,response_2,value_2)
p2 <- ggplot(data_2, aes(fill=response_2, y=value_2, x=license)) +
    geom_bar(position="stack", stat="identity")+theme(
    axis.text = element_text(size = 7),
        axis.title = element_text(size = 9, face = "bold"),
    legend.title = element_text(size=9),
    legend.text = element_text(size = 9))+scale_fill_brewer(palette = "Pastel2") + labs(fill="Response")

train_insured <- train %>% group_by(Previously_Insured,Response) %>% summarise(Count=n())
insured <- c(rep("No Insured",2),rep("Insured",2))
response_3 <- rep(c("Not Interested","Interested"),2)
value_3 <- train_insured$Count
data_3 <- data.frame(insured,response_3,value_3)
p3 <- ggplot(data_3, aes(fill=response_3, y=value_3, x=insured)) +
    geom_bar(position="stack", stat="identity")+theme(
    axis.text = element_text(size = 7),
        axis.title = element_text(size = 9, face = "bold"),
    legend.title = element_text(size=9),
    legend.text = element_text(size = 9))+scale_fill_brewer(palette = "Pastel2") + labs(fill="Response")

train_ve_age <- train %>% group_by(Vehicle_Age,Response) %>% summarise(Count=n())
vehicle_age <- c(rep("<1 Year",2),rep(">2 Years",2),rep("1-2 Year",2))
response_4 <- rep(c("Not Interested","Interested"),3)
value_4 <- train_ve_age$Count
data_4 <- data.frame(vehicle_age,response_4,value_4)
p4 <- ggplot(data_4, aes(fill=response_4, y=value_4, x=vehicle_age)) +
    geom_bar(position="stack", stat="identity")+theme(
    axis.text = element_text(size = 7),
        axis.title = element_text(size = 9, face = "bold"),
    legend.title = element_text(size=9),
    legend.text = element_text(size = 9))+scale_fill_brewer(palette = "Pastel2") + labs(fill="Response")

train_ve_damage <- train %>% group_by(Vehicle_Damage,Response) %>% summarise(Count=n())
vehicle_damage <- c(rep("No Damage",2),rep("Damage",2))
```

```
response_5 <- rep(c("Not Interested","Interested"),2)
value_5 <- train_ve_damage$Count
data_5 <- data.frame(vehicle_damage,response_5,value_5)
p5 <- ggplot(data_5, aes(fill=response_5, y=value_5, x=vehicle_damage)) +
    geom_bar(position="stack", stat="identity")+theme(
    axis.text = element_text(size = 7),
        axis.title = element_text(size = 9, face = "bold"),
    legend.title = element_text(size=9),
    legend.text = element_text(size = 9))+scale_fill_brewer(palette = "Pastel2") + labs(fill="Response")

## For continuout predictors
g1 <- ggplot(data=train)+geom_point(aes(x=Response,y=Age))
g1_1 <- ggMarginal(g1, type="density")

g2 <- ggplot(data=train)+geom_point(aes(x=Response,y=Annual_Premium))
g2_1 <- ggMarginal(g2, type="density")

grid.arrange(arrangeGrob(g1_1,g2_1,ncol=2))
```



The results of modeling

```
summary(fit)
```

```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 0) [glmerMod]
##  Family: binomial  ( logit )
```

```
## Formula: Response ~ Gender + c_Age + Driving_License + Previously_Insured +
##     Vehicle_Age + Vehicle_Damage + log_Annual_Premium + (1 |
##     Region_Code) + (1 | Policy_Sales_Channel)
##    Data: train
## Control: glmerControl("bobyqa")
##
##       AIC       BIC    logLik  deviance  df.resid
##  205001.1  205109.6 -102490.5  204981.1    381071
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -1.204 -0.458 -0.033 -0.020 87.851
##
## Random effects:
##  Groups               Name        Variance Std.Dev.
##  Policy_Sales_Channel (Intercept) 0.17028  0.4126
##  Region_Code          (Intercept) 0.06357  0.2521
## Number of obs: 381081, groups:  Policy_Sales_Channel, 143; Region_Code, 53
##
## Fixed effects:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -5.123281   0.189627 -27.018  < 2e-16 ***
## Gender              0.067770   0.011309   5.993 2.07e-09 ***
## c_Age              -0.385233   0.008377 -45.985  < 2e-16 ***
## Driving_License     1.180040   0.163751   7.206 5.75e-13 ***
## Previously_Insured -3.945729   0.082775 -47.668  < 2e-16 ***
## Vehicle_Age         0.267306   0.014952  17.878  < 2e-16 ***
## Vehicle_Damage      1.955334   0.034621  56.479  < 2e-16 ***
## log_Annual_Premium  0.025359   0.006260   4.051 5.10e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Gender c_Age  Drvn_L Prvs_I Vhcl_A Vhcl_D
## Gender      -0.039
## c_Age        0.007 -0.040
## Drivng_Lcns -0.862  0.007  0.046
## Prvsly_Insr -0.047 -0.002 -0.003  0.000
## Vehicle_Age -0.127 -0.023 -0.411 -0.015  0.007
## Vehicle_Dmg -0.169 -0.004 -0.008  0.001  0.256 -0.026
## lg_Annl_Prm -0.319 -0.002 -0.005  0.003 -0.002 -0.002 -0.013
```

```
ranef(fit)
```

```
## $Policy_Sales_Channel
##       (Intercept)
## 1   -0.2544735351
## 2    0.0246979677
## 3    0.7195172022
## 4    0.3648128028
## 6   -0.0344621941
## 7    0.2718339813
## 8    0.0514927351
## 9    0.0921266722
## 10   0.0872927054
```

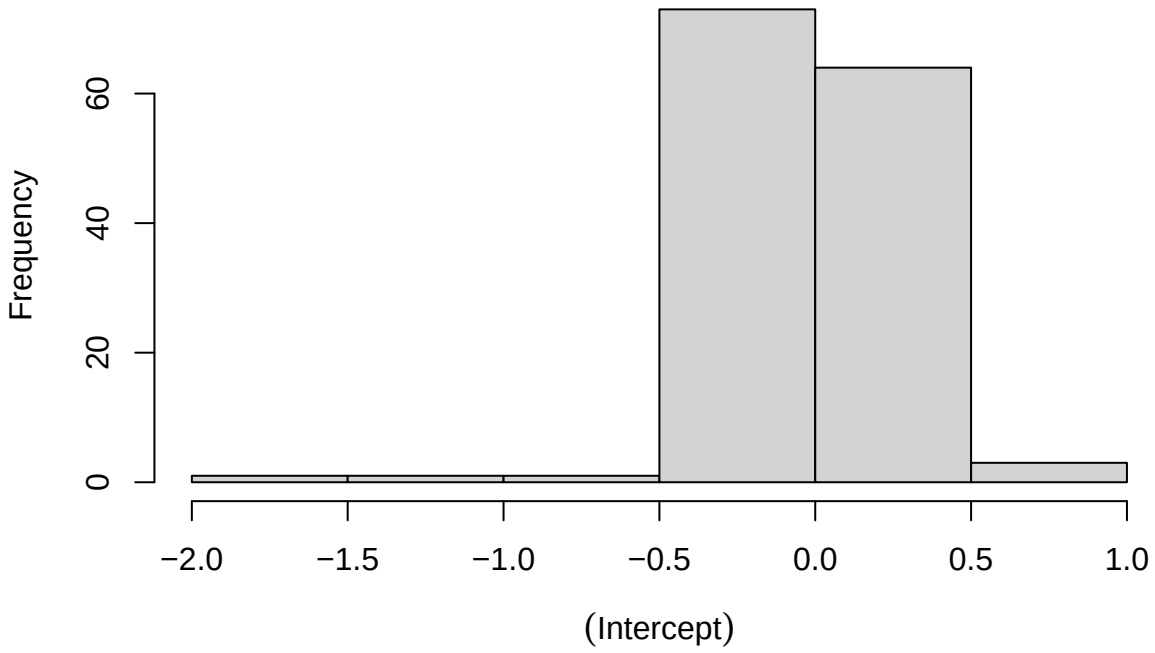
```
## 11  -0.2336533945
## 12   0.0724592431
## 13   0.0137563544
## 14   0.1221037781
## 15   0.2397063660
## 16   0.0149195586
## 17   0.1837598686
## 18  -0.3098413269
## 19  -0.0665381528
## 20   0.0744849838
## 21  -0.0698570035
## 22  -0.2858420210
## 23   0.0985040899
## 24   0.0658306638
## 25   0.4947870931
## 26   0.4444631661
## 29   0.0733898776
## 30   0.0822154854
## 31   0.4781900598
## 32  -0.0280180365
## 33  -0.0882827910
## 34  -0.0734653348
## 35   0.0036810551
## 36   0.4993883473
## 37  -0.2005474600
## 38  -0.0933641314
## 39  -0.1463135421
## 40  -0.0550631360
## 42   0.2626836703
## 43   0.1227660398
## 44   0.2763901375
## 45  -0.1640804763
## 46  -0.2232502582
## 47  -0.0767025058
## 48  -0.1631458138
## 49   0.0603166338
## 51  -0.0797956249
## 52  -0.1353046943
## 53   0.1525639867
## 54  -0.1116461816
## 55   0.0334408042
## 56   0.1326890565
## 57  -0.0135493330
## 58  -0.0784944868
## 59   0.1492275421
## 60  -0.1463085622
## 61  -0.2839378747
## 62   0.0834001865
## 63   0.0048112953
## 64  -0.2329261653
## 65  -0.1703911999
## 66  -0.0102411639
## 69   0.0649898801
## 70  -0.0991513939
```

```
## 71  -0.0875474920
## 73  -0.1450967940
## 74  -0.0276941737
## 76  -0.0556607784
## 78   0.1106809826
## 79  -0.1831334064
## 80   0.2390664591
## 81   0.3560799103
## 82  -0.0355516790
## 83  -0.0331656163
## 86   0.1657450679
## 87   0.1125907493
## 88  -0.3890423955
## 89   0.0353873690
## 90   0.3592800924
## 91   0.3394071962
## 92   0.1069389175
## 93  -0.0398063248
## 94   0.2623895251
## 95  -0.0988292910
## 96  -0.2002246581
## 97   0.0255485603
## 98  -0.1385467882
## 99  -0.0695503595
## 100  0.1348797038
## 101  0.1145266180
## 102 -0.0265197814
## 103  0.1347273475
## 105 -0.0003247827
## 106  0.3863062371
## 107  0.0159756084
## 108 -0.0024180347
## 109 -0.1966990161
## 110 -0.1003774555
## 111 -0.1044952327
## 112 -0.0486699733
## 113 -0.0018558044
## 114 -0.0809630133
## 115 -0.0329105850
## 116 -0.0618078483
## 117 -0.0008904407
## 118 -0.1984961212
## 119 -0.0089015940
## 120  0.0385514571
## 121  0.5981802558
## 122  0.1621625476
## 123  0.1251399773
## 124  0.2596030264
## 125 -0.0896178245
## 126 -0.0393775403
## 127 -0.2579925963
## 128 -0.0708309669
## 129 -0.2592151424
## 130  0.0924980732
```

```
## 131 -0.2366864215
## 132 -0.2178391110
## 133 -0.2227915372
## 134 -0.0872690772
## 135 -0.1881217782
## 136  0.3437804241
## 137 -0.0989945543
## 138 -0.1007329381
## 139 -0.3026568948
## 140 -0.2869227652
## 145  0.2427725800
## 146 -0.1633228622
## 147  0.1827090256
## 148  0.0392496848
## 150  0.3187242167
## 151 -1.0193599635
## 152 -0.8780012591
## 153 -0.4000445787
## 154  0.2829224535
## 155  0.6624556016
## 156  0.0928540639
## 157  0.3534033570
## 158  0.3262680200
## 159 -0.3877594487
## 160 -1.8104366268
## 163  0.4783027224
##
## $Region_Code
##    (Intercept)
## 0  -0.55454050
## 1  -0.32830150
## 2  -0.10311322
## 3   0.29630936
## 4   0.21962385
## 5   0.06745275
## 6   0.25846728
## 7   0.06494315
## 8  -0.05280745
## 9  -0.23141014
## 10  0.02316073
## 11  0.46711938
## 12  0.09918208
## 13  0.01610686
## 14  0.15411801
## 15 -0.08090063
## 16 -0.05170363
## 17 -0.15878499
## 18  0.39260200
## 19  0.09353991
## 20 -0.26648617
## 21  0.22244154
## 22 -0.25554303
## 23  0.24701932
## 24  0.13235825
```

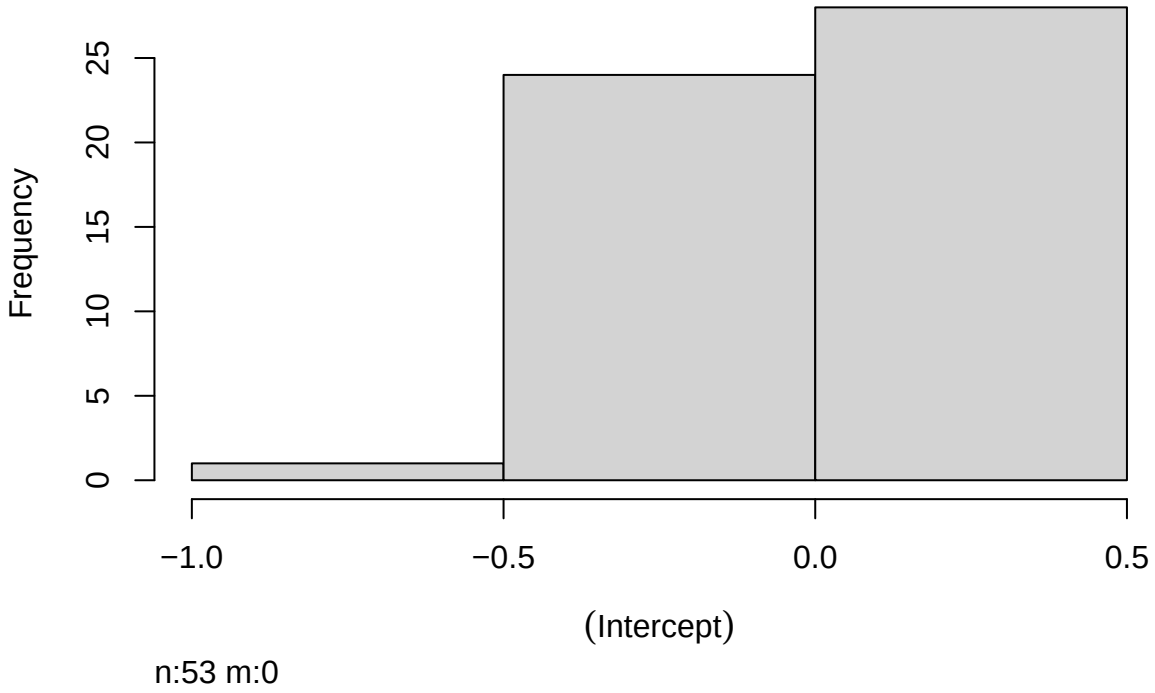
```
## 25 -0.37147205
## 26 -0.32235469
## 27 -0.06472367
## 28  0.18879809
## 29  0.41895614
## 30  0.27552187
## 31 -0.20444535
## 32  0.14679843
## 33  0.07098706
## 34 -0.23793882
## 35  0.39585161
## 36  0.01248704
## 37 -0.04425154
## 38  0.27541773
## 39 -0.03598342
## 40 -0.04115976
## 41  0.37328691
## 42 -0.24814025
## 43 -0.19608912
## 44 -0.36085553
## 45  0.13293961
## 46  0.10117041
## 47 -0.15999111
## 48 -0.38318675
## 49 -0.21811523
## 50 -0.29738274
## 51  0.08908954
## 52  0.03393239
##
## with conditional variances for "Policy_Sales_Channel" "Region_Code"
```

```r
ranef <- ranef(fit)
hist(ranef$Policy_Sales_Channel)
```



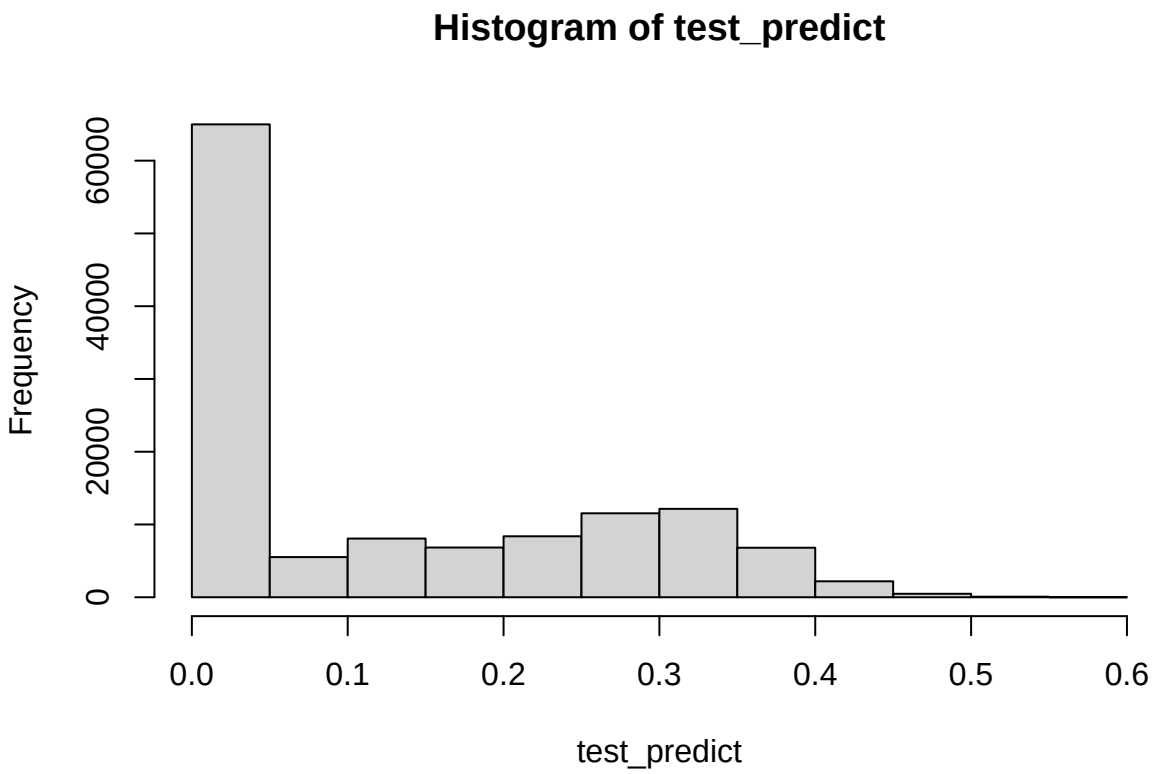
n:143 m:0

```
hist(ranef$Region_Code)
```



The prediction of test data

```
hist(test_predict)
```


Histogram of test_predict
Frequency
0
20000
40000
60000
0.0
0.1
0.2
0.3
0.4
0.5
0.6
test_predict