ISOM5610 Project

Team 1

14 December 2018

```
setwd("~/MSBA/ISOM5610/final")
claim <- read.table("Claim.csv", sep = ",", header = TRUE)
str(claim)
```

```
## 'data.frame':    412412 obs. of  10 variables:
##  $ PolicyID : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Claim    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Exposure : num  0.09 0.84 0.52 0.45 0.15 0.75 0.81 0.05 0.76 0.34 ...
##  $ Power    : Factor w/ 12 levels "d","e","f","g",..: 4 4 3 3 4 4 1 1 1 6 ...
##  $ CarAge   : int  0 0 2 2 0 0 1 0 9 0 ...
##  $ DriverAge: int  46 46 38 38 41 41 27 27 23 44 ...
##  $ Brand    : Factor w/ 7 levels "Fiat","Japanese (except Nissan) or Korean",..: 2 2 2 2 2 2 2 2 1 2
##  $ Gas      : Factor w/ 2 levels "Diesel","Regular": 1 1 2 2 1 1 2 2 2 2 ...
##  $ Region   : Factor w/ 10 levels "R11","R23","R24",..: 9 9 5 5 6 6 9 9 5 1 ...
##  $ Density  : int  76 76 3003 3003 60 60 695 695 7887 27000 ...
```

```
claim <- claim[-1]
summary(claim)
```

```
##      Claim            Exposure            Power           CarAge
##  Min.   :0.00000   Min.   :0.002732   f      :95538   Min.   :  0.000
##  1st Qu.:0.00000   1st Qu.:0.200000   g      :91050   1st Qu.:  3.000
##  Median :0.00000   Median :0.530000   e      :76863   Median :  7.000
##  Mean   :0.03548   Mean   :0.560810   d      :67889   Mean   :  7.533
##  3rd Qu.:0.00000   3rd Qu.:1.000000   h      :26650   3rd Qu.: 12.000
##  Max.   :1.00000   Max.   :1.990000   j      :18002   Max.   :100.000
##                                       (Other):36420
##    DriverAge                                    Brand
##  Min.   :18.00   Fiat                                : 16691
##  1st Qu.:34.00   Japanese (except Nissan) or Korean: 78898
##  Median :44.00   Mercedes, Chrysler or BMW           : 19248
##  Mean   :45.32   Opel, General Motors or Ford        : 37330
##  3rd Qu.:54.00   Renault, Nissan or Citroen          :217822
##  Max.   :99.00   Volkswagen, Audi, Skoda or Seat     : 32575
##                  other                               :  9848
##       Gas             Region          Density
##  Diesel :205559   R24    :160392   Min.   :    2
##  Regular:206853   R11    : 69603   1st Qu.:   67
##                   R53    : 42047   Median :  287
##                   R52    : 38675   Mean   : 1983
##                   R72    : 31263   3rd Qu.: 1408
##                   R31    : 27219   Max.   :27000
##                   (Other): 43213
```

```
sum(is.na(claim)) # check missing value
```

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

```r
summary(claim$Power)
```

```
##     d     e     f     g     h     i     j     k     l     m     n     o
## 67889 76863 95538 91050 26650 17589 18002  9521  4673  1829  1303  1505
```

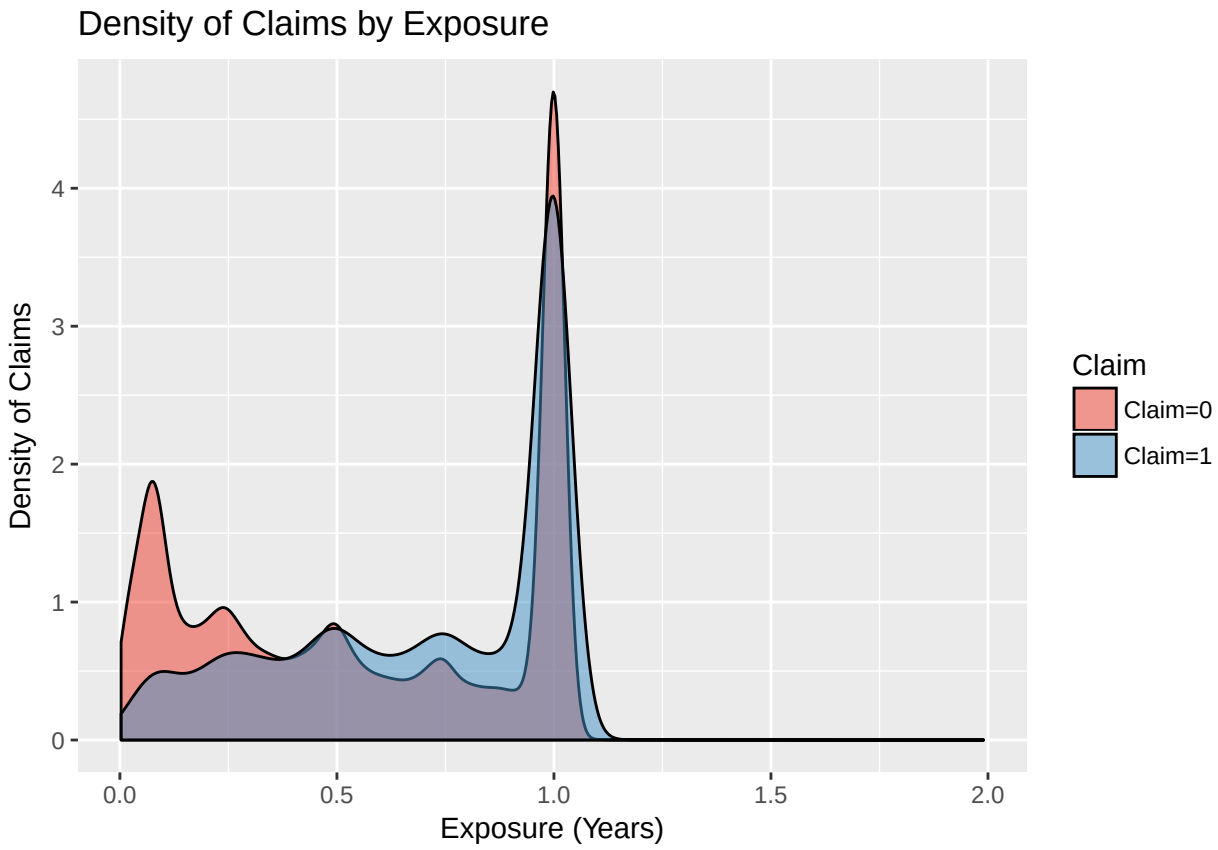
```r
summary(claim$Region)
```

```
##    R11    R23    R24    R25    R31    R52    R53    R54    R72    R74
##  69603   8773 160392  10870  27219  38675  42047  19015  31263   4555
```

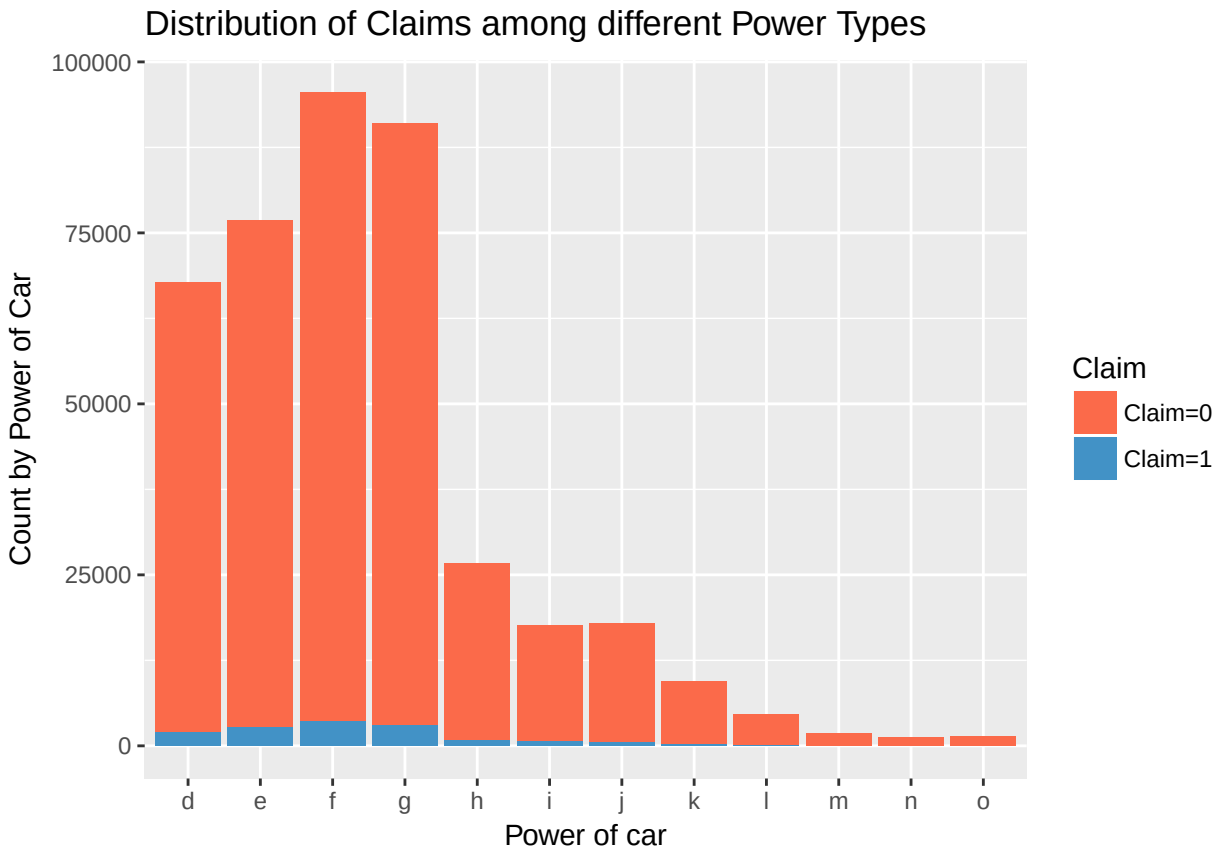
```r
avg_power <- data.frame(sapply(split(claim$Claim,claim$Power),mean))
# colnames(avg_power) <- 'avg'
# avg_power$Power <- rownames(avg_power)
avg_brand <- sapply(split(claim$Claim,claim$Brand),mean)
avg_region <- sapply(split(claim$Claim,claim$Region),mean)
## this chunk calculate average values in different categories
```

There is no missing value. Claim: binary. Power: 12 categories. Brand:7 categories. Gas: binary. Region: 10 regions.

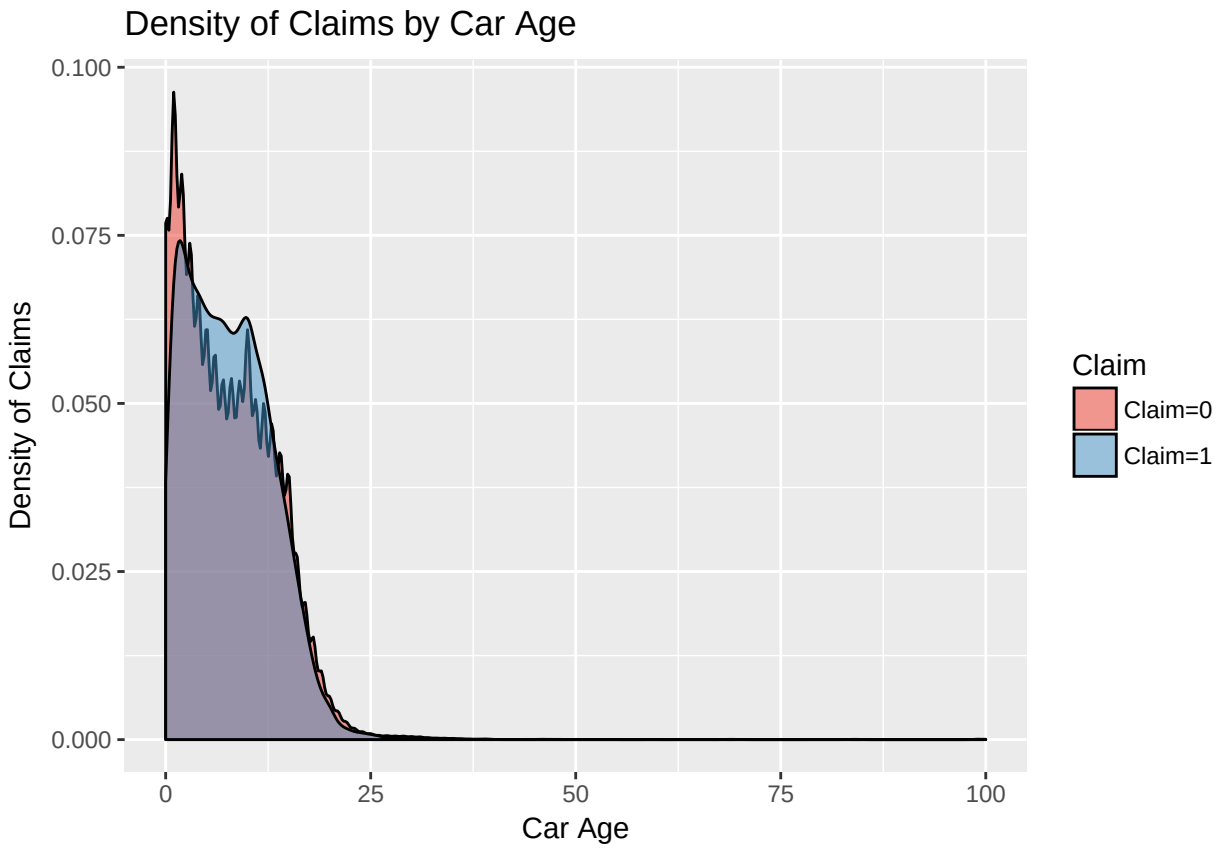
```r
library(ggplot2)
library(RColorBrewer)


ds1 <- ggplot(claim, aes(x=Exposure)) +
    geom_density(aes(fill=factor(Claim)),alpha=0.5) +
    labs(title="Density of Claims by Exposure",
       y="Density of Claims",
       x="Exposure (Years)") +
    scale_fill_manual(name = "Claim",
                      values = c(brewer.pal(7, "Reds")[5], brewer.pal(7, "Blues")[5]),
                      labels = c("Claim=0", "Claim=1"))
plot(ds1)
```



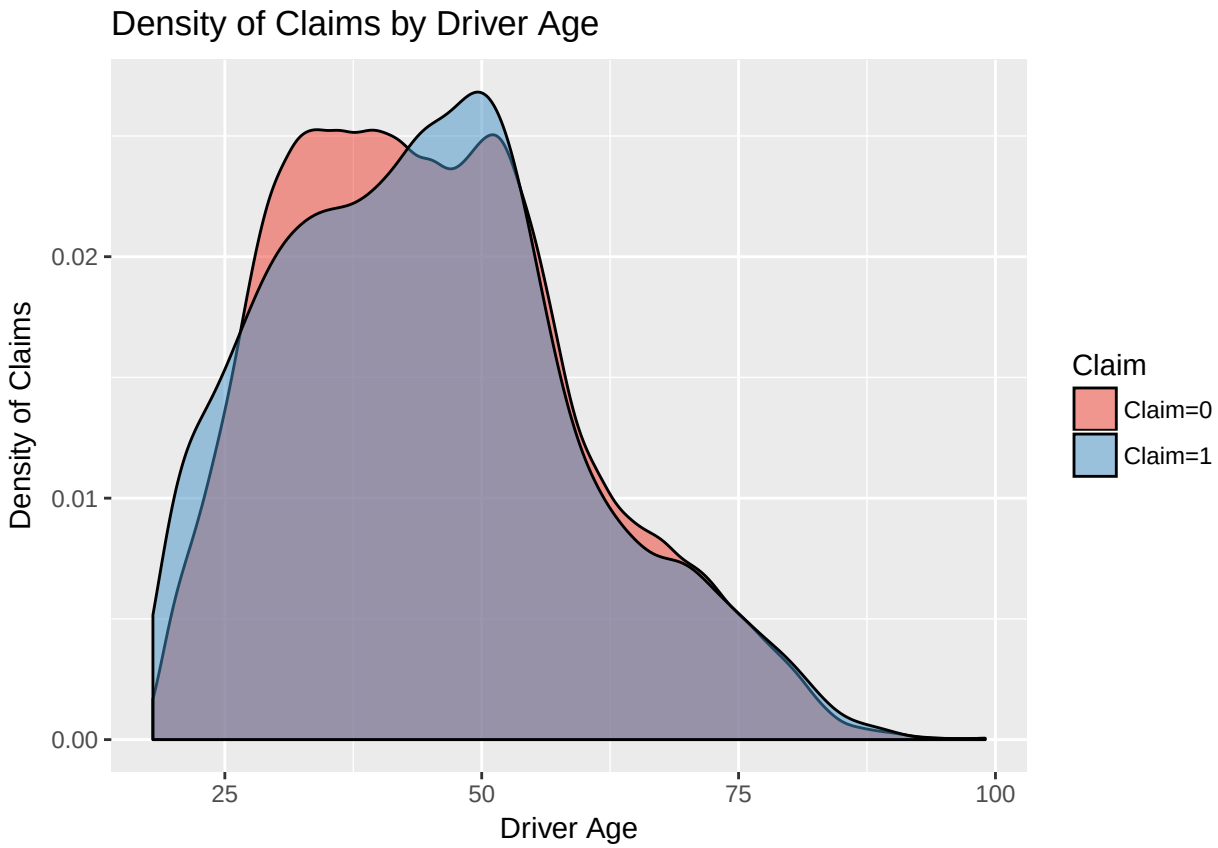

```
bar1 <- ggplot() + geom_bar(data=claim, aes(x=Power,fill=factor(Claim)))+
   labs(title="Distribution of Claims among different Power Types", y="Count by Power of Car", x="Power
scale_fill_manual(name = "Claim",values = c(brewer.pal(7, "Reds")[4],brewer.pal(7, "Blues")[5]),
                    labels = c("Claim=0", "Claim=1"))

plot(bar1)
```



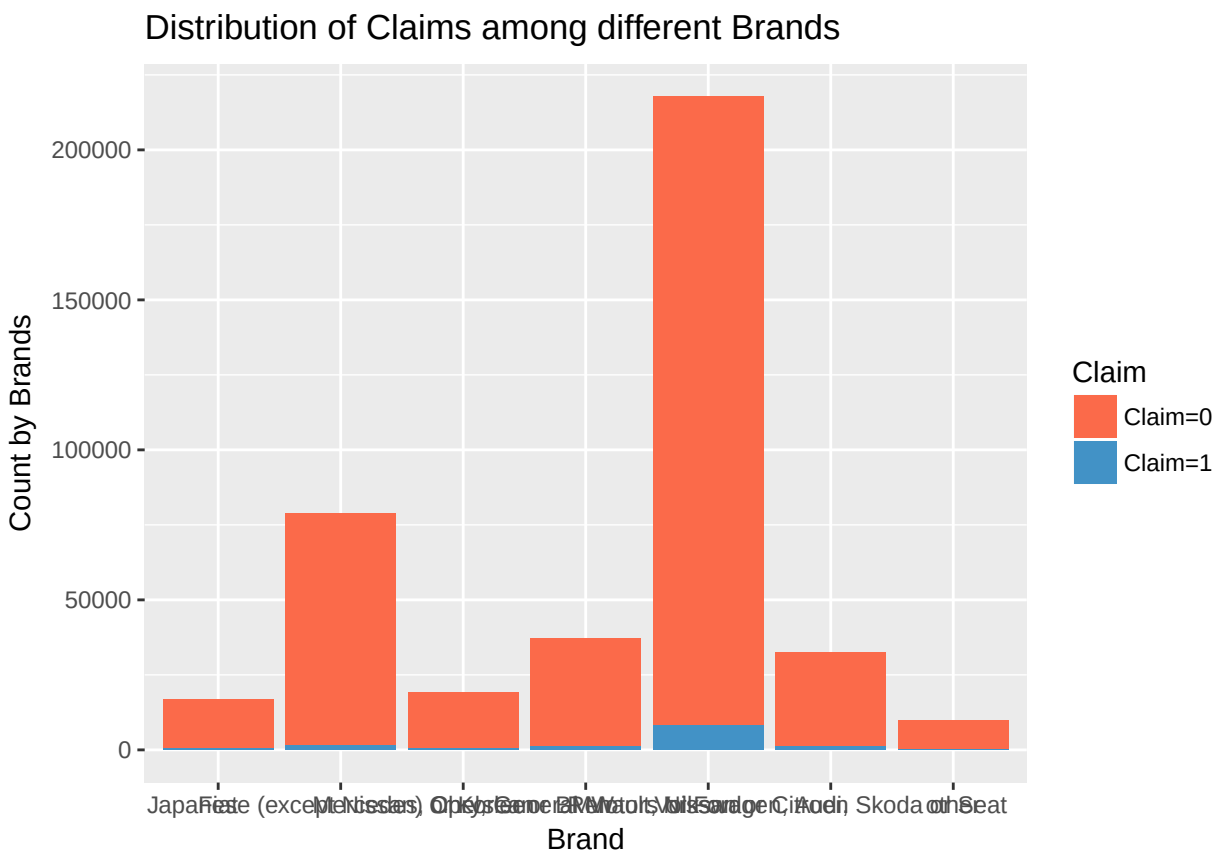

```
ds2 <- ggplot(claim, aes(x=CarAge)) +
  geom_density(aes(fill=factor(Claim)),alpha=0.5)+
  labs(title="Density of Claims by Car Age", y="Density of Claims", x="Car Age")+
  scale_fill_manual(name = "Claim",values = c(brewer.pal(7, "Reds")[5],brewer.pal(7, "Blues")[5]),
                    labels = c("Claim=0", "Claim=1"))
plot(ds2)
```



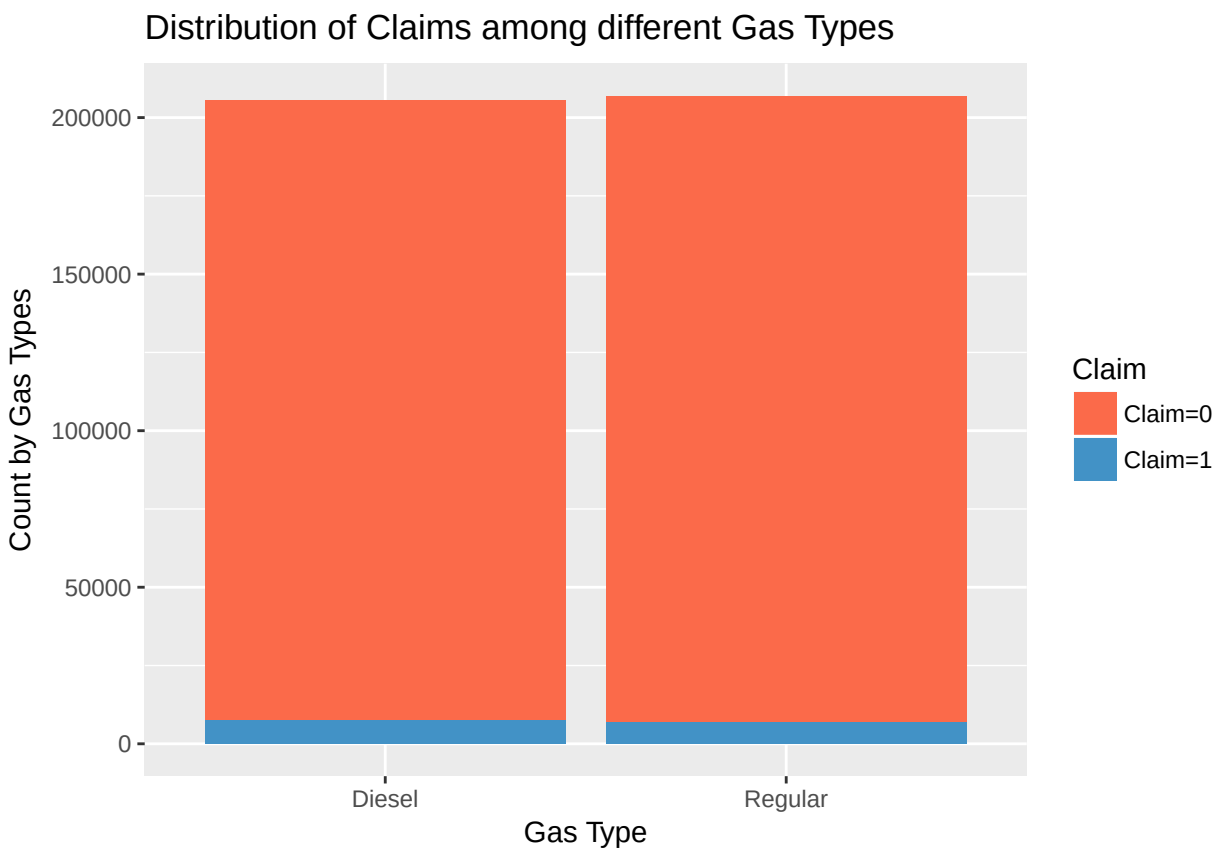

```
ds3 <- ggplot(claim, aes(x=DriverAge)) +
  geom_density(aes(fill=factor(Claim)),alpha=0.5)+
  labs(title="Density of Claims by Driver Age", y="Density of Claims", x="Driver Age")+
  scale_fill_manual(name = "Claim",values = c(brewer.pal(7, "Reds")[5],brewer.pal(7, "Blues")[5]),
                    labels = c("Claim=0", "Claim=1"))
plot(ds3)
```



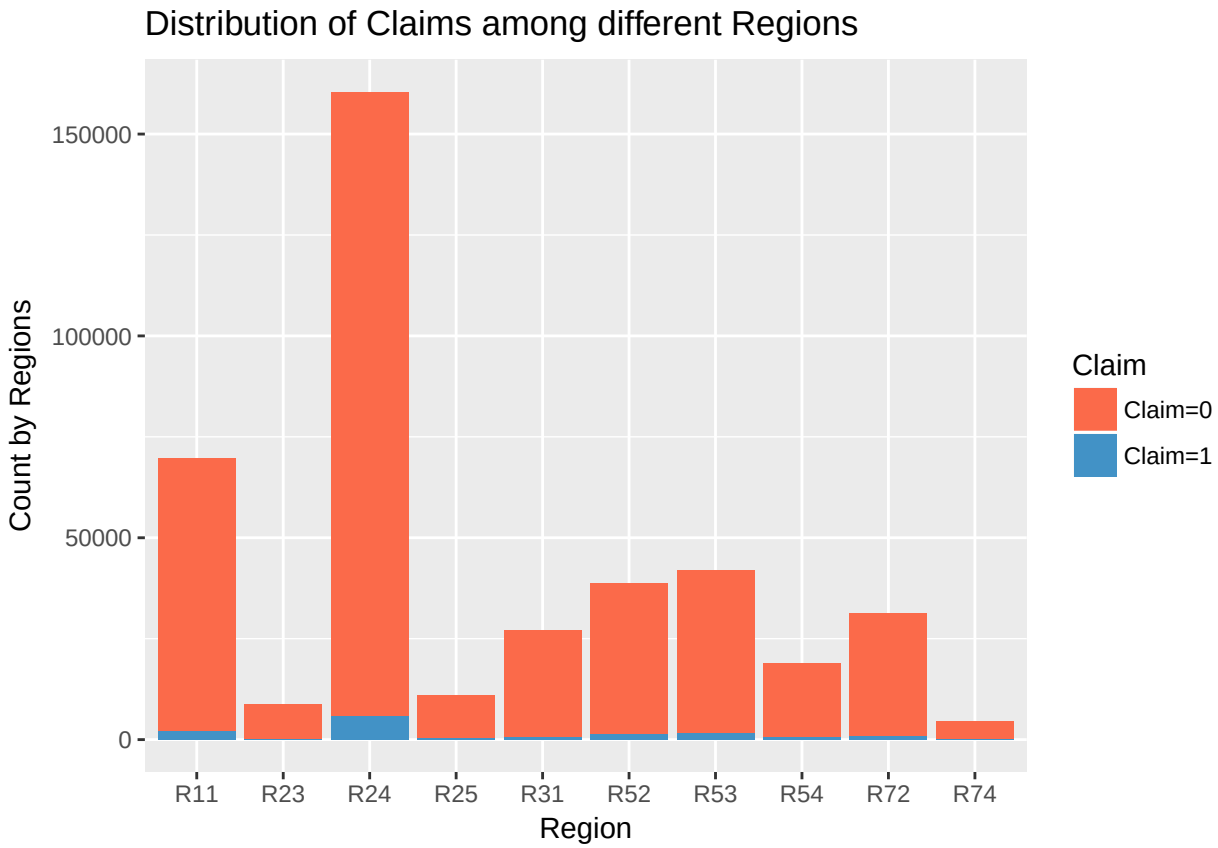

```
bar2 <- ggplot(claim, aes(x = Brand)) + geom_bar(aes(fill=factor(Claim)))+
   labs(title="Distribution of Claims among different Brands", y="Count by Brands", x="Brand")+
scale_fill_manual(name = "Claim",values = c(brewer.pal(7, "Reds")[4],brewer.pal(7, "Blues")[5]),
                    labels = c("Claim=0", "Claim=1"))

plot(bar2)
```



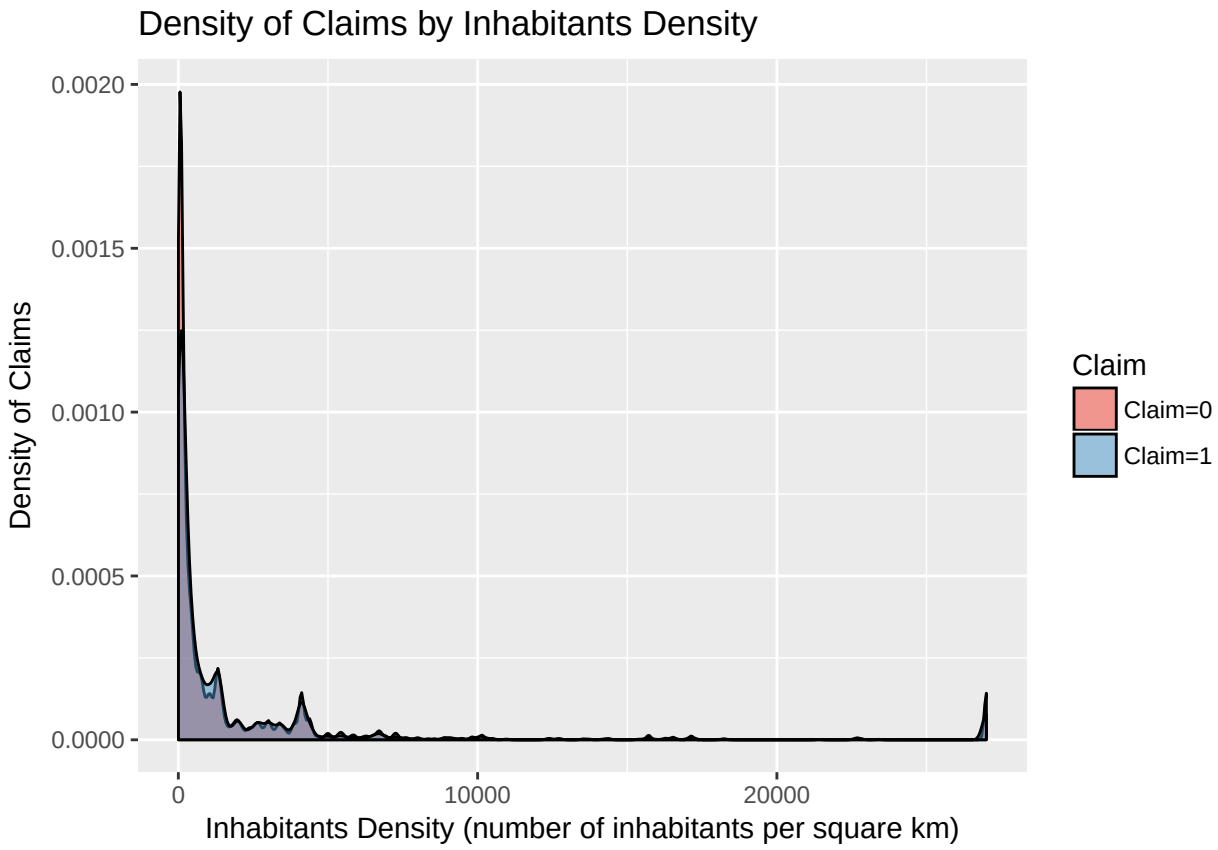

```
bar3 <- ggplot(claim, aes(x = Gas)) + geom_bar(aes(fill=factor(Claim)))+
   labs(title="Distribution of Claims among different Gas Types", y="Count by Gas Types", x="Gas Type")+
scale_fill_manual(name = "Claim",values = c(brewer.pal(7, "Reds")[4],brewer.pal(7, "Blues")[5]),
                  labels = c("Claim=0", "Claim=1"))

plot(bar3)
```




```
bar4 <- ggplot(claim, aes(x = Region)) + geom_bar(aes(fill=factor(Claim)))+
   labs(title="Distribution of Claims among different Regions", y="Count by Regions", x="Region")+
scale_fill_manual(name = "Claim",values = c(brewer.pal(7, "Reds")[4],brewer.pal(7, "Blues")[5]),
                    labels = c("Claim=0", "Claim=1"))

plot(bar4)
```



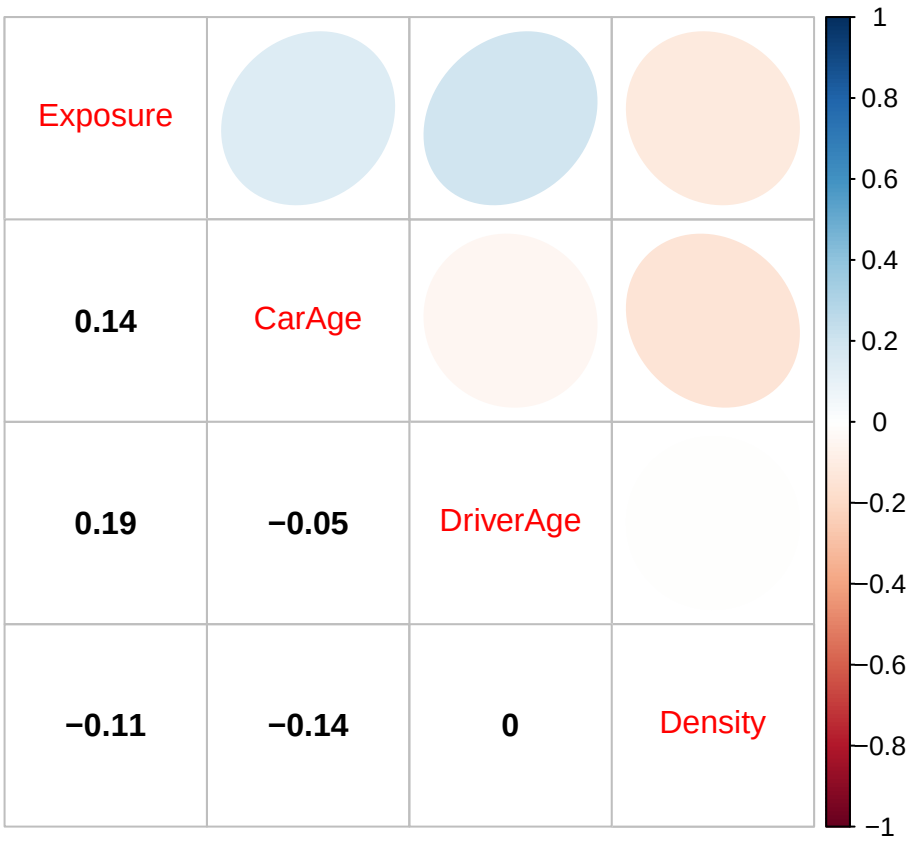

```
ds4 <- ggplot(claim, aes(x=Density)) +
  geom_density(aes(fill=factor(Claim)),alpha=0.5)+
  labs(title="Density of Claims by Inhabitants Density", y="Density of Claims", x="Inhabitants Density
  scale_fill_manual(name = "Claim",values = c(brewer.pal(7, "Reds")[5],brewer.pal(7, "Blues")[5]),
                    labels = c("Claim=0", "Claim=1"))
plot(ds4)
```




```
# check inter-correlation between non-categorical variables
claim1<-claim[c(2,4,5,9)]
library(corrplot)
```

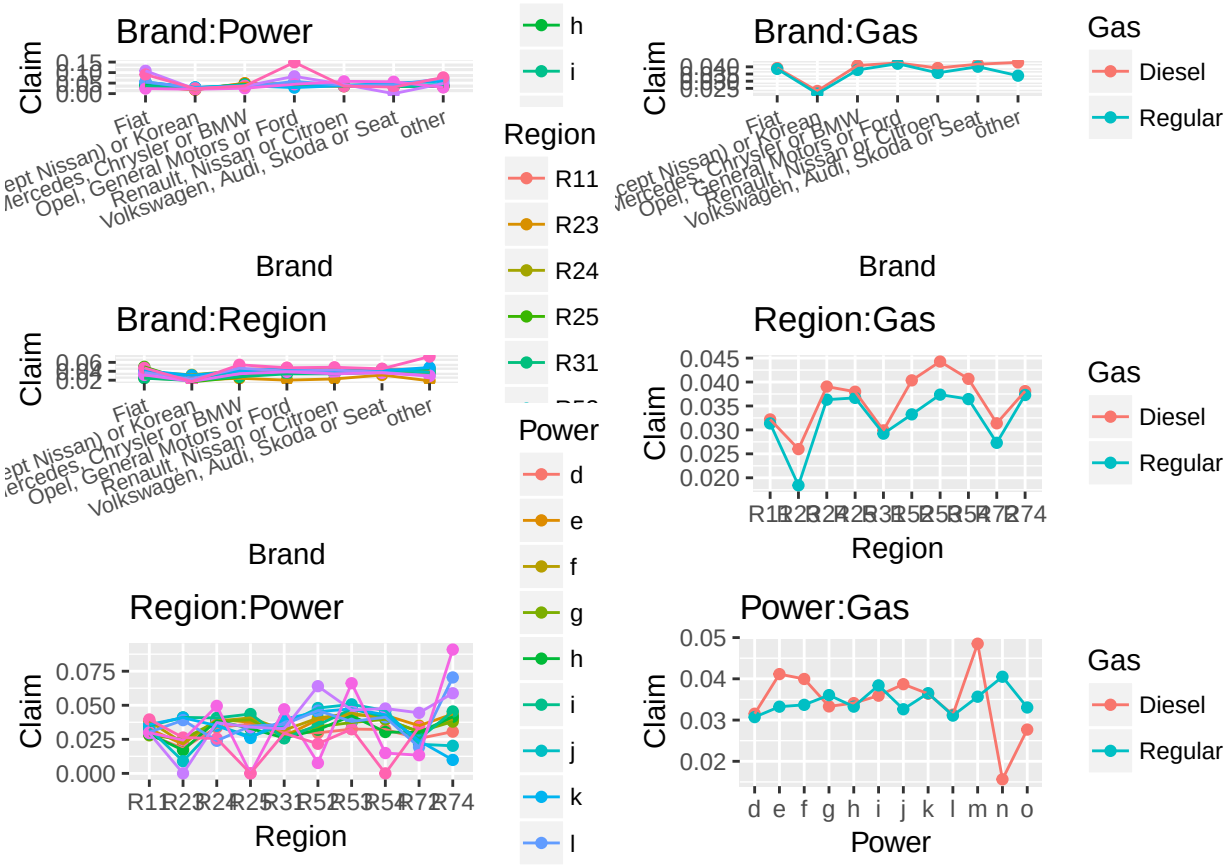
```
## corrplot 0.84 loaded
```

```
corrplot.mixed(cor(claim1), upper = "ellipse", lower.col = "black")
```




```
# check relationship between categorical variables
attach(claim)
r1 <- ggplot() +
  aes(x = Brand, color = Power, group = Power, y = Claim) +
  stat_summary(fun.y = mean, geom = "point") +
  stat_summary(fun.y = mean, geom = "line") + ggtitle('Brand:Power')+
  theme(axis.text.x = element_text(angle = 20, hjust = 1,size=8))


r2 <- ggplot() +
  aes(x = Brand, color = Gas, group = Gas, y = Claim) +
  stat_summary(fun.y = mean, geom = "point") +
  stat_summary(fun.y = mean, geom = "line") + ggtitle('Brand:Gas')+
  theme(axis.text.x = element_text(angle = 20, hjust = 1,size=8))


r3 <- ggplot() +
  aes(x = Brand, color = Region, group = Region, y = Claim) +
  stat_summary(fun.y = mean, geom = "point") +
  stat_summary(fun.y = mean, geom = "line") + ggtitle('Brand:Region')+
  theme(axis.text.x = element_text(angle = 20, hjust = 1,size=8))


r4 <- ggplot() +
  aes(x = Region, color = Gas, group = Gas, y = Claim) +
  stat_summary(fun.y = mean, geom = "point") +
  stat_summary(fun.y = mean, geom = "line") + ggtitle('Region:Gas')
```

```
r5 <- ggplot() +
  aes(x = Region, color = Power, group = Power, y = Claim) +
  stat_summary(fun.y = mean, geom = "point") +
  stat_summary(fun.y = mean, geom = "line") + ggtitle('Region:Power')


r6 <- ggplot() +
  aes(x = Power, color = Gas, group = Gas, y = Claim) +
  stat_summary(fun.y = mean, geom = "point") +
  stat_summary(fun.y = mean, geom = "line") + ggtitle('Power:Gas')

library(gridExtra)
grid.arrange(r1, r2, r3, r4, r5, r6, nrow = 3, ncol=2)
```




```
# geographical plot of claim %
# library(dplyr)
# claim_by_region <- tapply(claim$Claim, claim$Region, sum)
# count_by_region <- summary(claim$Region)
# regionID <- names(count_by_region)
# regionIdx <- sub('.', '', regionID)
#
# Sys.setlocale('LC_ALL','French')
# library(readxl)
# url1<-'https://insee.fr/fr/statistiques/fichier/1893198/estim-pop-dep-sexe-gca-1975-2018.xls '
```

```
# tempdb <- tempfile()
# download.file(url1, tempdb, mode="wb")
# raw_db <- as.data.frame(read_excel(path = tempdb, range="2018!A6:B101", col_names=FALSE))
# names(raw_db) <- c("RIdx", "RName")
#
# region_table <- data.frame(regionID=regionID,
#                            regionName=raw_db$RName[match(regionIdx, raw_db$RIdx)],
#                            regionCount=count_by_region,
#                            regionClaim=claim_by_region,
#                            regionClaimPct=claim_by_region/count_by_region*100
#                            )
#
# library(maps)
# france_map <- map_data("france")
# claim_map <- merge(france_map, region_table, by.x = "region", by.y = "regionName", all.x = TRUE)
# claim_map <- arrange(claim_map, group, order)
# ggplot(claim_map, aes(x = long, y = lat, group = group, fill = regionClaimPct)) +
#     geom_polygon(colour = "white")+
#     labs(title="Claim Rate (%) by Region", fill = "Claim Rate\n(%)") +
#     scale_fill_viridis_c() +
#     theme_void()
```

Try to fit

```
fit.full <- glm(Claim~.,family=binomial,data = claim)  ## this one with default link func
summary(fit.full)
```

```
##
## Call:
## glm(formula = Claim ~ ., family = binomial, data = claim)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.7432  -0.3130  -0.2491  -0.2050   3.0881
##
## Coefficients:
##                                       Estimate Std. Error z value
## (Intercept)                         -3.585e+00  6.208e-02 -57.740
## Exposure                             1.195e+00  2.623e-02  45.557
## Powere                               7.988e-02  3.024e-02   2.641
## Powerf                               1.052e-01  2.948e-02   3.570
## Powerg                               7.115e-02  2.928e-02   2.430
## Powerh                               1.024e-01  4.186e-02   2.446
## Poweri                               2.131e-01  4.603e-02   4.629
## Powerj                               1.956e-01  4.726e-02   4.138
## Powerk                               2.531e-01  6.010e-02   4.212
## Powerl                               1.328e-01  8.960e-02   1.483
## Powerm                               1.648e-01  1.273e-01   1.294
## Powern                               1.732e-01  1.506e-01   1.151
## Powero                               2.242e-01  1.498e-01   1.497
## CarAge                              -1.064e-02  1.686e-03  -6.311
## DriverAge                           -7.203e-03  6.191e-04 -11.635
```

```
## BrandJapanese (except Nissan) or Korean -4.645e-01  4.919e-02  -9.442
## BrandMercedes, Chrysler or BMW          -6.532e-03  5.701e-02  -0.115
## BrandOpel, General Motors or Ford        6.876e-02  4.812e-02   1.429
## BrandRenault, Nissan or Citroen         -6.456e-02  4.211e-02  -1.533
## BrandVolkswagen, Audi, Skoda or Seat     1.984e-02  4.938e-02   0.402
## Brandother                              -6.564e-02  6.687e-02  -0.982
## GasRegular                              -8.982e-02  1.850e-02  -4.856
## RegionR23                               -2.666e-01  7.747e-02  -3.441
## RegionR24                               -7.121e-02  3.374e-02  -2.110
## RegionR25                               -3.716e-02  5.891e-02  -0.631
## RegionR31                               -6.040e-02  4.556e-02  -1.326
## RegionR52                               -1.401e-02  4.008e-02  -0.350
## RegionR53                               -1.625e-02  3.933e-02  -0.413
## RegionR54                                2.729e-02  4.849e-02   0.563
## RegionR72                               -7.362e-02  4.402e-02  -1.672
## RegionR74                                1.404e-01  8.356e-02   1.680
## Density                                  1.487e-05  2.146e-06   6.932
##                                        Pr(>|z|)
## (Intercept)                             < 2e-16 ***
## Exposure                                < 2e-16 ***
## Powere                                 0.008260 **
## Powerf                                 0.000357 ***
## Powerg                                 0.015080 *
## Powerh                                 0.014446 *
## Poweri                                 3.68e-06 ***
## Powerj                                 3.51e-05 ***
## Powerk                                 2.53e-05 ***
## Powerl                                 0.138192
## Powerm                                 0.195543
## Powern                                 0.249869
## Powero                                 0.134409
## CarAge                                 2.77e-10 ***
## DriverAge                               < 2e-16 ***
## BrandJapanese (except Nissan) or Korean  < 2e-16 ***
## BrandMercedes, Chrysler or BMW         0.908787
## BrandOpel, General Motors or Ford      0.153009
## BrandRenault, Nissan or Citroen        0.125280
## BrandVolkswagen, Audi, Skoda or Seat   0.687900
## Brandother                             0.326302
## GasRegular                             1.20e-06 ***
## RegionR23                              0.000579 ***
## RegionR24                              0.034817 *
## RegionR25                              0.528118
## RegionR31                              0.184926
## RegionR52                              0.726691
## RegionR53                              0.679464
## RegionR54                              0.573653
## RegionR72                              0.094449 .
## RegionR74                              0.092947 .
## Density                                4.15e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 126452  on 412411  degrees of freedom
## Residual deviance: 123394  on 412380  degrees of freedom
## AIC: 123458
##
## Number of Fisher Scoring iterations: 6
```

brand and region should be recategorized, the other 6 predictors should be significant.