Choice Modelling

```
#Step 1 - Load the dataset and quickly examine it
minivan<-read.csv("minivanconjoint.csv")
str(minivan, give.attr = FALSE)
```

```
## 'data.frame':    9000 obs. of  9 variables:
##  $ resp.id: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ ques   : int  1 1 1 2 2 2 3 3 3 4 ...
##  $ alt    : int  1 2 3 1 2 3 1 2 3 1 ...
##  $ carpool: Factor w/ 2 levels "no","yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ seat   : int  6 8 6 6 7 6 8 7 8 7 ...
##  $ cargo  : Factor w/ 2 levels "2ft","3ft": 1 2 2 1 2 1 2 2 1 2 ...
##  $ eng    : Factor w/ 3 levels "elec","gas","hyb": 2 3 2 2 2 1 2 1 1 1 ...
##  $ price  : int  35 30 30 30 35 35 35 30 40 40 ...
##  $ choice : int  0 0 1 0 1 0 1 0 0 1 ...
```

```
head(minivan)
```

```
##   resp.id ques alt carpool seat cargo  eng price choice
## 1       1    1   1     yes    6   2ft  gas    35      0
## 2       1    1   2     yes    8   3ft  hyb    30      0
## 3       1    1   3     yes    6   3ft  gas    30      1
## 4       1    2   1     yes    6   2ft  gas    30      0
## 5       1    2   2     yes    7   3ft  gas    35      1
## 6       1    2   3     yes    6   2ft elec    35      0
```

```
str(minivan, give.attr = FALSE)
```

```
## 'data.frame':    9000 obs. of  9 variables:
##  $ resp.id: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ ques   : int  1 1 1 2 2 2 3 3 3 4 ...
##  $ alt    : int  1 2 3 1 2 3 1 2 3 1 ...
##  $ carpool: Factor w/ 2 levels "no","yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ seat   : int  6 8 6 6 7 6 8 7 8 7 ...
##  $ cargo  : Factor w/ 2 levels "2ft","3ft": 1 2 2 1 2 1 2 2 1 2 ...
##  $ eng    : Factor w/ 3 levels "elec","gas","hyb": 2 3 2 2 2 1 2 1 1 1 ...
##  $ price  : int  35 30 30 30 35 35 35 30 40 40 ...
##  $ choice : int  0 0 1 0 1 0 1 0 0 1 ...
```

#Step 2 - Load the required libraries

```
library(mlogit)
```

```
## Loading required package: Formula
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
```

```
##     as.Date, as.Date.numeric
```

```
## Loading required package: lmtest
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

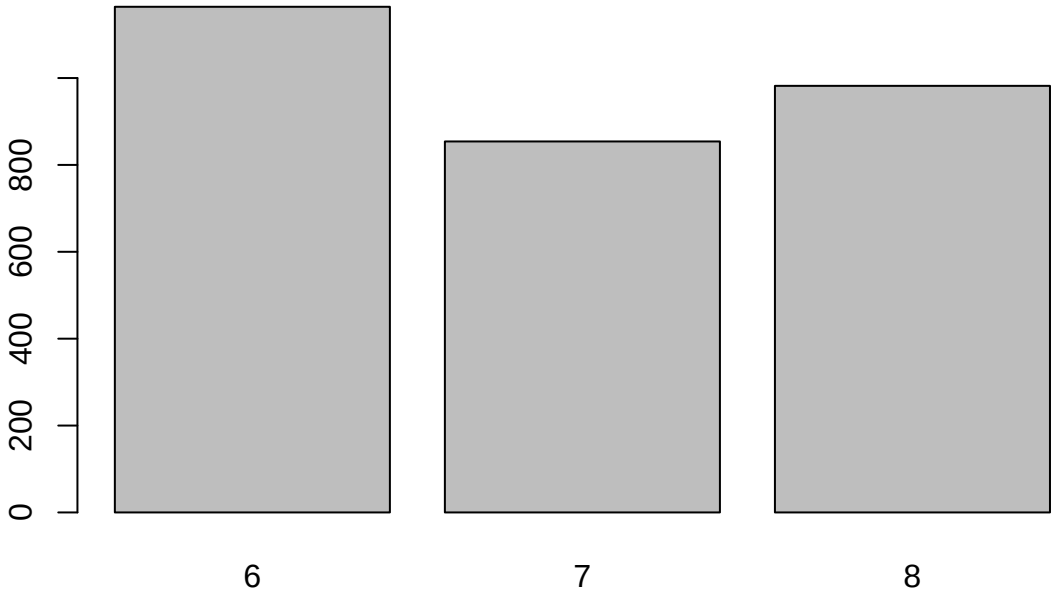
```
library(conjoint)
library(ggplot2)
```

```
minivan$seat <- as.factor(minivan$seat)
```

```
# Print crosstabs of chosen minivan by seats
chosen_by_seats <- xtabs(choice ~ seat, data=minivan)
chosen_by_seats
```

```
## seat
##    6    7    8
## 1164  854  982
```

```
barplot(chosen_by_seats)
```




```
# Print crosstabs of chosen minivan by carpool
chosen_by_carpool <- xtabs(choice ~ carpool, data=minivan)
chosen_by_carpool
```

```
## carpool
##   no  yes
## 2115  885
```

```r
# Print crosstabs of chosen minivan by cargo
chosen_by_cargo <- xtabs(choice ~ cargo, data=minivan)
chosen_by_cargo
```

```
## cargo
##  2ft  3ft
## 1312 1688
```

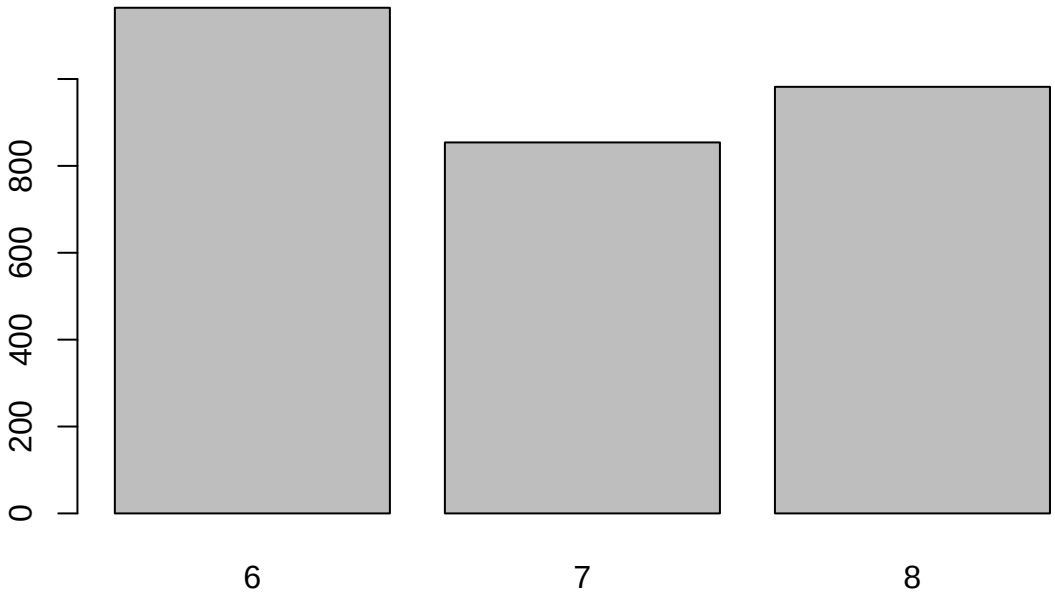
```r
# Print crosstabs of chosen minivan by eng
chosen_by_eng <- xtabs(choice ~ eng, data=minivan)
chosen_by_eng
```

```
## eng
## elec  gas  hyb
##  608 1444  948
```

```r
# Print crosstabs of chosen minivan by price
chosen_by_price <- xtabs(choice ~ price, data=minivan)
chosen_by_price
```

```
## price
##   30   35   40
## 1486  956  558
```

```r
# Plot the chosen_by_trans object
barplot(chosen_by_seats)
```

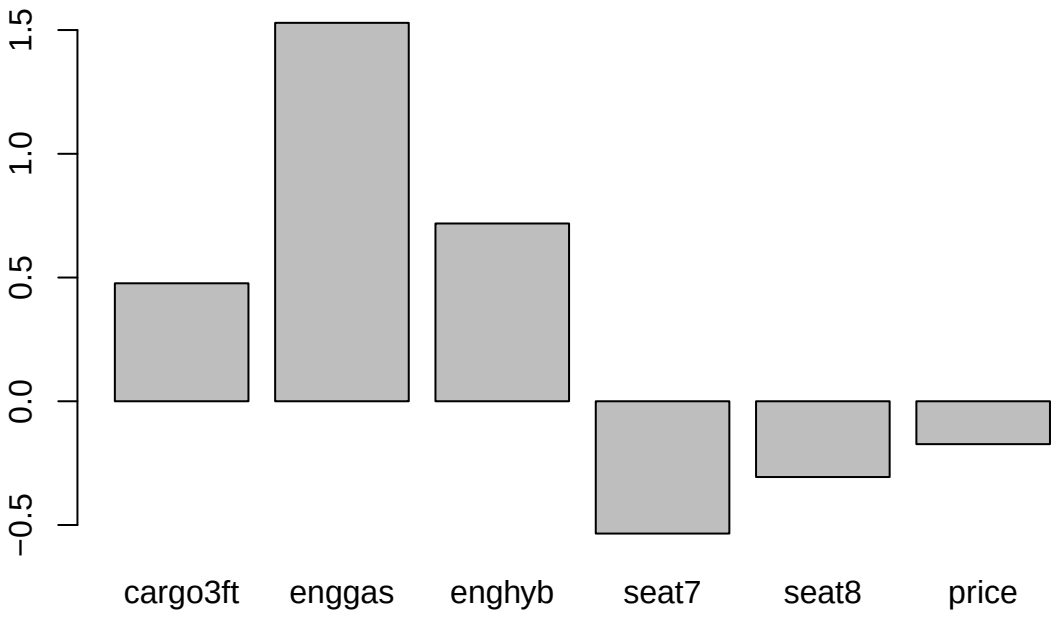



```r
# load the mlogit library
library(mlogit)

minivanm1 <- mlogit(choice ~ 0 + cargo +eng +seat + price, data=minivan, alt.var="alt", choice = "choic
# summarize the minivanm1 object to see the output
summary(minivanm1)
```

```
##
## Call:
## mlogit(formula = choice ~ 0 + cargo + eng + seat + price, data = minivan,
##     alt.var = "alt", choice = "choice", method = "nr")
```

```
##
## Frequencies of alternatives:
##       1       2       3
## 0.32700 0.33467 0.33833
##
## nr method
## 5 iterations, 0h:0m:0s
## g'(-H)^-1g = 8E-05
## successive function values within tolerance limits
##
## Coefficients :
##            Estimate Std. Error  z-value  Pr(>|z|)
## cargo3ft  0.4766936  0.0508632   9.3721 < 2.2e-16 ***
## enggas    1.5291247  0.0673982  22.6879 < 2.2e-16 ***
## enghyb    0.7183908  0.0654963  10.9684 < 2.2e-16 ***
## seat7    -0.5345392  0.0623518  -8.5730 < 2.2e-16 ***
## seat8    -0.3061074  0.0611184  -5.0084 5.488e-07 ***
## price    -0.1733053  0.0069398 -24.9726 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-Likelihood: -2582.1
```

```r
barplot(coef(minivanm1))
```



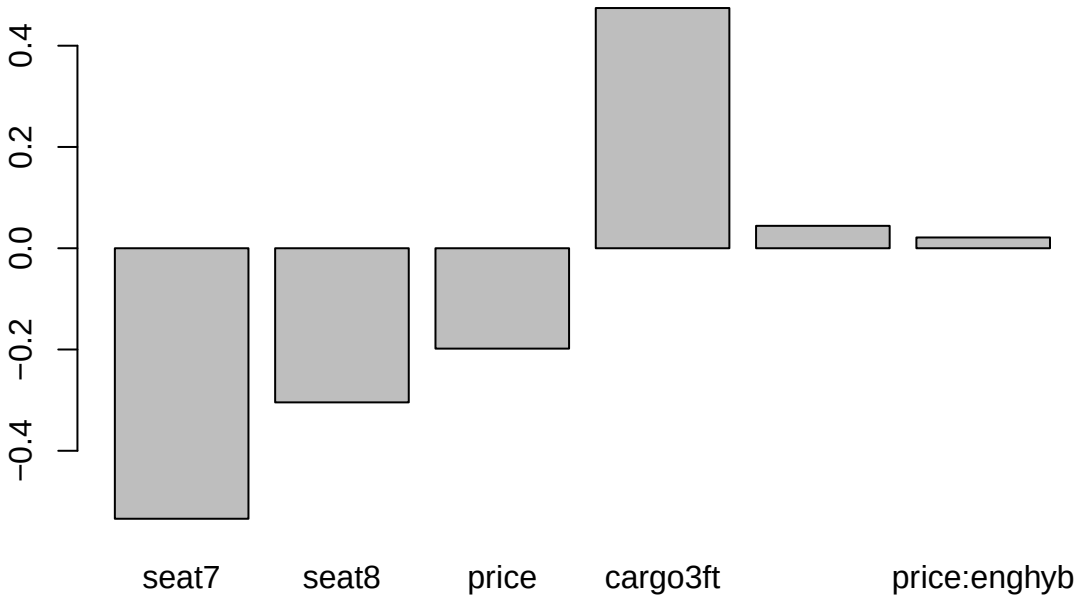
willingness to pay

```r
# Convert data to mlogit.data, varying = 4:8 indicates the attribute columns.
minivan.ml <- mlogit.data(minivan, shape = 'long', choice = 'choice', alt.var = 'alt', varying = 4:8)
```

```r
minivan_model <- mlogit(choice ~ 0 + seat + price +cargo + price:eng, data = minivan.ml)

# summarize the m1 object to see the output
summary(minivan_model)
```

```
##
## Call:
## mlogit(formula = choice ~ 0 + seat + price + cargo + price:eng,
##     data = minivan.ml, method = "nr")
##
## Frequencies of alternatives:
##       1       2       3
## 0.32700 0.33467 0.33833
##
## nr method
## 5 iterations, 0h:0m:0s
## g'(-H)^-1g = 0.000204
## successive function values within tolerance limits
##
## Coefficients :
##                 Estimate Std. Error  z-value  Pr(>|z|)
## seat7        -0.5342749  0.0622069  -8.5887 < 2.2e-16 ***
## seat8        -0.3044073  0.0610012  -4.9902 6.032e-07 ***
## price        -0.1982535  0.0073294 -27.0491 < 2.2e-16 ***
## cargo3ft      0.4744273  0.0507311   9.3518 < 2.2e-16 ***
## price:enggas  0.0442660  0.0019744  22.4203 < 2.2e-16 ***
## price:enghyb  0.0212594  0.0019422  10.9463 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-Likelihood: -2585.7
```

```
barplot(coef(minivan_model))
```




```
WTPminivan <- coef(minivan_model)/- coef(minivan_model)[3]
WTPminivan
```

```
##        seat7        seat8        price     cargo3ft price:enggas price:enghyb
##   -2.6949078   -1.5354450   -1.0000000    2.3930338    0.2232796    0.1072335
```

```r
## Predict choice share based on model
predict_mnl <- function(model, products) {
  data.model <- model.matrix(update(model$formula, 0 ~ .),
                             data = products)[,-1]
  utility <- data.model%*%model$coef
  share <- exp(utility)/sum(exp(utility))
  cbind(share, products)
}
```

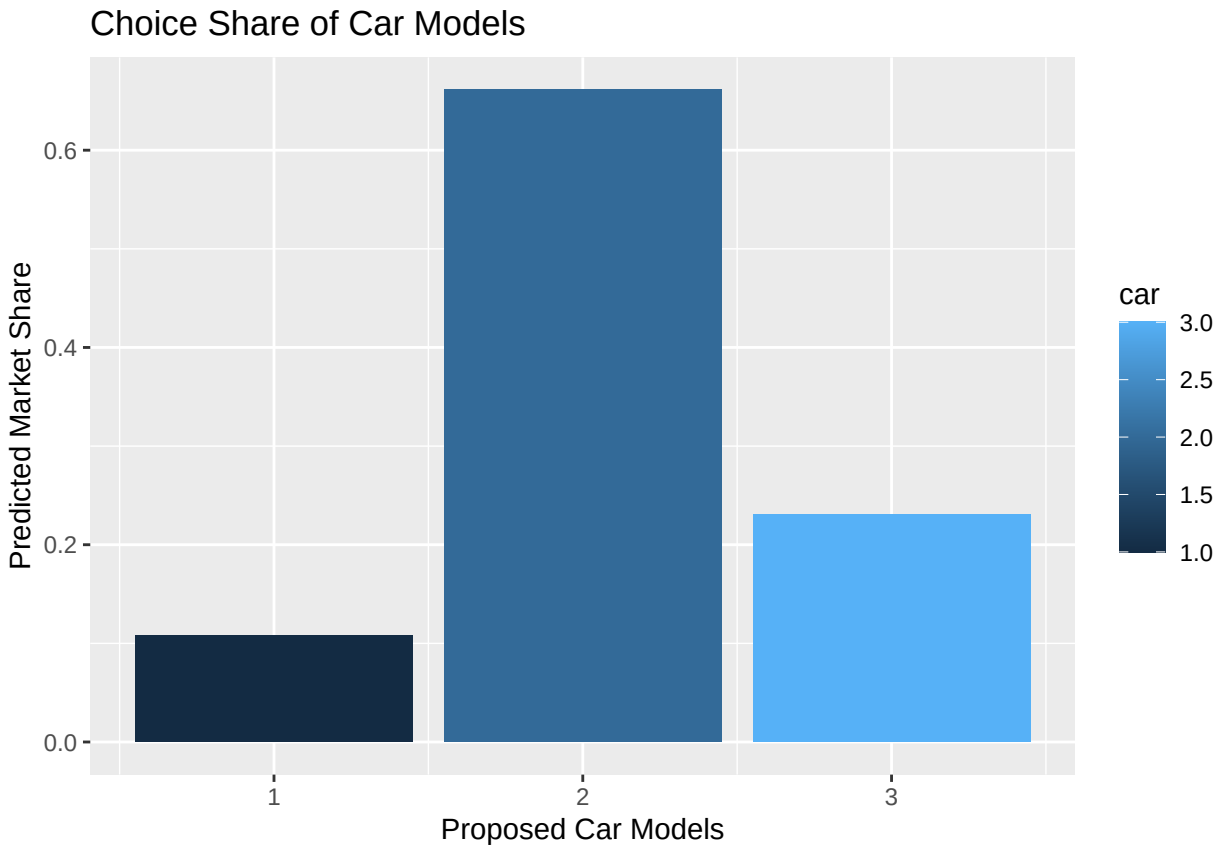
```r
# Create hypothetical data for choice share prediction
car <- c(1,2,3)
price <- c(40, 30, 35)
seat <- factor(c(8, 6, 7), levels=c(6,7,8))
eng <- factor(c("elec", "elec","elec"), levels=c("elec", "gas","hyb"))
cargo <- factor(c("3ft", "2ft", "3ft"), levels=c("2ft", "3ft"))
carpool <- factor(c("no", "no", "no"), levels=c("no", "yes"))
prod <- data.frame(car,seat, carpool,cargo, price, eng)
prod
```

```
##   car seat carpool cargo price  eng
## 1   1    8      no   3ft    40 elec
## 2   2    6      no   2ft    30 elec
## 3   3    7      no   3ft    35 elec
```

```r
# Predict choice shares of hypothetical 3-option sports car selection
shares <- predict_mnl(minivan_model, prod)
shares
```

```
##       share car seat carpool cargo price  eng
## 1 0.1079085   1    8      no   3ft    40 elec
## 2 0.6610299   2    6      no   2ft    30 elec
## 3 0.2310615   3    7      no   3ft    35 elec
```

```r
#plot it to see it visually
ggplot(shares, aes(x = car, y = share, fill = car))+
  geom_bar(stat = 'identity')+
  ylab('Predicted Market Share')+
  xlab('Proposed Car Models')+
  ggtitle('Choice Share of Car Models')
```



Hierarchical Modeling

```
# Step 4 - setting up the data for the hierarchical choice model,
# Prepare data for hierarchical model by recoding the non-binary factors
# first run the model to add parameters
minivan_model2 <- mlogit(choice ~ 0 + seat + price + cargo + price:eng, data = minivan.ml)
minivan.ml2 <- minivan.ml <- mlogit.data(minivan, shape = 'long', choice = 'choice', alt.var = 'alt', va
# Set the contrasts for non-binary factor variables to code against the
# effects of the baseline level
contrasts(minivan.ml2$eng) <- contr.sum(levels(minivan.ml2$eng))
dimnames(contrasts(minivan.ml2$eng))[[2]] <- levels(minivan.ml2$eng)[1:2]
contrasts(minivan.ml2$seat) <- contr.sum(levels(minivan.ml2$seat))
dimnames(contrasts(minivan.ml2$seat))[[2]] <- levels(minivan.ml2$seat)[1:2]

# Create character vector of "n" for every independent variable in the model,
# which will let the hierarchical model know where to use a distribution
my_rpar <- rep("n", length(minivan_model2$coef))
names(my_rpar) <- names(minivan_model2$coef)

# Fit a hierarchical model that assumes a heterogeneity of preferences,
# fit a model with the assumption that there may be correlations between
# variables. This is stored into sportcar_model2 to keep it separate.
minivan_model3 <- mlogit(choice ~ 0 + seat + price + cargo + price:eng, data = minivan.ml, panel = TRUE
#This time, we add the following arguments to the mlogit function:
#panel: We specify this to be TRUE, as we want to use panel techniques to specify that each simulated r
#rpar: We set rpar equal to our custom variable "my_rpar," which is a vector coded with 'n' for every c
#correlation = We set this to TRUE to take into account any correlations that may exist between indepen
```

```r
# Step 5 - Now let's look at the result of the revised model
# Generate model summary
summary(minivan_model3)
```

```
##
## Call:
## mlogit(formula = choice ~ 0 + seat + price + cargo + price:eng,
##     data = minivan.ml, rpar = my_rpar, correlation = TRUE, panel = TRUE)
##
## Frequencies of alternatives:
##       1       2       3
## 0.32700 0.33467 0.33833
##
## bfgs method
## 12 iterations, 0h:0m:16s
## g'(-H)^-1g = 0.769
## last step couldn't find higher value
##
## Coefficients :
##                                 Estimate Std. Error  z-value  Pr(>|z|)
## seat7                         -0.7126280  0.0697582 -10.2157 < 2.2e-16 ***
## seat8                         -0.5309296  0.0749733  -7.0816 1.425e-12 ***
## price                         -0.2574905  0.0098786 -26.0654 < 2.2e-16 ***
## cargo3ft                       0.6300717  0.0613985  10.2620 < 2.2e-16 ***
## price:enggas                   0.0575822  0.0026182  21.9932 < 2.2e-16 ***
## price:enghyb                   0.0277396  0.0024294  11.4185 < 2.2e-16 ***
## chol.seat7:seat7              -0.4913997  0.0924937  -5.3128 1.080e-07 ***
## chol.seat7:seat8              -1.0382951  0.1022564 -10.1538 < 2.2e-16 ***
## chol.seat8:seat8               0.3035667  0.0937059   3.2396  0.001197 **
## chol.seat7:price              -0.0068171  0.0107322  -0.6352  0.525297
## chol.seat8:price               0.0398546  0.0126289   3.1558  0.001600 **
## chol.price:price              -0.0733678  0.0147378  -4.9782 6.418e-07 ***
## chol.seat7:cargo3ft           -0.1106496  0.0774929  -1.4279  0.153330
## chol.seat8:cargo3ft           -0.3902190  0.0891140  -4.3789 1.193e-05 ***
## chol.price:cargo3ft           -0.0718572  0.1023781  -0.7019  0.482754
## chol.cargo3ft:cargo3ft         0.0901587  0.1012432   0.8905  0.373189
## chol.seat7:price:enggas        0.0045906  0.0029047   1.5804  0.114008
## chol.seat8:price:enggas       -0.0028548  0.0034892  -0.8182  0.413248
## chol.price:price:enggas        0.0270267  0.0040832   6.6190 3.616e-11 ***
## chol.cargo3ft:price:enggas     0.0203275  0.0038589   5.2677 1.382e-07 ***
## chol.price:enggas:price:enggas 0.0217072  0.0038569   5.6282 1.821e-08 ***
## chol.seat7:price:enghyb        0.0023082  0.0029104   0.7931  0.427737
## chol.seat8:price:enghyb        0.0069120  0.0034277   2.0165  0.043749 *
## chol.price:price:enghyb        0.0225582  0.0038685   5.8313 5.500e-09 ***
## chol.cargo3ft:price:enghyb     0.0201817  0.0041753   4.8336 1.341e-06 ***
## chol.price:enggas:price:enghyb 0.0104794  0.0040081   2.6145  0.008935 **
## chol.price:enghyb:price:enghyb -0.0046820  0.0039811  -1.1761  0.239568
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-Likelihood: -2506.8
##
## random coefficients
##              Min.      1st Qu.      Median        Mean     3rd Qu. Max.
```

```
## seat7        -Inf -1.044072048 -0.71262797 -0.71262797 -0.38118389  Inf
## seat8        -Inf -1.260567105 -0.53092957 -0.53092957  0.19870796  Inf
## price        -Inf -0.313993681 -0.25749054 -0.25749054 -0.20098740  Inf
## cargo3ft     -Inf  0.345659079  0.63007168  0.63007168  0.91448429  Inf
## price:enggas -Inf  0.030233562  0.05758221  0.05758221  0.08493085  Inf
## price:enghyb -Inf  0.005359018  0.02773961  0.02773961  0.05012021  Inf
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

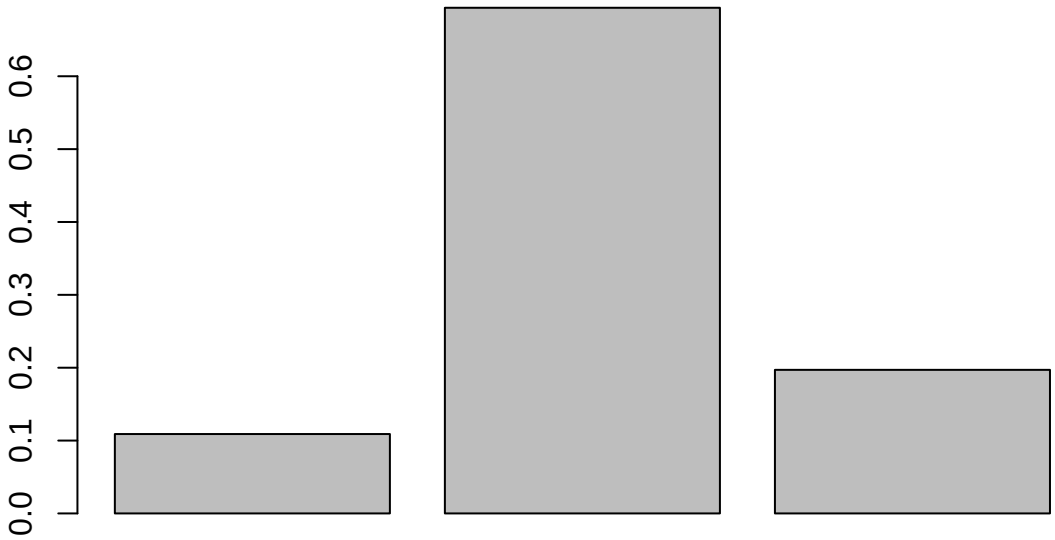
```
## Build share prediction function
coef_means <- minivan_model3$coef[1:6]
Sigma <- cov.mlogit(minivan_model3)

model3_coded <- model.matrix(update(minivan_model3$formula, 0 ~ .), data = prod)[,-1]

share <- matrix(NA, nrow=1000, ncol=nrow(model3_coded))

# Compute a random draw of 1,000 buyers
for (i in 1:1000) {
  # Draw a coefficient vector from the normal distribution
  coef <- mvrnorm(1, mu=coef_means, Sigma=Sigma)
  # Compute utilities for those coef
  utility <- model3_coded %*% coef
  # Compute probabilites according to logit formula
  share[i,] <- exp(utility) / sum(exp(utility))
}

# examine the output of the N=1000 simulation.the order of results may be
# different for your output since this is a simulation.However, the column
# means calculated in the next step will be close to the same.

head(share)
```

```
##             [,1]      [,2]       [,3]
## [1,] 0.004421212 0.9211198 0.07445899
## [2,] 0.254117638 0.5260409 0.21984150
## [3,] 0.086715280 0.6814118 0.23187295
## [4,] 0.118762705 0.5228955 0.35834183
## [5,] 0.062514975 0.7634624 0.17402266
## [6,] 0.042771754 0.8201295 0.13709870
```

```
# examine the column means to see if car 1 beats car 2 and car 3.
new = colMeans(share)
new
```

```
## [1] 0.1089500 0.6939964 0.1970536
```

```
#plot it to see it visually
barplot(new)
```


0.0
0.1
0.2
0.3
0.4
0.5
0.6