Tarea 3

Marco Antonio Ramos

12/6/2020

Ambiente de trabajo.

```
library(readstata13)
library(dplyr)
library(knitr)
library(fastDummies)
library(nnet)
library(mlogit)

data<-read.dta13("Stove.dta")
```

1.

```
data_1<- data %>% select(stove, income, age, num_people, region)
kable(summary(data_1))
```

stove	income	age	num_people	region
gc:573	Min. :2.000	Min. :20.00	Min. :2.000	valley:177
gr:129	1st Qu.:3.000	1st Qu.:30.00	1st Qu.:3.000	scostl:361
ec: 64	Median :5.000	Median :45.00	Median :4.000	mountn:102
er: 84	Mean :4.641	Mean :42.94	Mean :4.424	ncostl:260
hp: 50	3rd Qu.:6.000	3rd Qu.:55.00	3rd Qu.:6.000	NA
NA	Max. :7.000	Max. :65.00	Max. :7.000	NA

2.

Para este ejercicio, creo 4 dataframes y luego los junto.

```
#Expando en dummies y luego colapso la base de datos
data_2<- data%>%
  select(stove,region)%>%
  dummy_cols("stove")%>%
  select(-stove)%>%
```

```
  group_by(region)%>%
   summarise_each(funs(sum))%>%
   mutate(total = rowSums(.[2:6]))

#Divido cada valor entre el total para obtener el porcentaje
for (i in 2:7) {
  data_2[i]<-data_2[i]/data_2[7]
}

#Transpongo mis datos
respuesta_2 <- as.data.frame(t(data_2[,-1]))
colnames(respuesta_2) <- data_2$region

kable(respuesta_2)
```

	valley	scostl	mountn	ncostl
stove_gc	0.6045198	0.6121884	0.5784314	0.7153846
stove_gr	0.1525424	0.1606648	0.1666667	0.1038462
stove_ec	0.0734463	0.0692521	0.0784314	0.0692308
stove_er	0.1016949	0.1024931	0.1078431	0.0692308
stove_hp	0.0677966	0.0554017	0.0686275	0.0423077
total	1.0000000	1.0000000	1.0000000	1.0000000

3.

Ahota estimo el modelo multinomial.[1]

```
#Ahora la estufa hp es mi base
model_3 <- multinom(factor(stove) ~ factor(region), data = data)
```

```
## # weights:  25 (16 variable)
## initial  value 1448.494121
## iter  10 value 1025.042195
## iter  20 value 1016.150702
## final  value 1016.147753
## converged
```

```
model_3
```

```
## Call:
## multinom(formula = factor(stove) ~ factor(region), data = data)
##
## Coefficients:
##    (Intercept) factor(region)scostl factor(region)mountn factor(region)ncostl
## gr   -1.377068          0.039350341           0.13273681           -0.5528996
## ec   -2.107766         -0.071420999           0.10956737           -0.2275815
```

[1]Nota: se uso el comando multinom porque te permite hacer el modelo sin la necesidad de transformar tus datos. Asimismo, los comandos tradicionales para hacer tablas del modelo no pueden interpretar ni multinom ni mlogit de manera correta por lo que solo se puso el output a través de *summary*

```
## er   -1.782446         -0.004751305           0.10287785           -0.5528914
## hp   -2.187985         -0.214514840           0.05630972           -0.6400025
##
## Residual Deviance: 2032.296
## AIC: 2064.296
```

Finalmente para "recuperar" el porcentaje simplemente calculo $\frac{e^{X_i'\beta_{j,i}}}{1+\sum_{l=1}^{J} e^{X_i'\beta_{l.i}}}$ donde j se refiere a las distintas opciones de nuestra variable objetivo(estufas) y l se refiere a cada estufa excepto la que use como base. En este sentido podría recrear la tabla haciendo el calculo para cada combinación. Por ejemplo, recreamos la probabilidad conidiconal de tener una estufa *gr* dado que estamos en la región *mountain*

```
exp(0.07636369+0.8107438)/(1+exp(0.07636369+0.8107438)
                           +exp(-0.05626617+2.1877087)+ exp(0.05356228+.0797629)
                           + exp(0.04664417+0.4052078))
```

```
## [1] 0.1666624
```

4^2

```
#En primer lugar, adecuó los datos en el formato requerido por mlogit
wide_data<-mlogit.data(data, choice = "stove", shape = "wide", varying = 3:12, sep = "_")
```

```
model_4 <- mlogit(
  factor(stove) ~  oc + ic | factor(region) + age + income  + num_people , wide_data)

model_4
```

```
##
## Call:
## mlogit(formula = factor(stove) ~ oc + ic | factor(region) + age +     income + num_people, data = wi
##
## Coefficients:
##          (Intercept):er           (Intercept):gc           (Intercept):gr
##               1.6571202                0.4448721               -0.3956144
##          (Intercept):hp                       oc                       ic
##              -0.7415765               -0.0069541               -0.0015138
## factor(region)scostl:er  factor(region)scostl:gc  factor(region)scostl:gr
##               0.0108250                0.0450527                0.0694040
## factor(region)scostl:hp  factor(region)mountn:er  factor(region)mountn:gc
##              -0.1749933               -0.1126465               -0.1210075
## factor(region)mountn:gr  factor(region)mountn:hp  factor(region)ncostl:er
##              -0.0052629               -0.0750777               -0.3502491
## factor(region)ncostl:gc  factor(region)ncostl:gr  factor(region)ncostl:hp
##               0.2488076               -0.3271693               -0.3919262
##                  age:er                   age:gc                   age:gr
##              -0.0257541               -0.0043358               -0.0012827
##                  age:hp                income:er                income:gc
```

[2]Nota: los comandos tradicionales para hacer tablas del modelo no pueden interpretar objetos mlogit de manera correta por lo que solo se puso el output a través de *summary*.

```
##              -0.0193964              -0.0380605              -0.0068375
##               income:gr               income:hp           num_people:er
##              -0.1168082               0.0594682              -0.0257637
##           num_people:gc           num_people:gr           num_people:hp
##              -0.0522045              -0.0684133              -0.0438382
```

5.

Para esta pregunta se usará una aproximación exacta

```
#Primero imprimo las probabilidades calculadas para cada observación
#El comando predict ya me da la probabillidad final
predicciones<-as.data.frame(predict(model_4,newdata=wide_data))

#Ahora calculo las probabilidades calculadas para cada observacion pero aumentando
#en dos individuos la variable num_people
#Para esto tengo que volver a transformar la base de datos para que sea
#acorde a mlogit.
data_5<-data %>% mutate (num_people=num_people+2)
wide_data_5<-mlogit.data(data_5, choice = "stove", shape = "wide", varying = 3:12, sep = "_")
predicciones_5<-as.data.frame(predict(model_4,newdata=wide_data_5))
compare_5<-as.data.frame(cbind(predicciones$er,predicciones_5$er))
names(compare_5)[1] <- "original"
names(compare_5)[2] <- "modificado"

#Finalmente calculo la diferencia y lo promedio
compare_5<-compare_5%>% mutate(dif=modificado-original)
sum(compare_5$dif)/900
```

```
## [1] 0.004032669
```

El efecto parcial promedio (calculado mediante una estimación exacta) es de 0.004032669. Esto quiere decir que ante un aumento en dos personas en el número de habitants del hogar, la probabilidad de tener una estufa er aumenta en .4 puntos porcentuales.

6.

Para calcular la bondad del modelo simplemente hay que comparar cuantas deciciones logra explicar de manera correcta vs en cuantas se equivoca.

Para realizar la predicción simplemente le asigno a mi predicción el valor de la estufa con la mayor probabilidad dado las Xs.

```
#Primero imprimo las probabilidades calculadas para cada observación
predicciones<-as.data.frame(predict(model_4,newdata=wide_data))

#Segundo, asumo que la estufa con la mayor probabilidad es la predicción
pred<-as.data.frame(colnames(predicciones)[max.col(predicciones, ties.method = "first")])

#Junto la predicción y el valor real
compare<-cbind(pred,data$stove)
```

```r
#Finalmente, cuento los casos de predicción exitosa

#Antes de eso manipulo debo asignarle el formato adecuado a mis datos.
names(compare)[1] <- "pred"
names(compare)[2] <- "real"
compare$pred<-as.character(compare$pred)
compare$real<-as.character(compare$real)

compare<-compare%>%
         mutate(exito=ifelse(compare$pred==compare$real,1,0))

sum(compare$exito)/900
```

```
## [1] 0.6366667
```

El modelo predijo exitosamente el 63% de las observaciones.