

Ch12_2e_ChapterExamples

```
require(knitr)
require(haven)
#library(probitmfx) #install.packages("probitmfx")
library(AER)
library(apsrtable)
library(tidyverse)

opts_chunk$set(echo = TRUE)
options(digits = 3)
opts_knit$set(root.dir =
"C:/Users/baileyma/Dropbox/Bailey_RealStats_RealEconometrics_2e_Supplements_March2021/ChapterExamples_R

# Remove objects from the previous session
rm(list = ls(all = TRUE))
```

```
dog          = read_dta("Ch12_ChapterExample_DogPolitics.dta")
DogProb      = glm(obamavote~ dog + as.numeric(libcon), family = binomial(link = "probit"), na.action = na.omit)
summary(DogProb)
```

Preparation

```
##
## Call:
## glm(formula = obamavote ~ dog + as.numeric(libcon), family = binomial(link = "probit"),
##      data = dog, na.action = na.exclude)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -3.306  -0.526   0.247   0.630   2.833
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      3.3861    0.0486   69.74  <2e-16 ***
## dog             -0.2126    0.0249  -8.53  <2e-16 ***
## as.numeric(libcon) -0.7527    0.0105 -71.68  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 21517  on 15595  degrees of freedom
```

```
## Residual deviance: 13460  on 15593  degrees of freedom
## (3638 observations deleted due to missingness)
## AIC: 13466
##
## Number of Fisher Scoring iterations: 5
```

```
#probitmfx(obamavote~ dog + as.numeric(libcon), data=dog, atmean= FALSE)
```

```
## Continuous variable, observed-value approach, discrete change
## ProbFit = fitted(DogProb) # Missing data can mess up fitted values unless use na.exclude option
P1b = pnorm(DogProb$coef[1] + DogProb$coef[2]*dog$dog + DogProb$coef[3]*as.numeric(dog$libcon))
P2 = pnorm(DogProb$coef[1] + DogProb$coef[2]*dog$dog + DogProb$coef[3]*(as.numeric(dog$libcon)+1))
mean(P2[is.na(dog$obamavote)==0]-P1b[is.na(dog$obamavote)==0], na.rm=T)
```

```
## [1] -0.18
```

```
# marginal effects
Marg = DogProb$coef[3]* dnorm(DogProb$coef[1] + DogProb$coef[2]*dog$dog + DogProb$coef[3]*as.numeric(dog$libcon))
mean(Marg[is.na(dog$obamavote)==0], na.rm=T)
```

```
## [1] -0.183
```

```
Marg[1:10]
```

```
## dog dog dog dog dog dog dog dog dog dog
## -0.280 NA -0.296 -0.122 -0.159 -0.296 -0.159 -0.159 -0.296 -0.280
```

```
## Dummy variable, observed-value approach, discrete change
P0 = pnorm(DogProb$coef[1] + DogProb$coef[2]*0 + DogProb$coef[3]*as.numeric(dog$libcon))
P1 = pnorm(DogProb$coef[1] + DogProb$coef[2]*1 + DogProb$coef[3]*as.numeric(dog$libcon))
#mean(P1[is.na(ProbFit)==0]-P0[is.na(ProbFit)==0], na.rm=T)
```

```
# Logit
DogLog = glm(obamavote~ dog + as.numeric(libcon),
             family = binomial(link = "logit"),
             na.action= na.exclude,
             data=dog)
summary(DogLog)
```

```
##
## Call:
## glm(formula = obamavote ~ dog + as.numeric(libcon), family = binomial(link = "logit"),
## data = dog, na.action = na.exclude)
##
## Deviance Residuals:
## Min 1Q Median 3Q Max
## -3.053 -0.508 0.266 0.598 2.720
##
## Coefficients:
## Estimate Std. Error z value Pr(>|z|)
```

```

## (Intercept)          5.9768      0.0951    62.85    <2e-16 ***
## dog                  -0.3680      0.0436    -8.44    <2e-16 ***
## as.numeric(libcon)  -1.3263      0.0207   -63.98    <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##    Null deviance: 21517  on 15595  degrees of freedom
## Residual deviance: 13406  on 15593  degrees of freedom
## (3638 observations deleted due to missingness)
## AIC: 13412
##
## Number of Fisher Scoring iterations: 5

P1 = fitted(DogLog)
P1b = exp(DogLog$coef[1] + DogLog$coef[2]*dog$dog + DogLog$coef[3]*as.numeric(dog$libcon)) / (1+ exp(DogLog$coef[2]*dog$dog + DogLog$coef[3]*as.numeric(dog$libcon)))
P2 = exp(DogLog$coef[1] + DogLog$coef[2]*dog$dog + DogLog$coef[3]*(as.numeric(dog$libcon)+1)) / (1+ exp(DogLog$coef[2]*dog$dog + DogLog$coef[3]*(as.numeric(dog$libcon)+1)))
mean(P2-P1, na.rm=T)          #mean(P1b-P1, na.rm=T)

## [1] -0.181

c(min(P1, na.rm=T), max(P1, na.rm=T))

## [1] 0.0247 0.9905

## Dummy variable, observed-value approach, discrete change
P0 = exp(DogLog$coef[1] + DogLog$coef[2]*0 + DogLog$coef[3]*as.numeric(dog$libcon)) / (1+ exp(DogLog$coef[2]*dog$dog + DogLog$coef[3]*as.numeric(dog$libcon)))
P1 = exp(DogLog$coef[1] + DogLog$coef[2]*1 + DogLog$coef[3]*as.numeric(dog$libcon)) / (1+ exp(DogLog$coef[2]*dog$dog + DogLog$coef[3]*as.numeric(dog$libcon)))

#LPM
DogLPM = lm(obamavote~ dog + as.numeric(libcon), na.action= na.exclude, data=dog)
summary(DogLPM)

##
## Call:
## lm(formula = obamavote ~ dog + as.numeric(libcon), data = dog,
##     na.action = na.exclude)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.2137 -0.1770 -0.0063  0.2512  1.0805
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.42098    0.00889   159.93 < 2e-16 ***
## dog           -0.05022    0.00617    -8.14 4.3e-16 ***
## as.numeric(libcon) -0.20733    0.00196  -106.00 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.378 on 15593 degrees of freedom

```

```
## (3638 observations deleted due to missingness)
## Multiple R-squared: 0.423, Adjusted R-squared: 0.423
## F-statistic: 5.72e+03 on 2 and 15593 DF, p-value: <2e-16
```

```
max(fitted(DogLPM), na.rm=T)
```

```
## [1] 1.21
```

```
min(fitted(DogLPM), na.rm=T)
```

```
## [1] -0.0805
```

```
# Standardized
```

```
DogLPMStd = lm(obamavote~ dog + scale(as.numeric(libcon)), na.action= na.exclude, data=dog)
summary(DogLPMStd)
```

```
##
## Call:
## lm(formula = obamavote ~ dog + scale(as.numeric(libcon)), data = dog,
##     na.action = na.exclude)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.2137 -0.1770 -0.0063  0.2512  1.0805
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.56262    0.00395  142.40 < 2e-16 ***
## dog             -0.05022    0.00617   -8.14 4.3e-16 ***
## scale(as.numeric(libcon)) -0.31516    0.00297 -106.00 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.378 on 15593 degrees of freedom
## (3638 observations deleted due to missingness)
## Multiple R-squared: 0.423, Adjusted R-squared: 0.423
## F-statistic: 5.72e+03 on 2 and 15593 DF, p-value: <2e-16
```

```
sqrt(var(as.numeric(dog$libcon), na.rm=T))
```

```
## [1] 1.52
```

```
coefficients(DogLPM)[3]* sqrt(var(as.numeric(dog$libcon), na.rm=T))
```

```
## as.numeric(libcon)
##                -0.315
```

```
table(dog$libcon)
```

```
##
##      1      2      3      4      5      6      7
## 588 3005 2206 5974 2610 3936 722
```

```
table(as.numeric(dog$libcon))
```

```
##
##      1      2      3      4      5      6      7
## 588 3005 2206 5974 2610 3936 722
```

```
# Fitted values
```

```
coefficients(DogLPM)[1] + 1*coefficients(DogLPM)[2] + 7*coefficients(DogLPM)[3]
```

```
## (Intercept)
##      -0.0805
```

```
coefficients(DogLPM)[1] + 0*coefficients(DogLPM)[2] + 1*coefficients(DogLPM)[3]
```

```
## (Intercept)
##      1.21
```

```
# Displays a table in Latex form
```

```
apsrtable(DogLPM, DogProb, DogLog, model.names= c("LPM", "Probit", "Logit"),
  coef.names=c("Constant", "Dog owner", "Ideology"), digits=3)
```

```
## \begin{table}[!ht]
## \caption{}
## \label{}
## \begin{tabular}{l D{.}{.}{3}D{.}{.}{3}D{.}{.}{3} }
## \hline
## & \multicolumn{1}{c}{ LPM } & \multicolumn{1}{c}{ Probit } & \multicolumn{1}{c}{ Logit }
## % & LPM & Probit & Logit \\
## Constant & 1.421 ^* & 3.386 ^* & 5.977 ^* \\
## & (0.009) & (0.049) & (0.095) \\
## Dog owner & -0.050 ^* & -0.213 ^* & -0.368 ^* \\
## & (0.006) & (0.025) & (0.044) \\
## Ideology & -0.207 ^* & -0.753 ^* & -1.326 ^* \\
## & (0.002) & (0.011) & (0.021) \\
## $N$ & 15596 & 15596 & 15596 \\
## $R^2$ & 0.423 & & \\
## adj. $R^2$ & 0.423 & & \\
## Resid. sd & 0.378 & & \\
## AIC & & 13465.748 & 13411.950 \\
## BIC & & 13557.605 & 13503.807 \\
## $\log L$ & & -6720.874 & -6693.975 \\
## \multicolumn{4}{l}{\footnotesize{Standard errors in parentheses}} \\
## \multicolumn{4}{l}{\footnotesize{$^*$ indicates significance at $p < 0.05$}} \\
## \end{tabular}
## \end{table}
```

```

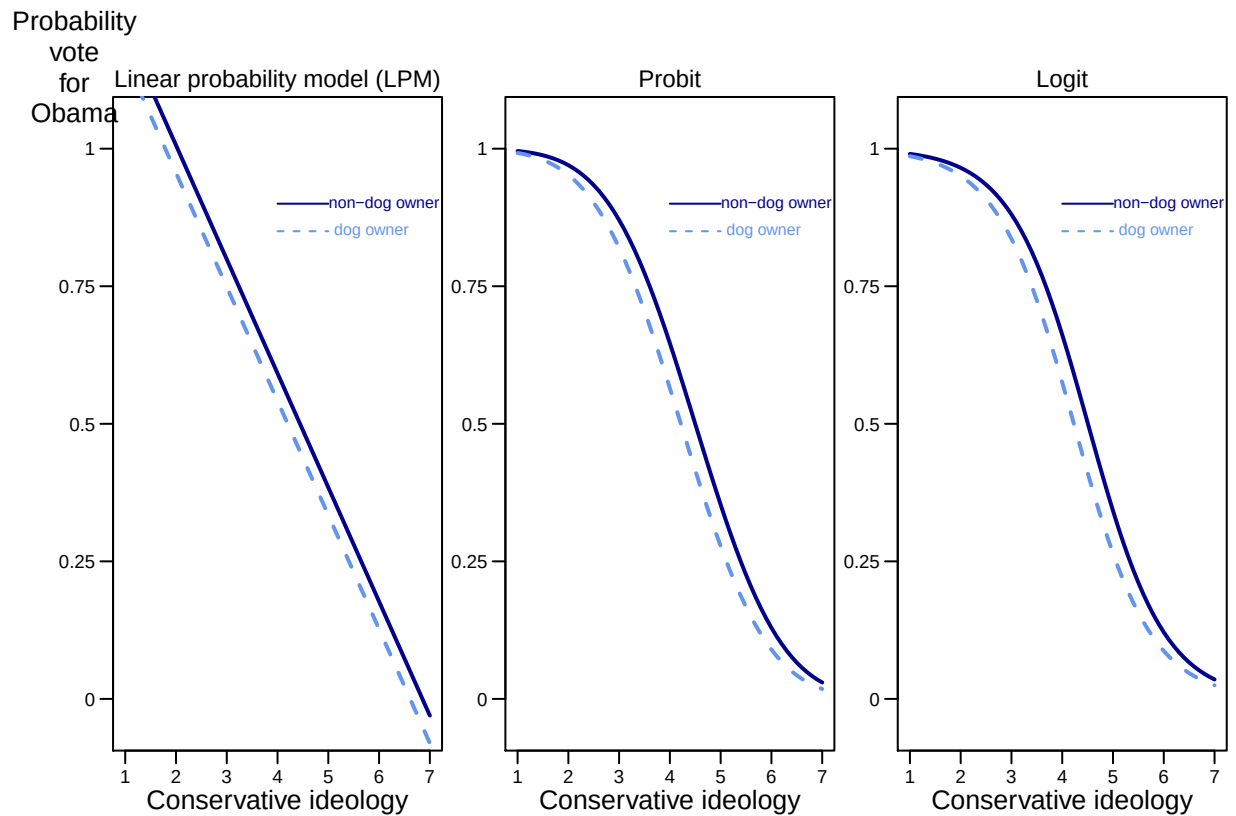
LibConJitter      = as.numeric(dog$libcon) + rnorm(length(dog$libcon))/3
LibConSpan        = seq(1, 7, by = 0.1)
ProbFitLineDog0   = pnorm(coef(DogProb)[1] + coef(DogProb)[2]*0 + coef(DogProb)[3]*LibConSpan )
ProbFitLineDog1   = pnorm(coef(DogProb)[1] + coef(DogProb)[2]*1 + coef(DogProb)[3]*LibConSpan )
LogFitLineDog0    = 1/(1+exp(-(coef(DogLog)[1] + coef(DogLog)[2]*0 + coef(DogLog)[3]*LibConSpan )))
LogFitLineDog1    = 1/(1+exp(-(coef(DogLog)[1] + coef(DogLog)[2]*1 + coef(DogLog)[3]*LibConSpan )))
LPMFitLineDog0    = coef(DogLPM)[1] + coef(DogLPM)[2]*0 + coef(DogLPM)[3]*LibConSpan
LPMFitLineDog1    = coef(DogLPM)[1] + coef(DogLPM)[2]*1 + coef(DogLPM)[3]*LibConSpan

par(mfrow=c(1, 3), mar=c(2.0, 2, 1.3,0.5), oma=c(2, 2.5, 3., 0.3)) # mar(south, west, north, east)
plot(as.numeric(dog$libcon), fitted(DogLPM), ylim=c(-0.05, 1.05), xlim=c(1, 7), type = "n", xlab="", x
      mtext("Probability\nvote\nfor\nObama", side = 3, line = -1.2, at = 0, cex = 0.8)
      mtext("Linear probability model (LPM)", side = 3, line = 0.2, cex = 0.75)
      axis(1, at=seq(1, 7, by=1), labels=seq(1, 7, by=1), cex.axis=0.85, padj=-1.)
      axis(2, at = seq(0.0, 1.0, by=0.25), labels = seq(0.0, 1.0, by=0.25),
            tick = T, las = 1, cex.axis = .85, mgp = c(2,.7,0))
      mtext("Conservative ideology", side = 1, line = 1.5, cex = 0.8)
      lines(LibConSpan, LPMFitLineDog0, type="l", lwd=2.0, col="darkblue")
      lines(LibConSpan, LPMFitLineDog1, type="l", lwd=2.0, col="cornflowerblue", lty = 2)
      text(6.1, 0.9, "non-dog owner", col="darkblue", cex=0.75)
      text(5.85, 0.85, "dog owner", col="cornflowerblue", cex=0.75)
      lines(c(4, 5.), c(0.90, 0.90), type="l", lwd=1.0, col="darkblue")
      lines(c(4, 5.), c(0.85, 0.85), type="l", lwd=1.0, col="cornflowerblue", lty = 2)

plot(as.numeric(dog$libcon), fitted(DogLPM), ylim=c(-0.05, 1.05), xlim=c(1, 7), type = "n", xlab="", x
      axis(1, at=seq(1, 7, by=1), labels=seq(1, 7, by=1), cex.axis=0.85, padj=-1.)
      axis(2, at = seq(0.0, 1.0, by=0.25), labels = seq(0.0, 1.0, by=0.25),
            tick = T, las = 1, cex.axis = .85, mgp = c(2,.7,0))
      mtext("Conservative ideology", side = 1, line = 1.5, cex = 0.8)
      mtext("Probit", side = 3, line = 0.2, cex = 0.75)
      lines(LibConSpan, ProbFitLineDog0, type="l", lwd=2.0, col="darkblue")
      lines(LibConSpan, ProbFitLineDog1, type="l", lwd=2.0, col="cornflowerblue", lty = 2)
      text(6.1, 0.9, "non-dog owner", col="darkblue", cex=0.75)
      text(5.85, 0.85, "dog owner", col="cornflowerblue", cex=0.75)
      lines(c(4, 5.), c(0.90, 0.90), type="l", lwd=1.0, col="darkblue")
      lines(c(4, 5.), c(0.85, 0.85), type="l", lwd=1.0, col="cornflowerblue", lty = 2)

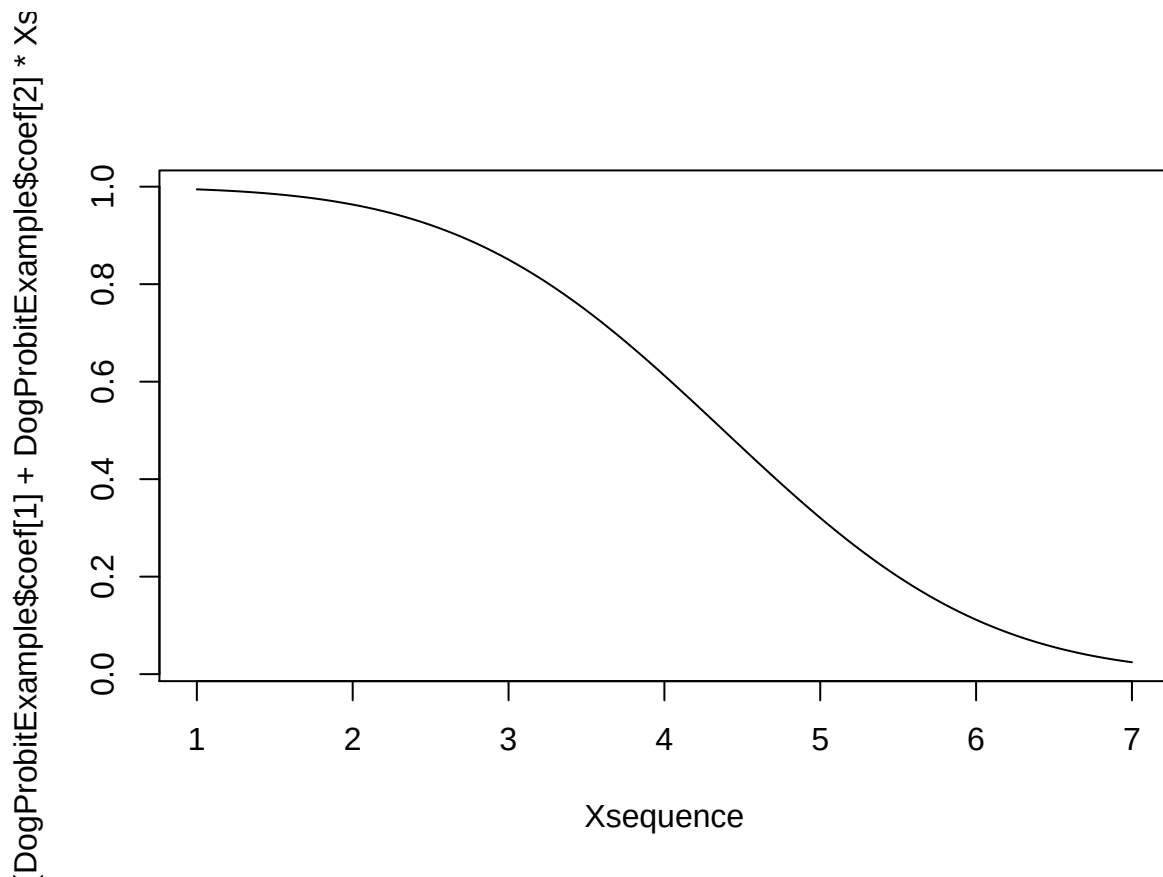
plot(as.numeric(dog$libcon), fitted(DogLPM), ylim=c(-0.05, 1.05), xlim=c(1, 7), type = "n", xlab="", x
      axis(1, at=seq(1, 7, by=1), labels=seq(1, 7, by=1), cex.axis=0.85, padj=-1.)
      axis(2, at = seq(0.0, 1.0, by=0.25), labels = seq(0.0, 1.0, by=0.25),
            tick = T, las = 1, cex.axis = .85, mgp = c(2,.7,0))
      mtext("Conservative ideology", side = 1, line = 1.5, cex = 0.8)
      mtext("Logit", side = 3, line = 0.2, cex = 0.75)
      lines(LibConSpan, LogFitLineDog0, type="l", lwd=2.0, col="darkblue")
      lines(LibConSpan, LogFitLineDog1, type="l", lwd=2.0, col="cornflowerblue", lty = 2)
      text(6.1, 0.9, "non-dog owner", col="darkblue", cex=0.75)
      text(5.85, 0.85, "dog owner", col="cornflowerblue", cex=0.75)
      lines(c(4, 5.), c(0.90, 0.90), type="l", lwd=1.0, col="darkblue")
      lines(c(4, 5.), c(0.85, 0.85), type="l", lwd=1.0, col="cornflowerblue", lty = 2)

```



Example for computing corner

```
DogProbitExample = glm(obamavote~ as.numeric(libcon), family = binomial(link = "probit"),
P1Example       = fitted(DogProbitExample)
Xsequence = seq(1, 7, 0.05)
plot(Xsequence, pnorm(DogProbitExample$coef[1] + DogProbitExample$coef[2]*Xsequence), type = "l")
```



Dogs and cats example

```
## Estimate unrestricted model
DogCatUR = glm(obamavote ~ dog + cat + as.numeric(libcon), family = binomial(link = "probit"),
summary(DogCatUR)

##
## Call:
## glm(formula = obamavote ~ dog + cat + as.numeric(libcon), family = binomial(link = "probit"),
## data = dog, na.action = na.exclude)
##
## Deviance Residuals:
## Min 1Q Median 3Q Max
## -3.327 -0.534 0.241 0.650 2.870
##
## Coefficients:
## Estimate Std. Error z value Pr(>|z|)
## (Intercept) 3.4101 0.0496 68.75 <2e-16 ***
## dog -0.2075 0.0250 -8.29 <2e-16 ***
## cat -0.0650 0.0261 -2.49 0.013 *
## as.numeric(libcon) -0.7537 0.0105 -71.68 <2e-16 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```



```
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 21517 on 15595 degrees of freedom
## Residual deviance: 13454 on 15592 degrees of freedom
## (3638 observations deleted due to missingness)
## AIC: 13462
##
## Number of Fisher Scoring iterations: 5
```

Estimate restricted model

```
dog$DogCat = dog$dog + dog$cat
DogCatR = glm(obamavote~ DogCat + as.numeric(libcon), family = binomial(link = "probit"), na.action = na.exclude)
summary(DogCatR)
```

```
##
## Call:
## glm(formula = obamavote ~ DogCat + as.numeric(libcon), family = binomial(link = "probit"),
## data = dog, na.action = na.exclude)
##
## Deviance Residuals:
## Min 1Q Median 3Q Max
## -3.326 -0.532 0.241 0.654 2.881
##
## Coefficients:
## Estimate Std. Error z value Pr(>|z|)
## (Intercept) 3.4104 0.0496 68.82 < 2e-16 ***
## DogCat -0.1392 0.0173 -8.04 9.1e-16 ***
## as.numeric(libcon) -0.7546 0.0105 -71.85 < 2e-16 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 21517 on 15595 degrees of freedom
## Residual deviance: 13468 on 15593 degrees of freedom
## (3638 observations deleted due to missingness)
## AIC: 13474
##
## Number of Fisher Scoring iterations: 5
```

Calculate LR test statistic

```
logLik(DogCatUR)[1]
```

```
## [1] -6727
```

```
logLik(DogCatR)[1]
```

```
## [1] -6734
```

```
LR = 2*(logLik(DogCatUR)[1] - logLik(DogCatR)[1]) # LR = 2*(-6726.7899 +6733.9411)
14.3024
```

```
## [1] 14.3
```

```
## Calculate p value and critical value for LR test statistic  
1-pchisq(LR, 1)      # p value
```

```
## [1] 0.000156
```

```
qchisq(0.95, 1)      # Critical value
```

```
## [1] 3.84
```

```
## Wald test  
library(car)  
linearHypothesis(DogCatUR,c("dog = cat"),test="F")
```

```
## Linear hypothesis test  
##  
## Hypothesis:  
## dog - cat = 0  
##  
## Model 1: restricted model  
## Model 2: obamavote ~ dog + cat + as.numeric(libcon)  
##  
##   Res.Df Df    F Pr(>F)  
## 1  15593  
## 2  15592  1 14.3 0.00015 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# library(lmtest)      #waldtest(DogCatUR, DogCatR)  
  
apsrtable(DogCatUR , DogCatR, model.names= c("UR", "R"),  
  coef.names=c("Constant", "Dog owner", "Cat owner", "Ideology", "Dog owner + Cat owner" ), digits=3)
```

```
## \begin{table}[!ht]  
## \caption{  
## \label{  
## \begin{tabular}{l D{.}{.}{3}D{.}{.}{3} }  
## \hline  
##   & \multicolumn{1}{c}{ UR } & \multicolumn{1}{c}{ R } & \\\hline  
##   %                & UR                & R                & \\  
## Constant          & 3.410 ^*          & 3.410 ^*          & \\  
##                   & (0.050)           & (0.050)           & \\  
## Dog owner         & -0.207 ^*         &                   & \\  
##                   & (0.025)           &                   & \\  
## Cat owner         & -0.065 ^*         &                   & \\  
##                   & (0.026)           &                   & \\  
## Ideology          & -0.754 ^*         & -0.755 ^*         & \\  
##                   & (0.011)           & (0.011)           & \\  
## Dog owner + Cat owner &                   & -0.139 ^*         & \\  
##                   &                   & (0.017)           & \\  

```

```
##   $N$                & 15596      & 15596      \\
##   AIC                & 13461.580 & 13473.882\\
##   BIC                & 13584.056 & 13565.739\\
##   $\log L$          & -6714.790 & -6724.941 \\ \hline
##   \multicolumn{3}{l}{\footnotesize{Standard errors in parentheses}}\\
##   \multicolumn{3}{l}{\footnotesize{$^*$ indicates significance at $p < 0.05$}}
##   \end{tabular}
##   \end{table}
```

```
# APSRTABLE gets likelihood wrong
```

More dog politics

```
# marginal effects
Marg      = DogProb$coef[3]* dnorm(DogProb$coef[1] + DogProb$coef[2]*dog$dog + DogProb$coef[3]*as
mean(Marg[is.na(dog$bamavote)==0], na.rm=T)
```

```
## [1] -0.183
```

```
Marg[1:10]
```

```
##   dog   dog   dog   dog   dog   dog   dog   dog   dog   dog
## -0.280    NA -0.296 -0.122 -0.159 -0.296 -0.159 -0.159 -0.296 -0.280
```

```
## Dummy variable, observed-value approach, discrete change
P0      = pnorm(DogProb$coef[1] + DogProb$coef[2]*0 + DogProb$coef[3]*as.numeric(dog$libcon))
P1      = pnorm(DogProb$coef[1] + DogProb$coef[2]*1 + DogProb$coef[3]*as.numeric(dog$libcon))
```

Case Study from Real Econometrics

```
library(foreign)
#ketchup      = read.dta("C:\\Users\\baileyma\\Dropbox\\Textbook_drop\\TextbookData\\Ketchup\\Ketchup_Sta
#save(ketchup, file = "Ch12_ChapterExample_Ketchup.RData")
load(file = "Ch12_ChapterExample_Ketchup.RData")

# Subset
ketchupExam = ketchup %>%
  filter(BoughtKetchup ==1) %>%
  select(StoreBrand, Pdiff, "display" = dp4, "featured" = ft4)
save(ketchupExam, file = "PracticeExam_Ketchup.RData")

# LPM for core model
KetchLPM    = lm(StoreBrand~      Pdiff + DisplayBrand + dp4 + FeatureBrand + ft4 +
                  hhold_income + hhold_members, na.action=na.exclude,
                  data=ketchup[ketchup$BoughtKetchup ==1,])

# Estimate heteroscedasticity-consistent standard errors for LPM
```

```
library(lmtest)
library(sandwich) ##
lmtest::coeftest(KetchLPM, sandwich::vcovHC(KetchLPM, type = "HC1"))
```

```
##
## t test of coefficients:
##
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.171253   0.007217   23.73 < 2e-16 ***
## Pdiff         0.134034   0.015539    8.63 < 2e-16 ***
## DisplayBrand -0.042162   0.004719   -8.93 < 2e-16 ***
## dp4           0.213519   0.021729    9.83 < 2e-16 ***
## FeatureBrand -0.083315   0.003843  -21.68 < 2e-16 ***
## ft4           0.303644   0.020635   14.71 < 2e-16 ***
## hhold_income -0.010183   0.000778  -13.09 < 2e-16 ***
## hhold_members 0.005249   0.001527    3.44 0.00059 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(KetchLPM)
```

```
##
## Call:
## lm(formula = StoreBrand ~ Pdiff + DisplayBrand + dp4 + FeatureBrand +
##      ft4 + hhold_income + hhold_members, data = ketchup[ketchup$BoughtKetchup ==
##      1, ], na.action = na.exclude)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7654 -0.1454 -0.0985 -0.0384  1.0836
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.171253   0.006861   24.96 < 2e-16 ***
## Pdiff         0.134034   0.012793   10.48 < 2e-16 ***
## DisplayBrand -0.042162   0.005644   -7.47 8.3e-14 ***
## dp4           0.213519   0.013939   15.32 < 2e-16 ***
## FeatureBrand -0.083315   0.004389  -18.98 < 2e-16 ***
## ft4           0.303644   0.013026   23.31 < 2e-16 ***
## hhold_income -0.010183   0.000774  -13.16 < 2e-16 ***
## hhold_members 0.005249   0.001525    3.44 0.00058 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.312 on 23428 degrees of freedom
## Multiple R-squared:  0.0829, Adjusted R-squared:  0.0827
## F-statistic: 303 on 7 and 23428 DF, p-value: <2e-16
```

```
# Missing data can mess up fitted values unless use na.exclude option
LPMFit = fitted(KetchLPM)
```

```
# Min and max of fitted values
c(min(LPMFit), max(LPMFit))
```

```
## [1] -0.133 0.765
```

```
# Probit
KetchProbit = glm(StoreBrand~ Pdiff + DisplayBrand + dp4 + FeatureBrand + ft4 +
                  hhold_income + hhold_members,
                  family = binomial(link = "probit"), na.action=na.exclude,
                  data=ketchup[ketchup$BoughtKetchup ==1,])
summary(KetchProbit)
```

```
##
## Call:
## glm(formula = StoreBrand ~ Pdiff + DisplayBrand + dp4 + FeatureBrand +
##      ft4 + hhold_income + hhold_members, family = binomial(link = "probit"),
##      data = ketchup[ketchup$BoughtKetchup == 1, ], na.action = na.exclude)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.995  -0.553  -0.429  -0.290   3.128
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.92868    0.03695  -25.13  < 2e-16 ***
## Pdiff         0.68453    0.07358   9.30  < 2e-16 ***
## DisplayBrand -0.26572    0.03503  -7.59  3.3e-14 ***
## dp4          0.68340    0.06001  11.39  < 2e-16 ***
## FeatureBrand -0.53662    0.02749 -19.52  < 2e-16 ***
## ft4          0.94825    0.05485  17.29  < 2e-16 ***
## hhold_income -0.05537    0.00440 -12.57  < 2e-16 ***
## hhold_members 0.03008    0.00831   3.62   3e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 17221  on 23435  degrees of freedom
## Residual deviance: 15600  on 23428  degrees of freedom
## AIC: 15616
##
## Number of Fisher Scoring iterations: 5
```

```
#log likelihood in the probit model
logLik(KetchProbit)
```

```
## 'log Lik.' -7800 (df=8)
```

```
# Missing data can mess up fitted values unless use na.exclude option
ProbFit = fitted(KetchProbit)
c(min(ProbFit), max(ProbFit))
```

```
## [1] 0.00369 0.86338
```

```
KetchLogit = glm(StoreBrand~ Pdiff + DisplayBrand + dp4 + FeatureBrand + ft4 +
                hhold_income + hhold_members,
                family = binomial(link = "logit"), na.action=na.exclude,
                data=ketchup[ketchup$BoughtKetchup ==1,])
summary(KetchLogit)
```

```
##
## Call:
## glm(formula = StoreBrand ~ Pdiff + DisplayBrand + dp4 + FeatureBrand +
##      ft4 + hhold_income + hhold_members, family = binomial(link = "logit"),
##      data = ketchup[ketchup$BoughtKetchup == 1, ], na.action = na.exclude)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.086  -0.550  -0.424  -0.295   3.004
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.53691    0.06933  -22.17  < 2e-16 ***
## Pdiff         1.41509    0.14070   10.06  < 2e-16 ***
## DisplayBrand  -0.54268    0.07063   -7.68  1.5e-14 ***
## dp4           1.16360    0.10125   11.49  < 2e-16 ***
## FeatureBrand  -1.03859    0.05641  -18.41  < 2e-16 ***
## ft4           1.60578    0.09148   17.55  < 2e-16 ***
## hhold_income  -0.10927    0.00844  -12.95  < 2e-16 ***
## hhold_members  0.05996    0.01562    3.84  0.00012 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 17221  on 23435  degrees of freedom
## Residual deviance: 15600  on 23428  degrees of freedom
## AIC: 15616
##
## Number of Fisher Scoring iterations: 5
```

```
# Missing data can mess up fitted values unless use na.exclude option
LogitFit = fitted(KetchLogit )
c(min(LogitFit), max(LogitFit))
```

```
## [1] 0.00655 0.88655
```

```
logLik(KetchLogit ) # Note apsrtable function provides wrong likelihood (this function produces same va
```

```
## 'log Lik.' -7800 (df=8)
```

```
## Continuous variable, observed-value approach, discrete change
PdiffPlus = ketchup$Pdiff+ 1 # 0.16 is one std dev
P1b = pnorm(KetchProbit$coef[1] + KetchProbit$coef[2]*ketchup$Pdiff + KetchProbit$coef[3]*ketchup$D
          KetchProbit$coef[8]*ketchup$hhold_members )
```

```
P2      = pnorm(KetchProbit$coef[1] + KetchProbit$coef[2]*PdiffPlus+ KetchProbit$coef[3]*ketchup$Display
              KetchProbit$coef[8]*ketchup$hhold_members )
mean(P2-P1b, na.rm=T)
```

```
## [1] 0.191
```

```
# marginal effects of Pdiff
```

```
Marg      = KetchProbit$coef[2]* dnorm(KetchProbit$coef[1] + KetchProbit$coef[2]*ketchup$Pdiff + KetchProbit$coef[3]*ketchup$Display
              KetchProbit$coef[8]*ketchup$hhold_members)
mean(Marg, na.rm=T)
```

```
## [1] 0.142
```

```
Marg[1:10]
```

```
## [1] 0.0903 0.1019 0.0907 0.0371 0.0469 0.1015 0.1112 0.1017 0.1003 0.0897
```

```
P2      = pnorm(KetchProbit$coef[1] + KetchProbit$coef[2]*ketchup$Pdiff+ KetchProbit$coef[3]*ketchup$Display
              KetchProbit$coef[7]*(ketchup$hhold_income + 1 ) + KetchProbit$coef[8]*ketchup$hhold_members )
mean(P2-P1b, na.rm=T)
```

```
## [1] -0.0111
```

```
P2      = pnorm(KetchProbit$coef[1] + KetchProbit$coef[2]*ketchup$Pdiff+ KetchProbit$coef[3]*ketchup$Display
              KetchProbit$coef[7]*ketchup$hhold_income + KetchProbit$coef[8]*(ketchup$hhold_members + 1))
mean(P2-P1b, na.rm=T)
```

```
## [1] 0.00634
```

```
## Dummy variable, observed-value approach, discrete change
```

```
POdumProbit = pnorm(KetchProbit$coef[1] + KetchProbit$coef[2]*ketchup$Pdiff + KetchProbit$coef[3]*ketchup$Display
                  KetchProbit$coef[8]*ketchup$hhold_members)
P1dumProbit = pnorm(KetchProbit$coef[1] + KetchProbit$coef[2]*ketchup$Pdiff + KetchProbit$coef[3]*ketchup$Display
                  KetchProbit$coef[8]*ketchup$hhold_members)
mean(P1dumProbit-POdumProbit, na.rm=T)
```

```
## [1] 0.19
```

```
POdumProbit = pnorm(KetchProbit$coef[1] + KetchProbit$coef[2]*ketchup$Pdiff + KetchProbit$coef[3]*ketchup$Display
                  KetchProbit$coef[8]*ketchup$hhold_members)
P1dumProbit = pnorm(KetchProbit$coef[1] + KetchProbit$coef[2]*ketchup$Pdiff + KetchProbit$coef[3]*ketchup$Display
                  KetchProbit$coef[8]*ketchup$hhold_members)
mean(P1dumProbit-POdumProbit, na.rm=T)
```

```
## [1] -0.0487
```

```

POdumProbit      = pnorm(KetchProbit$coef[1] + KetchProbit$coef[2]*ketchup$Pdiff + KetchProbit$coef[3]*k
                  KetchProbit$coef[8]*ketchup$hhold_members)
P1dumProbit      = pnorm(KetchProbit$coef[1] + KetchProbit$coef[2]*ketchup$Pdiff + KetchProbit$coef[3]*k
                  KetchProbit$coef[8]*ketchup$hhold_members)
mean(P1dumProbit-POdumProbit, na.rm=T)

```

```
## [1] -0.089
```

```

POdumProbit      = pnorm(KetchProbit$coef[1] + KetchProbit$coef[2]*ketchup$Pdiff + KetchProbit$coef[3]*k
                  KetchProbit$coef[8]*ketchup$hhold_members)
P1dumProbit      = pnorm(KetchProbit$coef[1] + KetchProbit$coef[2]*ketchup$Pdiff + KetchProbit$coef[3]*k
                  KetchProbit$coef[8]*ketchup$hhold_members)
mean(P1dumProbit-POdumProbit, na.rm=T)

```

```
## [1] 0.286
```

```

# Logit
P1Logit          = fitted(KetchLogit)
P1bLogit         = exp(KetchLogit$coef[1] + KetchLogit$coef[2]*ketchup$Pdiff + KetchLogit$coef[3]*ketchup$Dis
                  KetchLogit$coef[8]*ketchup$hhold_members) / (1+ exp(KetchLogit$coef[1] + KetchLogit$coef[2]
                  KetchLogit$coef[8]*ketchup$hhold_members))
P2Logit          = exp(KetchLogit$coef[1] + KetchLogit$coef[2]*PdiffPlus + KetchLogit$coef[3]*ketchup$Display
                  KetchLogit$coef[8]*ketchup$hhold_members) / (1+ exp(KetchLogit$coef[1] + KetchLogit$coef[2]
                  KetchLogit$coef[8]*ketchup$hhold_members))
mean(P2Logit-P1bLogit, na.rm=T)      #mean(P1b-P1, na.rm=T) # mean(P2Logit, na.rm=T)

```

```
## [1] 0.237
```

```
c(min(P1, na.rm=T), max(P1, na.rm=T))
```

```
## [1] 0.0181 0.9923
```

```

P2Logit          = exp(KetchLogit$coef[1] + KetchLogit$coef[2]*ketchup$Pdiff+ KetchLogit$coef[3]*ketchup$Dis
                  KetchLogit$coef[8]*ketchup$hhold_members) / (1+ exp(KetchLogit$coef[1] + KetchLogit$coef[2]
                  KetchLogit$coef[8]*ketchup$hhold_members))
mean(P2Logit-P1bLogit, na.rm=T)

```

```
## [1] -0.0118
```

```

P2Logit          = exp(KetchLogit$coef[1] + KetchLogit$coef[2]*ketchup$Pdiff+ KetchLogit$coef[3]*ketchup$Dis
                  KetchLogit$coef[8]*(ketchup$hhold_members+1)) / (1+ exp(KetchLogit$coef[1] + KetchLogit$coef[2]
                  KetchLogit$coef[8]*(ketchup$hhold_members+1)))
mean(P2Logit-P1bLogit, na.rm=T)

```

```
## [1] 0.00685
```



```
## Dummy variable, observed-value approach, discrete change
```

```
P0dumLogit      = exp(KetchLogit$coef[1] + KetchLogit$coef[2]*ketchup$Pdiff + KetchLogit$coef[3]*ketchup$
  KetchLogit$coef[8]*ketchup$hhold_members) / (1+ exp(KetchLogit$coef[1] + KetchLogit$coef[2]*
  KetchLogit$coef[8]*ketchup$hhold_members))
P1dumLogit      = exp(KetchLogit$coef[1] + KetchLogit$coef[2]*ketchup$Pdiff + KetchLogit$coef[3]*ketchup$
  KetchLogit$coef[8]*ketchup$hhold_members) / (1+ exp(KetchLogit$coef[1] + KetchLogit$coef[2]*
  KetchLogit$coef[8]*ketchup$hhold_members))
mean(P1dumLogit-P0dumLogit, na.rm=T)
```

```
## [1] 0.184
```

```
P0dumLogit      = exp(KetchLogit$coef[1] + KetchLogit$coef[2]*ketchup$Pdiff + KetchLogit$coef[3]*0+ Ket
  KetchLogit$coef[8]*ketchup$hhold_members) / (1+ exp(KetchLogit$coef[1] + KetchLogit$coef[2]*
  KetchLogit$coef[8]*ketchup$hhold_members))
P1dumLogit      = exp(KetchLogit$coef[1] + KetchLogit$coef[2]*ketchup$Pdiff + KetchLogit$coef[3]*1 + Ke
  KetchLogit$coef[8]*ketchup$hhold_members) / (1+ exp(KetchLogit$coef[1] + KetchLogit$coef[2]*
  KetchLogit$coef[8]*ketchup$hhold_members))
mean(P1dumLogit-P0dumLogit, na.rm=T)
```

```
## [1] -0.0519
```

```
P0dumLogit      = exp(KetchLogit$coef[1] + KetchLogit$coef[2]*ketchup$Pdiff + KetchLogit$coef[3]*ketchup$
  KetchLogit$coef[8]*ketchup$hhold_members) / (1+ exp(KetchLogit$coef[1] + KetchLogit$coef[2]*
  KetchLogit$coef[8]*ketchup$hhold_members))
P1dumLogit      = exp(KetchLogit$coef[1] + KetchLogit$coef[2]*ketchup$Pdiff + KetchLogit$coef[3]*ketchup$
  KetchLogit$coef[8]*ketchup$hhold_members) / (1+ exp(KetchLogit$coef[1] + KetchLogit$coef[2]*
  KetchLogit$coef[8]*ketchup$hhold_members))
mean(P1dumLogit-P0dumLogit, na.rm=T)
```

```
## [1] -0.0887
```

```
P0dumLogit      = exp(KetchLogit$coef[1] + KetchLogit$coef[2]*ketchup$Pdiff + KetchLogit$coef[3]*ketchup$
  KetchLogit$coef[8]*ketchup$hhold_members) / (1+ exp(KetchLogit$coef[1] + KetchLogit$coef[2]*
  KetchLogit$coef[8]*ketchup$hhold_members))
P1dumLogit      = exp(KetchLogit$coef[1] + KetchLogit$coef[2]*ketchup$Pdiff + KetchLogit$coef[3]*ketchup$
  KetchLogit$coef[8]*ketchup$hhold_members) / (1+ exp(KetchLogit$coef[1] + KetchLogit$coef[2]*
  KetchLogit$coef[8]*ketchup$hhold_members))
mean(P1dumLogit-P0dumLogit, na.rm=T)
```

```
## [1] 0.279
```

```
library(apsrtable)
apsrtable(KetchLPM, KetchProbit, KetchLogit, model.names= c("LPM", "Probit", "Logit"),
  coef.names=c("Constant", "Price difference", "Brand name displayed", "Store brand displayed", "Brand name displayed"))
```

```
## \begin{table}[!ht]
## \caption{}
## \label{}
## \begin{tabular}{l D{.}{.}{3}D{.}{.}{3}D{.}{.}{3} }
## \hline
```

```
## & \multicolumn{1}{c}{ LPM } & \multicolumn{1}{c}{ Probit } & \multicolumn{1}{c}{ Logit }
## % & LPM & Probit & Logit \\
## Constant & 0.171 ^* & -0.929 ^* & -1.537 ^*\\
## & (0.007) & (0.037) & (0.069) \\
## Price difference & 0.134 ^* & 0.685 ^* & 1.415 ^* \\
## & (0.013) & (0.074) & (0.141) \\
## Brand name displayed & -0.042 ^* & -0.266 ^* & -0.543 ^*\\
## & (0.006) & (0.035) & (0.071) \\
## Store brand displayed & 0.214 ^* & 0.683 ^* & 1.164 ^* \\
## & (0.014) & (0.060) & (0.101) \\
## Brand name featured & -0.083 ^* & -0.537 ^* & -1.039 ^*\\
## & (0.004) & (0.027) & (0.056) \\
## Store brand featured & 0.304 ^* & 0.948 ^* & 1.606 ^* \\
## & (0.013) & (0.055) & (0.091) \\
## Household income & -0.010 ^* & -0.055 ^* & -0.109 ^*\\
## & (0.001) & (0.004) & (0.008) \\
## Household size & 0.005 ^* & 0.030 ^* & 0.060 ^* \\
## & (0.002) & (0.008) & (0.016) \\
## $N$ & 23436 & 23436 & 23436 \\
## $R^2$ & 0.083 & & \\
## adj. $R^2$ & 0.083 & & \\
## Resid. sd & 0.312 & & \\
## AIC & & 15616.404 & 15615.610\\
## BIC & & 15874.389 & 15873.595\\
## $log L$ & & -7776.202 & -7775.805 \\ \hline
## \multicolumn{4}{l}{\footnotesize{Standard errors in parentheses}}\\
## \multicolumn{4}{l}{\footnotesize{$^*$ indicates significance at $p< 0.05$}}
## \end{tabular}
## \end{table}
```

Other analyses

```
# Estimate household income as a factor (instead of psuedo-continuous as above)
KetchLPMfactor = lm(StoreBrand~ Pdifff + DisplayBrand + dp4 + FeatureBrand + ft4 +
                    factor(hhold_income) + hhold_members, na.action=na.exclude,
                    data=ketchup[ketchup$BoughtKetchup ==1,])
summary(KetchLPMfactor)
```

```
##
## Call:
## lm(formula = StoreBrand ~ Pdifff + DisplayBrand + dp4 + FeatureBrand +
##     ft4 + factor(hhold_income) + hhold_members, data = ketchup[ketchup$BoughtKetchup ==
##     1, ], na.action = na.exclude)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7813 -0.1453 -0.0973 -0.0348  1.0738
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.14823    0.01434   10.34 < 2e-16 ***
## Pdifff         0.13302    0.01279   10.40 < 2e-16 ***
## DisplayBrand  -0.04224    0.00564   -7.49 7.2e-14 ***
## dp4           0.21179    0.01393   15.20 < 2e-16 ***
## FeatureBrand  -0.08267    0.00439  -18.84 < 2e-16 ***
```

```

## ft4                0.30492    0.01302    23.42 < 2e-16 ***
## factor(hhold_income)2  0.00878    0.01638     0.54 0.59201
## factor(hhold_income)3 -0.00586    0.01532    -0.38 0.70238
## factor(hhold_income)4 -0.00462    0.01496    -0.31 0.75755
## factor(hhold_income)5 -0.03333    0.01455    -2.29 0.02203 *
## factor(hhold_income)6 -0.04427    0.01469    -3.01 0.00259 **
## factor(hhold_income)7 -0.07885    0.01488    -5.30 1.2e-07 ***
## factor(hhold_income)8 -0.04779    0.01534    -3.12 0.00184 **
## factor(hhold_income)9 -0.08426    0.01575    -5.35 8.9e-08 ***
## factor(hhold_income)10 -0.09056    0.01650    -5.49 4.1e-08 ***
## factor(hhold_income)11 -0.06220    0.01676    -3.71 0.00021 ***
## factor(hhold_income)12 -0.11625    0.01953    -5.95 2.7e-09 ***
## factor(hhold_income)13 -0.11962    0.02779    -4.30 1.7e-05 ***
## factor(hhold_income)14 -0.05509    0.02534    -2.17 0.02974 *
## hhold_members         0.00626    0.00155     4.03 5.7e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.311 on 23416 degrees of freedom
## Multiple R-squared:  0.0853, Adjusted R-squared:  0.0846
## F-statistic: 115 on 19 and 23416 DF, p-value: <2e-16

```