Logistic Regression

load all data from csv in working directory

```
df <- read_csv("cleandata.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   .default = col_integer(),
##   targdol = col_double(),
##   datead6 = col_date(format = ""),
##   datelp6 = col_date(format = ""),
##   recentseason = col_character(),
##   consistencycategory = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

to do: impute values for lpuryr

```
table(df$lpuryear, df$recency_year, useNA = "ifany")
```

```
##
##         1980  2002  2003  2004  2005  2006  2007  2008  2009  2010  2011
##   2003     0     0  1958    46     0     0     1     0     2     1    17
##   2004     0     0     0  3105     4     0     0     0     0     6    13
##   2005     0     0     0     0  5438     6     0     0     0    12    22
##   2006     0     0     0     0     0  6728    11     0     1    11    46
##   2007     0     0     0     0     0     0  9156     8   205    29    81
##   2008     0     0     0     0     0     0     0  3443  8349   281   135
##   2009     0     0     0     0     0     0     0     0  4208 11004   215
##   2010     0     0     0     0     0     0     0     0     0  4405 10507
##   2011     0     0     0     0     0     0     0     0     0     0  7256
##   2012     0     0     0     0     0     0     0     0     0     0     0
##   <NA>    17   687    12     6     0     0     0     0     0     0     3
##
##         2012
##   2003  2322
##   2004    24
##   2005    41
##   2006    36
##   2007    66
##   2008   119
##   2009   178
##   2010    19
##   2011 12477
##   2012  8812
##   <NA>     3
```

```
df$lpuryear[is.na(df$lpuryear) == TRUE] <- df$recency_year[is.na(df$lpuryear) == TRUE]
```

```
table(df$consistencycategory)
```

```
##
##     1    10    11     2     3     4     5     6     7     8     9   Ref
##  1351  1842  1282  1411  1030   637   825  2670  2774  3194  1683 82833
```

```
df$consistencycategory <- relevel(factor(df$consistencycategory), 'Ref')
```

create new column to predict called buy

```
head(df)
```

```
## # A tibble: 6 x 22
##      X1 targdol    datead6    datelp6 lpuryear slstyr slslyr sls2ago
##   <int>   <dbl>     <date>     <date>    <int>  <int>  <int>   <int>
## 1     1       0 1980-01-01 1980-01-01     1980      0      0       0
## 2     2       0 1980-01-01 1980-01-01     1980      0      0       0
## 3     3       0 1980-01-01 1980-01-01     1980      0      0       0
## 4     4       0 1980-01-01 1980-01-01     1980      0      0       0
## 5     5       0 1980-01-01 1980-01-01     1980      0      0       0
## 6     6       0 1980-01-01 1980-01-01     1980      0      0       0
## # ... with 14 more variables: sls3ago <int>, slshist <int>, ordtyr <int>,
## #   ordlyr <int>, ord2ago <int>, ord3ago <int>, ordhist <int>,
## #   falord <int>, sprord <int>, train <int>, recentseason <chr>,
## #   recency_year <int>, years_since_purchase <int>,
## #   consistencycategory <fctr>
```

```
df$buy <- 0
df$buy[df$targdol > 0] <- 1
table(df$buy, useNA = "ifany")
```

```
##
##     0     1
## 91961  9571
```

```
#9.4% buy rate
sum(df$buy)/nrow(df)
```

```
## [1] 0.09426585
```

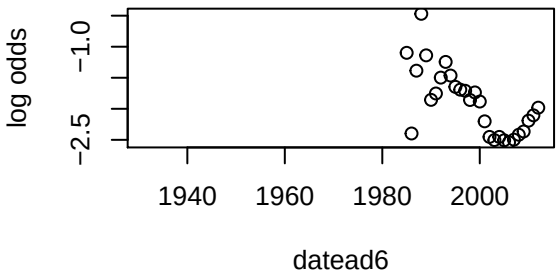
```
df[df$buy == 1,]
```

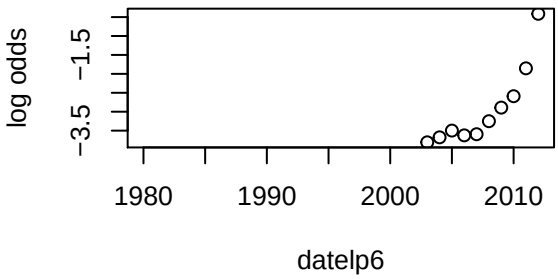
```
## # A tibble: 9,571 x 23
##       X1   targdol    datead6    datelp6 lpuryear slstyr slslyr sls2ago
##    <int>     <dbl>     <date>     <date>    <int>  <int>  <int>   <int>
##  1    19  12.95000 2011-01-15 2012-01-13     2012      4      0       0
##  2    20  45.00000 2011-05-01 2012-03-11     2012     42      0       0
##  3    50 109.94995 2009-11-27 2011-11-15     2011     34     35       0
##  4    63  25.45000 2011-06-10 2011-11-15     2011     31      0       0
##  5    95  13.89999 2011-10-11 2011-11-15     2011     27      0       0
##  6   108  66.89996 2012-03-19 2012-03-19     2012     20      0       0
##  7   112  93.75000 2009-10-17 2011-11-15     2011    102    121       0
##  8   114  45.89999 2007-07-31 2012-03-25     2012     53      0      30
```

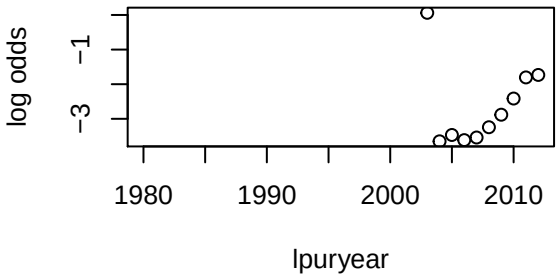
```
##  9   117 123.34998 1999-11-21 2009-11-15     2009      0      0     203
## 10   127 113.39996 2011-05-16 2012-02-25     2012    114      0       0
## # ... with 9,561 more rows, and 15 more variables: sls3ago <int>,
## #   slshist <int>, ordtyr <int>, ordlyr <int>, ord2ago <int>,
## #   ord3ago <int>, ordhist <int>, falord <int>, sprord <int>, train <int>,
## #   recentseason <chr>, recency_year <int>, years_since_purchase <int>,
## #   consistencycategory <fctr>, buy <dbl>
```

frequency tables/logistic transform plot (see pg 131 of book)

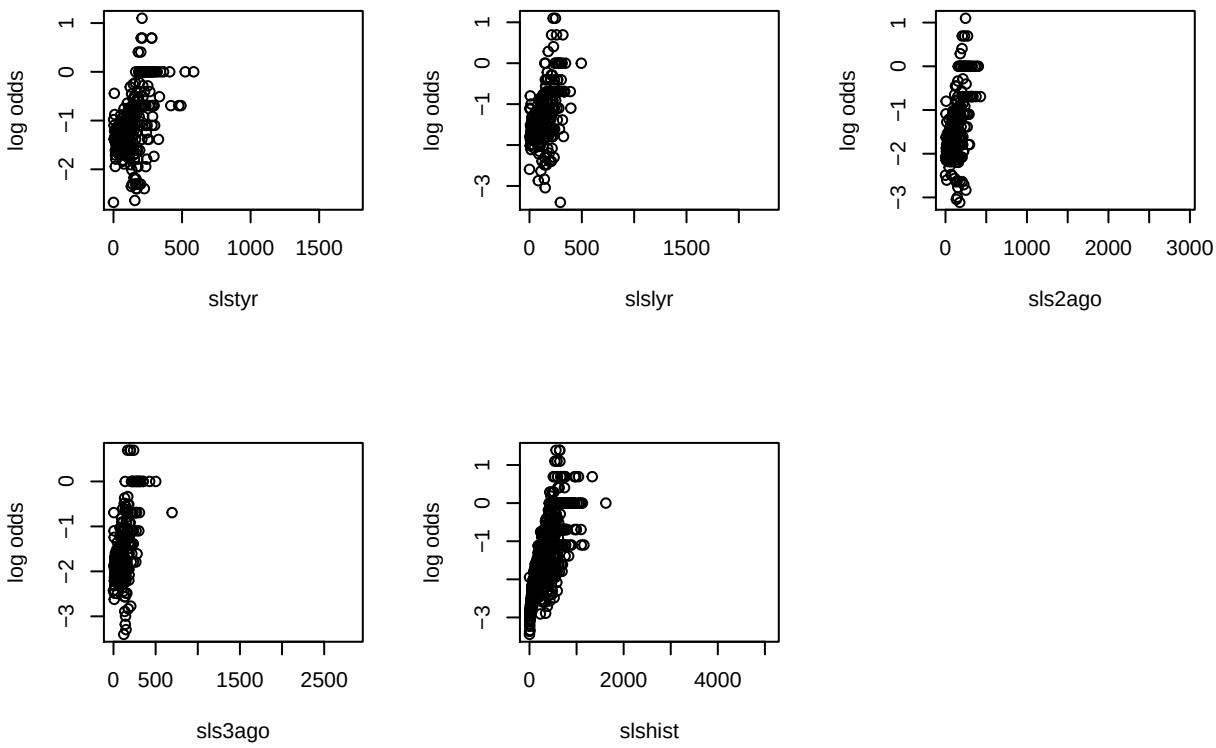
```
frequency_table <- function(predictor, name) {
  tab <- as.data.frame.matrix(table(predictor, df$buy))
  tab$buy_prop <- tab[,2]/rowSums(tab)
  tab$log_odds <- log(tab$buy_prop/(1 - tab$buy_prop))
  return (plot(x = as.numeric(rownames(tab)),
               y = tab$log_odds,
               xlab = name,
               ylab = 'log odds'))
}

par(mfrow = c(2, 2))
frequency_table(year(df$datead6), 'datead6')
frequency_table(year(df$datelp6), 'datelp6')
frequency_table(df$lpuryear, 'lpuryear')

par(mfrow = c(2, 3))
```



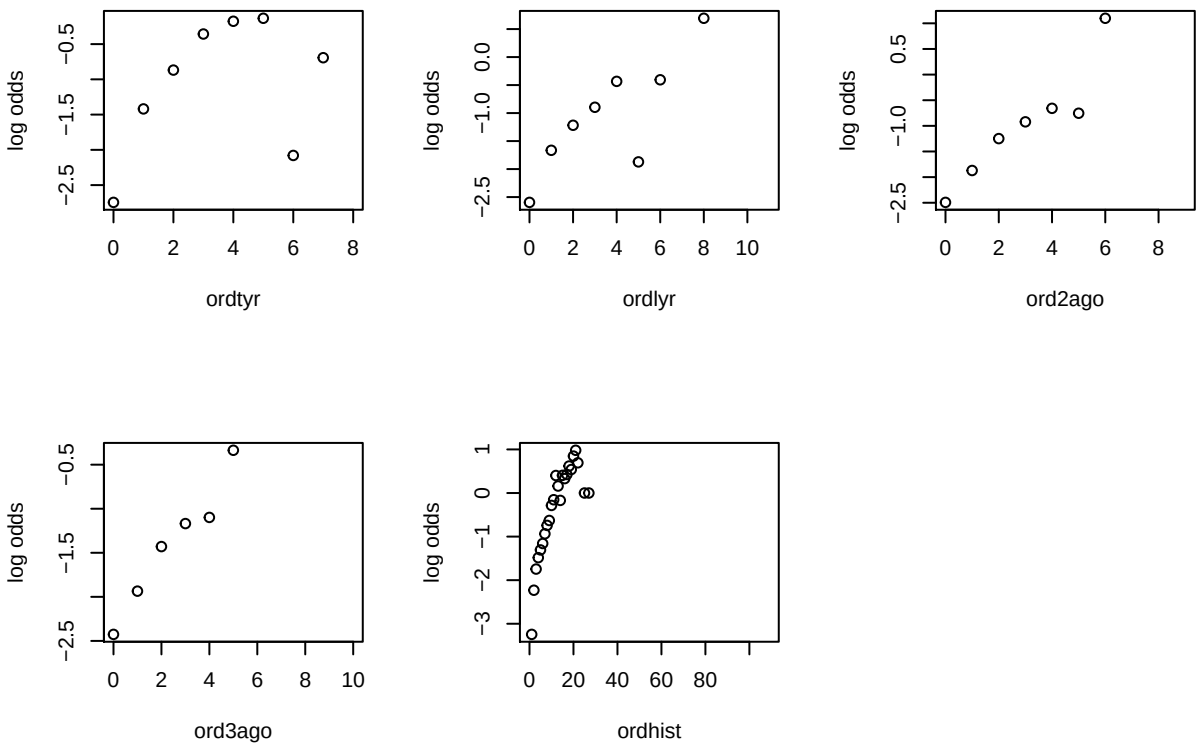




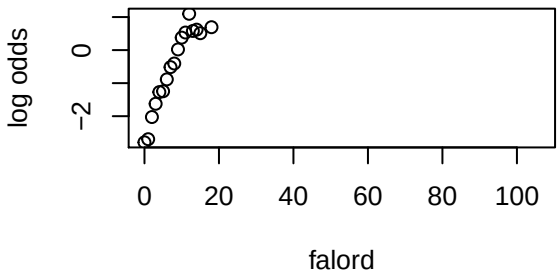

```r
frequency_table(df$slstyr, 'slstyr')
frequency_table(df$slslyr, 'slslyr')
frequency_table(df$sls2ago, 'sls2ago')
frequency_table(df$sls3ago, 'sls3ago')
frequency_table(df$slshist, 'slshist')

par(mfrow = c(2, 3))
```

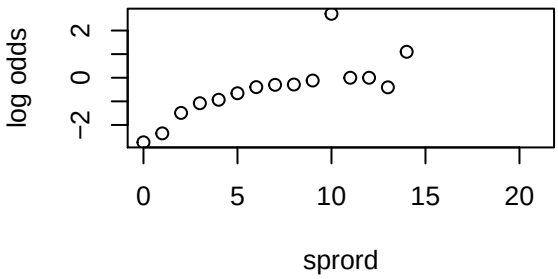


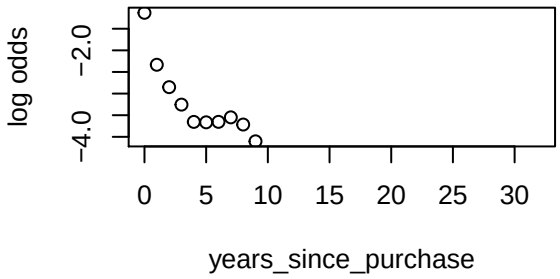

```
frequency_table(df$ordtyr, 'ordtyr')
frequency_table(df$ordlyr, 'ordlyr')
frequency_table(df$ord2ago, 'ord2ago')
frequency_table(df$ord3ago, 'ord3ago')
frequency_table(df$ordhist, 'ordhist')

par(mfrow = c(2, 2))
```




```
frequency_table(df$falord, 'falord')
frequency_table(df$sprord, 'sprord')
frequency_table(df$years_since_purchase, 'years_since_purchase')
```







split into test and train + other data cleaning

```
df_train <- df[df$train == 1, ]
df_test <- df[df$train == 0, ]

cat('\nActual targdol of test buyers', sum(df_test$targdol))
```

```
##
## Actual targdol of test buyers 226456.3
```

```
# #get names
# z <- NULL
# for (name in names(df_test)) {
#   z <- paste(z, name, "+")
# }
# z
```

nominal logistic regression model?

fit with glm

```
fit_glm <- glm(buy ~ datead6 + datelp6 + lpuryear + slstyr + slslyr + sls2ago + sls3ago + slshist + ord
           family = binomial(link = 'logit'),
```

```
          data = df_train)
summary(fit_glm)
```

```
##
## Call:
## glm(formula = buy ~ datead6 + datelp6 + lpuryear + slstyr + slslyr +
##     sls2ago + sls3ago + slshist + ordtyr + ordlyr + ord2ago +
##     ord3ago + falord + sprord + recentseason + years_since_purchase +
##     consistencycategory, family = binomial(link = "logit"), data = df_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.8853  -0.3944  -0.2937  -0.2219   3.3875
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)            9.042e+02  2.438e+01  37.084  < 2e-16 ***
## datead6               -4.403e-05  1.326e-05  -3.321 0.000896 ***
## datelp6                2.428e-03  1.924e-04  12.621  < 2e-16 ***
## lpuryear              -4.691e-01  1.207e-02 -38.880  < 2e-16 ***
## slstyr                 9.524e-04  5.474e-04   1.740 0.081857 .
## slslyr                 4.246e-04  5.644e-04   0.752 0.451870
## sls2ago               -6.180e-05  7.194e-04  -0.086 0.931545
## sls3ago                6.200e-04  5.815e-04   1.066 0.286359
## slshist               -5.751e-04  2.564e-04  -2.243 0.024881 *
## ordtyr                 2.109e-01  4.990e-02   4.226 2.37e-05 ***
## ordlyr                 2.952e-02  4.636e-02   0.637 0.524282
## ord2ago                4.044e-04  5.065e-02   0.008 0.993630
## ord3ago               -1.619e-02  5.185e-02  -0.312 0.754853
## falord                 2.170e-01  1.819e-02  11.931  < 2e-16 ***
## sprord                -2.866e-02  2.209e-02  -1.298 0.194423
## recentseasonSpring     5.003e-01  4.816e-02  10.388  < 2e-16 ***
## years_since_purchase   1.415e-01  6.293e-02   2.249 0.024518 *
## consistencycategory1   1.261e+00  1.385e-01   9.104  < 2e-16 ***
## consistencycategory10  4.139e-01  1.175e-01   3.521 0.000429 ***
## consistencycategory11 -8.394e-02  1.335e-01  -0.629 0.529507
## consistencycategory2   6.394e-01  1.333e-01   4.797 1.61e-06 ***
## consistencycategory3   8.079e-01  1.309e-01   6.170 6.84e-10 ***
## consistencycategory4   5.297e-01  1.582e-01   3.349 0.000811 ***
## consistencycategory5   6.821e-01  1.452e-01   4.697 2.64e-06 ***
## consistencycategory6   7.124e-01  8.785e-02   8.109 5.11e-16 ***
## consistencycategory7   4.303e-01  1.039e-01   4.142 3.45e-05 ***
## consistencycategory8   3.132e-01  1.109e-01   2.825 0.004728 **
## consistencycategory9   1.718e-01  1.150e-01   1.494 0.135165
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 31907  on 50417  degrees of freedom
## Residual deviance: 23444  on 50390  degrees of freedom
## AIC: 23500
##
## Number of Fisher Scoring iterations: 6
```

compare predicted to test for first fit

```
test_predict <- predict(fit_glm, newdata = df_test, type = 'response')
table(df_test$buy, round(test_predict, 1), dnn = c('true', 'predicted'))
```

```
##     predicted
## true     0   0.1   0.2   0.3   0.4   0.5   0.6   0.7   0.8   0.9     1
##    0 25243 17539  2192   721   295   161    84    54    30    62     7
##    1   647  1633   534   288   145    99   103    82    70   816   309
```

```
test_predict[test_predict < .1] <- 0
test_predict[test_predict >= .1] <- 1

tab <- table(df_test$buy, test_predict, dnn = c('true', 'predicted'))
tab
```

```
##     predicted
## true     0     1
##    0 39194  7194
##    1  1759  2967
```

```
cat('CCR:', sum(diag(tab))/sum(tab))
```

```
## CCR: 0.8248425
```

```
cat('\nTrue targdol of predicted buyers', sum(df_test$targdol[test_predict == 1]))
```

```
##
## True targdol of predicted buyers 146854.3
```

fit with glm, fewer predictors

```
fit2_glm <- glm(buy ~ datead6 + datelp6 + lpuryear + slshist + ordtyr + falord + sprord + recentseason +
           family = binomial(link = 'logit'),
           data = df_train)
summary(fit2_glm)
```

```
##
## Call:
## glm(formula = buy ~ datead6 + datelp6 + lpuryear + slshist +
##     ordtyr + falord + sprord + recentseason + years_since_purchase +
##     consistencycategory, family = binomial(link = "logit"), data = df_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.7243  -0.3929  -0.2957  -0.2219   3.3813
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)            9.028e+02  2.420e+01  37.309  < 2e-16 ***
## datead6               -4.212e-05  1.290e-05  -3.265  0.00109 **
## datelp6                2.432e-03  1.875e-04  12.972  < 2e-16 ***
## lpuryear              -4.685e-01  1.198e-02 -39.115  < 2e-16 ***
## slshist               -2.846e-04  1.424e-04  -1.998  0.04570 *
## ordtyr                 2.331e-01  3.960e-02   5.887 3.94e-09 ***
```

```
## falord                 2.072e-01  1.505e-02  13.769  < 2e-16 ***
## sprord                -3.475e-02  1.926e-02  -1.805  0.07115 .
## recentseasonSpring     4.983e-01  4.802e-02  10.377  < 2e-16 ***
## years_since_purchase   1.390e-01  6.142e-02   2.262  0.02367 *
## consistencycategory1   1.306e+00  1.120e-01  11.660  < 2e-16 ***
## consistencycategory10  4.536e-01  1.063e-01   4.268 1.97e-05 ***
## consistencycategory11 -8.506e-02  1.248e-01  -0.681  0.49564
## consistencycategory2   6.733e-01  1.146e-01   5.874 4.26e-09 ***
## consistencycategory3   8.491e-01  1.125e-01   7.550 4.35e-14 ***
## consistencycategory4   5.841e-01  1.442e-01   4.050 5.12e-05 ***
## consistencycategory5   6.717e-01  1.291e-01   5.204 1.95e-07 ***
## consistencycategory6   7.579e-01  7.873e-02   9.627  < 2e-16 ***
## consistencycategory7   4.565e-01  9.218e-02   4.952 7.34e-07 ***
## consistencycategory8   3.003e-01  9.595e-02   3.130  0.00175 **
## consistencycategory9   1.592e-01  1.072e-01   1.485  0.13760
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 31907  on 50417  degrees of freedom
## Residual deviance: 23449  on 50397  degrees of freedom
## AIC: 23491
##
## Number of Fisher Scoring iterations: 6
```

compare predicted to test for second fit

```
test_predict <- predict(fit2_glm, newdata = df_test, type = 'response')
table(df_test$buy, round(test_predict, 1), dnn = c('true', 'predicted'))
```

```
##     predicted
## true     0   0.1   0.2   0.3   0.4   0.5   0.6   0.7   0.8   0.9     1
##    0 24996 17776  2210   714   304   152    84    55    29    61     7
##    1   641  1640   540   280   148    94   104    84    72   816   307
```

```
test_predict[test_predict < .1] <- 0
test_predict[test_predict >= .1] <- 1

tab <- table(df_test$buy, test_predict, dnn = c('true', 'predicted'))
tab
```

```
##     predicted
## true     0     1
##    0 39113  7275
##    1  1757  2969
```

```
cat('CCR:', sum(diag(tab))/sum(tab))
```

```
## CCR: 0.8232969
```

```
cat('\nTrue targdol of predicted buyers', sum(df_test$targdol[test_predict == 1]))
```

```
##
## True targdol of predicted buyers 146164.8
```

mlogit giving weird singluarity issues

```
# install.packages('mlogit')
# library(mlogit)
# df_reshape <- mlogit.data(data = df, choice = "buy", shape = "wide")
#
# df_train <- df_reshape[df_reshape$train == 1, ]
# df_test <- df_reshape[df_reshape$train == 0, ]
#
# df2 <- df[, c('buy', 'years_since_purchase', 'consistencycategory')]
# df3 <- mlogit.data(data = df2, choice = "buy", shape = "wide")
#
# #get names
# z <- NULL
# for (name in names(df_test)) {
#   z <- paste(z, name, "+")
# }
# z
#
# #X1 + targdol + datead6 + datelp6 + lpuryear + slstyr + slslyr + sls2ago + sls3ago + slshist + ordtyr
#
# fit <- mlogit(buy ~ years_since_purchase + consistencycategory, data = df3)
#
#
# #fit <- mlogit(buy ~  datelp6 + slstyr + slslyr + sls2ago + sls3ago + slshist + ordtyr + ordlyr + ord
# summary(fit)
```