project.R

shiru

Fri Dec 22 21:03:19 2017

```
library(readr)
train <- read_csv("C:/Users/shiru/Desktop/train.csv",
                  col_types = cols(Marital_Status = col_number(),
                                   Occupation = col_number(), Product_Category_1 = col_number(),
                                   Product_Category_2 = col_number(),
                                   Product_Category_3 = col_number(),
                                   Purchase = col_number()))

mm=10000
# Replace the missing values by zero which should be reasonable in this case.
train[is.na(train)]=0
# Select the first 10000 observations to fit models.
train=train[1:mm,]

Gender=rep(0,c(mm))
Age=rep(0,c(mm))
City_CategoryA=rep(0,c(mm))
City_CategoryB=rep(0,c(mm))

# Construct dummy variable vector for "Gender".
for(i in 1:mm){
  if(train$Gender[i]=="F") Gender[i]=1
}

# Construct the "age" scores vector by 1 to 7 from the youngest intervals to the oldest intervals.
for(i in 1:mm){
  if(train$Age[i]=="0-17")
    Age[i]=1
  else if(train$Age[i]=="18-25")
    Age[i]=2
  else if(train$Age[i]=="26-35")
    Age[i]=3
  else if(train$Age[i]=="36-45")
    Age[i]=4
  else if(train$Age[i]=="46-50")
    Age[i]=5
  else if(train$Age[i]=="51-55")
    Age[i]=6
  else
    Age[i]=7
}

# Construct 2 dummy variable vector for "City_Category"
for(i in 1:mm){
  if(train$City_Category[i]=="A")
    City_CategoryA[i]=1
  else if(train$City_Category[i]=="B")
    City_CategoryB[i]=1
  else 0
}

# Construct the score vector for "Stay_In_Current_City_Years".
for(i in 1:mm){
  if(train$Stay_In_Current_City_Years[i]=="4+")
    train$Stay_In_Current_City_Years[i]="4"
}
train$Stay_In_Current_City_Years=as.numeric(train$Stay_In_Current_City_Years)

# Replacement of the orginal data variables.
head(as.numeric(train$Gender))
```

```
## Warning in head(as.numeric(train$Gender)): NAs introduced by coercion
```

```
## [1] NA NA NA NA NA NA
```

```
train$Gender=Gender
head(as.numeric(train$Age))
```

```
## Warning in head(as.numeric(train$Age)): NAs introduced by coercion
```

```
## [1] NA NA NA NA NA NA
```

```
train$Age=Age

trainm=as.data.frame(cbind(as.matrix(train[,3:5]),as.matrix(City_CategoryA),
            as.matrix(City_CategoryB),as.matrix(train[,7:12])))
names(trainm)[4:5]=c("City_CategoryA","City_CategoryB")

# Construct 20 dummy variables vectors for "Occupation"
# since there are 21 occupations.
Occupation0<-rep(0,10000)
Occupation1<-rep(0,10000)
Occupation2<-rep(0,10000)
Occupation3<-rep(0,10000)
Occupation4<-rep(0,10000)
Occupation5<-rep(0,10000)
Occupation6<-rep(0,10000)
Occupation7<-rep(0,10000)
Occupation8<-rep(0,10000)
Occupation9<-rep(0,10000)
Occupation10<-rep(0,10000)
Occupation11<-rep(0,10000)
Occupation12<-rep(0,10000)
Occupation13<-rep(0,10000)
Occupation14<-rep(0,10000)
Occupation15<-rep(0,10000)
Occupation16<-rep(0,10000)
Occupation17<-rep(0,10000)
Occupation18<-rep(0,10000)
Occupation19<-rep(0,10000)
for(i in 1:10000){
  if(trainm$Occupation[i]==0)
    Occupation0[i]=1
  else if (trainm$Occupation[i]==1)
    Occupation1[i]=1
  else if (trainm$Occupation[i]==2)
    Occupation2[i]=1
  else if (trainm$Occupation[i]==3)
    Occupation3[i]=1
  else if (trainm$Occupation[i]==4)
    Occupation4[i]=1
  else if (trainm$Occupation[i]==5)
    Occupation5[i]=1
  else if (trainm$Occupation[i]==6)
    Occupation6[i]=1
  else if (trainm$Occupation[i]==7)
    Occupation7[i]=1
  else if (trainm$Occupation[i]==8)
    Occupation8[i]=1
  else if (trainm$Occupation[i]==9)
    Occupation9[i]=1
  else if (trainm$Occupation[i]==10)
    Occupation10[i]=1
  else if (trainm$Occupation[i]==11)
    Occupation11[i]=1
  else if (trainm$Occupation[i]==12)
    Occupation12[i]=1
  else if (trainm$Occupation[i]==13)
    Occupation13[i]=1
  else if (trainm$Occupation[i]==14)
    Occupation14[i]=1
  else if (trainm$Occupation[i]==15)
    Occupation15[i]=1
  else if (trainm$Occupation[i]==16)
    Occupation16[i]=1
  else if (trainm$Occupation[i]==17)
    Occupation17[i]=1
  else if (trainm$Occupation[i]==18)
```

```
    Occupation18[i]=1
  else if (trainm$Occupation[i]==19)
    Occupation19[i]=1
  else
    0
}

# Construct 17 dummy variables vectors for "Product_Category_1"
# since there are 18 categorys in "Product_Category_1".
Product_Category_11=rep(0,10000)
Product_Category_12=rep(0,10000)
Product_Category_13=rep(0,10000)
Product_Category_14=rep(0,10000)
Product_Category_15=rep(0,10000)
Product_Category_16=rep(0,10000)
Product_Category_17=rep(0,10000)
Product_Category_18=rep(0,10000)
Product_Category_19=rep(0,10000)
Product_Category_110=rep(0,10000)
Product_Category_111=rep(0,10000)
Product_Category_112=rep(0,10000)
Product_Category_113=rep(0,10000)
Product_Category_114=rep(0,10000)
Product_Category_115=rep(0,10000)
Product_Category_116=rep(0,10000)
Product_Category_117=rep(0,10000)
for(i in 1:10000){
  if(trainm$Product_Category_1[i]==1)
    Product_Category_11[i]=1
  else if(trainm$Product_Category_1[i]==2)
    Product_Category_12[i]=1
  else if(trainm$Product_Category_1[i]==3)
    Product_Category_13[i]=1
  else if(trainm$Product_Category_1[i]==4)
    Product_Category_14[i]=1
  else if(trainm$Product_Category_1[i]==5)
    Product_Category_15[i]=1
  else if(trainm$Product_Category_1[i]==6)
    Product_Category_16[i]=1
  else if(trainm$Product_Category_1[i]==7)
    Product_Category_17[i]=1
  else if(trainm$Product_Category_1[i]==8)
    Product_Category_18[i]=1
  else if(trainm$Product_Category_1[i]==9)
    Product_Category_19[i]=1
  else if(trainm$Product_Category_1[i]==10)
    Product_Category_110[i]=1
  else if(trainm$Product_Category_1[i]==11)
    Product_Category_111[i]=1
  else if(trainm$Product_Category_1[i]==12)
    Product_Category_112[i]=1
  else if(trainm$Product_Category_1[i]==13)
    Product_Category_113[i]=1
  else if(trainm$Product_Category_1[i]==14)
    Product_Category_114[i]=1
  else if(trainm$Product_Category_1[i]==15)
    Product_Category_115[i]=1
  else if(trainm$Product_Category_1[i]==16)
    Product_Category_116[i]=1
  else if(trainm$Product_Category_1[i]==17)
    Product_Category_117[i]=1
  else
    0
}

# Construct 18 dummy variables vectors for "Product_Category_2"
# since there are 19 categorys in "Product_Category_2".
Product_Category_20=rep(0,10000)
Product_Category_21=rep(0,10000)
Product_Category_22=rep(0,10000)
Product_Category_23=rep(0,10000)
Product_Category_24=rep(0,10000)
Product_Category_25=rep(0,10000)
Product_Category_26=rep(0,10000)
Product_Category_27=rep(0,10000)
Product_Category_28=rep(0,10000)
```

```
Product_Category_29=rep(0,10000)
Product_Category_210=rep(0,10000)
Product_Category_211=rep(0,10000)
Product_Category_212=rep(0,10000)
Product_Category_213=rep(0,10000)
Product_Category_214=rep(0,10000)
Product_Category_215=rep(0,10000)
Product_Category_216=rep(0,10000)
Product_Category_217=rep(0,10000)
for(i in 1:10000){
  if(trainm$Product_Category_2[i]==0)
    Product_Category_20[i]=1
  else if(trainm$Product_Category_2[i]==1)
    Product_Category_21[i]=1
  else if(trainm$Product_Category_2[i]==2)
    Product_Category_22[i]=1
  else if(trainm$Product_Category_2[i]==3)
    Product_Category_23[i]=1
  else if(trainm$Product_Category_2[i]==4)
    Product_Category_24[i]=1
  else if(trainm$Product_Category_2[i]==5)
    Product_Category_25[i]=1
  else if(trainm$Product_Category_2[i]==6)
    Product_Category_26[i]=1
  else if(trainm$Product_Category_2[i]==7)
    Product_Category_27[i]=1
  else if(trainm$Product_Category_2[i]==8)
    Product_Category_28[i]=1
  else if(trainm$Product_Category_2[i]==9)
    Product_Category_29[i]=1
  else if(trainm$Product_Category_2[i]==10)
    Product_Category_210[i]=1
  else if(trainm$Product_Category_2[i]==11)
    Product_Category_211[i]=1
  else if(trainm$Product_Category_2[i]==12)
    Product_Category_212[i]=1
  else if(trainm$Product_Category_2[i]==13)
    Product_Category_213[i]=1
  else if(trainm$Product_Category_2[i]==14)
    Product_Category_214[i]=1
  else if(trainm$Product_Category_2[i]==15)
    Product_Category_215[i]=1
  else if(trainm$Product_Category_2[i]==16)
    Product_Category_216[i]=1
  else if(trainm$Product_Category_2[i]==17)
    Product_Category_217[i]=1
  else
    0
}

# Construct 18 dummy variables vectors for "Product_Category_3"
# since there are 19 categorys in "Product_Category_3".
Product_Category_30=rep(0,10000)
Product_Category_31=rep(0,10000)
Product_Category_32=rep(0,10000)
Product_Category_33=rep(0,10000)
Product_Category_34=rep(0,10000)
Product_Category_35=rep(0,10000)
Product_Category_36=rep(0,10000)
Product_Category_37=rep(0,10000)
Product_Category_38=rep(0,10000)
Product_Category_39=rep(0,10000)
Product_Category_310=rep(0,10000)
Product_Category_311=rep(0,10000)
Product_Category_312=rep(0,10000)
Product_Category_313=rep(0,10000)
Product_Category_314=rep(0,10000)
Product_Category_315=rep(0,10000)
Product_Category_316=rep(0,10000)
Product_Category_317=rep(0,10000)
for(i in 1:10000){
  if(trainm$Product_Category_3[i]==0)
    Product_Category_30[i]=1
  else if(trainm$Product_Category_3[i]==1)
    Product_Category_31[i]=1
  else if(trainm$Product_Category_3[i]==2)
```

```
    Product_Category_32[i]=1
  else if(trainm$Product_Category_3[i]==3)
    Product_Category_33[i]=1
  else if(trainm$Product_Category_3[i]==4)
    Product_Category_34[i]=1
  else if(trainm$Product_Category_3[i]==5)
    Product_Category_35[i]=1
  else if(trainm$Product_Category_3[i]==6)
    Product_Category_36[i]=1
  else if(trainm$Product_Category_3[i]==7)
    Product_Category_37[i]=1
  else if(trainm$Product_Category_3[i]==8)
    Product_Category_38[i]=1
  else if(trainm$Product_Category_3[i]==9)
    Product_Category_39[i]=1
  else if(trainm$Product_Category_3[i]==10)
    Product_Category_310[i]=1
  else if(trainm$Product_Category_3[i]==11)
    Product_Category_311[i]=1
  else if(trainm$Product_Category_3[i]==12)
    Product_Category_312[i]=1
  else if(trainm$Product_Category_3[i]==13)
    Product_Category_313[i]=1
  else if(trainm$Product_Category_3[i]==14)
    Product_Category_314[i]=1
  else if(trainm$Product_Category_3[i]==15)
    Product_Category_315[i]=1
  else if(trainm$Product_Category_3[i]==16)
    Product_Category_316[i]=1
  else if(trainm$Product_Category_3[i]==17)
    Product_Category_317[i]=1
  else
    0
}

# New dataframe after processing.
trainm=data.frame(cbind(as.matrix(trainm[,1:2]),Occupation0,Occupation1,Occupation2,Occupation3,
                        Occupation4,Occupation5,Occupation6,Occupation7,Occupation8,Occupation9,
                        Occupation10,Occupation11,Occupation12,Occupation13,Occupation14,Occupation15,
                        Occupation16,Occupation17,Occupation18,Occupation19,as.matrix(trainm[4:7]),
                        Product_Category_11,Product_Category_12,Product_Category_13,Product_Category_14,
                        Product_Category_15,Product_Category_16,Product_Category_17,Product_Category_18,
                        Product_Category_19,Product_Category_110,Product_Category_111,Product_Category_112,
                        Product_Category_113,Product_Category_114,Product_Category_115,Product_Category_116,
                        Product_Category_117,Product_Category_20,Product_Category_21,Product_Category_22,
                        Product_Category_23,Product_Category_24,Product_Category_25,Product_Category_26,
                        Product_Category_27,Product_Category_28,Product_Category_29,Product_Category_210,
                        Product_Category_211,Product_Category_212,Product_Category_213,Product_Category_214,
                        Product_Category_215,Product_Category_216,Product_Category_217,Product_Category_30,
                        Product_Category_31,Product_Category_32,Product_Category_33,Product_Category_34,
                        Product_Category_35,Product_Category_36,Product_Category_37,Product_Category_38,
                        Product_Category_39,Product_Category_310,Product_Category_311,Product_Category_312,
                        Product_Category_313,Product_Category_314,Product_Category_315,Product_Category_316,
                        Product_Category_317,as.matrix(trainm[,11])))
head(trainm)
```

```
##   Gender Age Occupation0 Occupation1 Occupation2 Occupation3 Occupation4
## 1      1   1           0           0           0           0           0
## 2      1   1           0           0           0           0           0
## 3      1   1           0           0           0           0           0
## 4      1   1           0           0           0           0           0
## 5      0   7           0           0           0           0           0
## 6      0   3           0           0           0           0           0
##   Occupation5 Occupation6 Occupation7 Occupation8 Occupation9 Occupation10
## 1           0           0           0           0           0            1
## 2           0           0           0           0           0            1
## 3           0           0           0           0           0            1
## 4           0           0           0           0           0            1
## 5           0           0           0           0           0            0
## 6           0           0           0           0           0            0
##   Occupation11 Occupation12 Occupation13 Occupation14 Occupation15
## 1            0            0            0            0            0
## 2            0            0            0            0            0
## 3            0            0            0            0            0
## 4            0            0            0            0            0
```

```
## 5            0            0            0            0            0
## 6            0            0            0            0            1
##   Occupation16 Occupation17 Occupation18 Occupation19 City_CategoryA
## 1            0            0            0            0              1
## 2            0            0            0            0              1
## 3            0            0            0            0              1
## 4            0            0            0            0              1
## 5            1            0            0            0              0
## 6            0            0            0            0              1
##   City_CategoryB Stay_In_Current_City_Years Marital_Status
## 1              0                          2              0
## 2              0                          2              0
## 3              0                          2              0
## 4              0                          2              0
## 5              0                          4              0
## 6              0                          3              0
##   Product_Category_11 Product_Category_12 Product_Category_13
## 1                   0                   0                   1
## 2                   1                   0                   0
## 3                   0                   0                   0
## 4                   0                   0                   0
## 5                   0                   0                   0
## 6                   1                   0                   0
##   Product_Category_14 Product_Category_15 Product_Category_16
## 1                   0                   0                   0
## 2                   0                   0                   0
## 3                   0                   0                   0
## 4                   0                   0                   0
## 5                   0                   0                   0
## 6                   0                   0                   0
##   Product_Category_17 Product_Category_18 Product_Category_19
## 1                   0                   0                   0
## 2                   0                   0                   0
## 3                   0                   0                   0
## 4                   0                   0                   0
## 5                   0                   1                   0
## 6                   0                   0                   0
##   Product_Category_110 Product_Category_111 Product_Category_112
## 1                    0                    0                    0
## 2                    0                    0                    0
## 3                    0                    0                    1
## 4                    0                    0                    1
## 5                    0                    0                    0
## 6                    0                    0                    0
##   Product_Category_113 Product_Category_114 Product_Category_115
## 1                    0                    0                    0
## 2                    0                    0                    0
## 3                    0                    0                    0
## 4                    0                    0                    0
## 5                    0                    0                    0
## 6                    0                    0                    0
##   Product_Category_116 Product_Category_117 Product_Category_20
## 1                    0                    0                   1
## 2                    0                    0                   0
## 3                    0                    0                   1
## 4                    0                    0                   0
## 5                    0                    0                   1
## 6                    0                    0                   0
##   Product_Category_21 Product_Category_22 Product_Category_23
## 1                   0                   0                   0
## 2                   0                   0                   0
## 3                   0                   0                   0
## 4                   0                   0                   0
## 5                   0                   0                   0
## 6                   0                   1                   0
##   Product_Category_24 Product_Category_25 Product_Category_26
## 1                   0                   0                   0
## 2                   0                   0                   1
## 3                   0                   0                   0
## 4                   0                   0                   0
## 5                   0                   0                   0
## 6                   0                   0                   0
##   Product_Category_27 Product_Category_28 Product_Category_29
## 1                   0                   0                   0
## 2                   0                   0                   0
## 3                   0                   0                   0
```

```
## 4                    0                    0                    0
## 5                    0                    0                    0
## 6                    0                    0                    0
##   Product_Category_210 Product_Category_211 Product_Category_212
## 1                    0                    0                    0
## 2                    0                    0                    0
## 3                    0                    0                    0
## 4                    0                    0                    0
## 5                    0                    0                    0
## 6                    0                    0                    0
##   Product_Category_213 Product_Category_214 Product_Category_215
## 1                    0                    0                    0
## 2                    0                    0                    0
## 3                    0                    0                    0
## 4                    0                    1                    0
## 5                    0                    0                    0
## 6                    0                    0                    0
##   Product_Category_216 Product_Category_217 Product_Category_30
## 1                    0                    0                   1
## 2                    0                    0                   0
## 3                    0                    0                   1
## 4                    0                    0                   1
## 5                    0                    0                   1
## 6                    0                    0                   1
##   Product_Category_31 Product_Category_32 Product_Category_33
## 1                   0                   0                   0
## 2                   0                   0                   0
## 3                   0                   0                   0
## 4                   0                   0                   0
## 5                   0                   0                   0
## 6                   0                   0                   0
##   Product_Category_34 Product_Category_35 Product_Category_36
## 1                   0                   0                   0
## 2                   0                   0                   0
## 3                   0                   0                   0
## 4                   0                   0                   0
## 5                   0                   0                   0
## 6                   0                   0                   0
##   Product_Category_37 Product_Category_38 Product_Category_39
## 1                   0                   0                   0
## 2                   0                   0                   0
## 3                   0                   0                   0
## 4                   0                   0                   0
## 5                   0                   0                   0
## 6                   0                   0                   0
##   Product_Category_310 Product_Category_311 Product_Category_312
## 1                    0                    0                    0
## 2                    0                    0                    0
## 3                    0                    0                    0
## 4                    0                    0                    0
## 5                    0                    0                    0
## 6                    0                    0                    0
##   Product_Category_313 Product_Category_314 Product_Category_315
## 1                    0                    0                    0
## 2                    0                    1                    0
## 3                    0                    0                    0
## 4                    0                    0                    0
## 5                    0                    0                    0
## 6                    0                    0                    0
##   Product_Category_316 Product_Category_317   V80
## 1                    0                    0  8370
## 2                    0                    0 15200
## 3                    0                    0  1422
## 4                    0                    0  1057
## 5                    0                    0  7969
## 6                    0                    0 15227
```

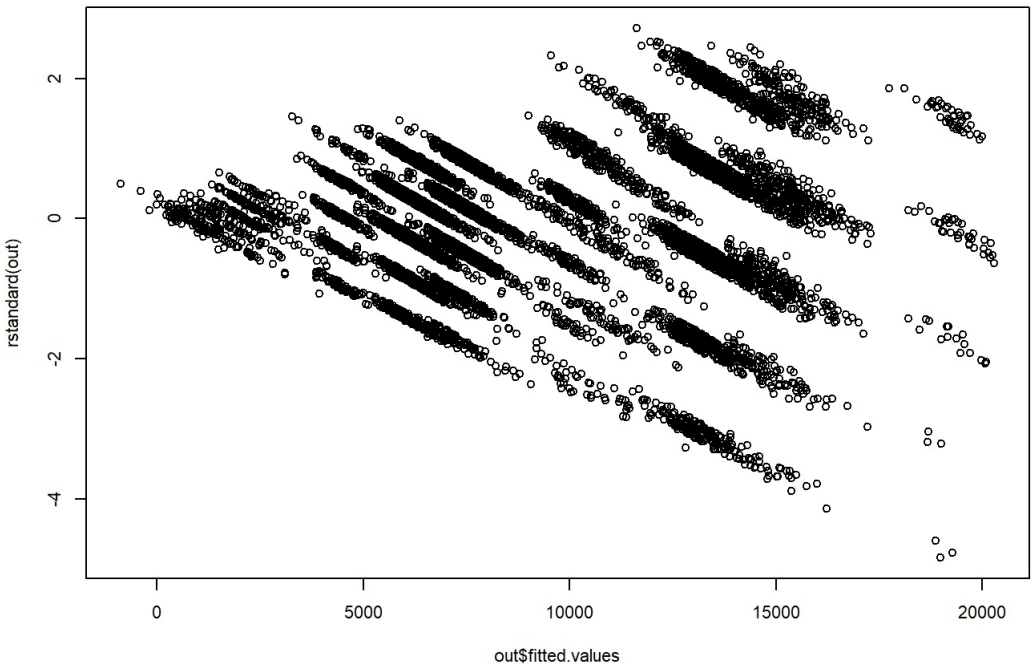
```
# Fit linear regression model first.
xx<-as.matrix(trainm[,1:79])
y<-as.matrix(trainm[,80])
out=lm(y~xx)
summary(out)
```

```
##
```

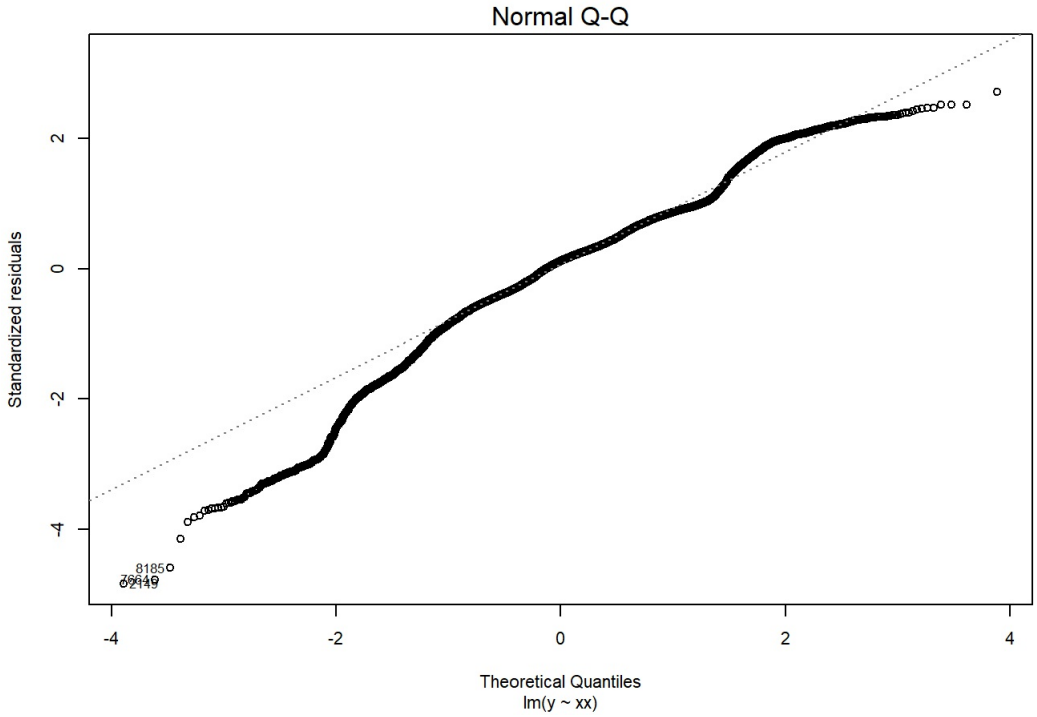
```
## Call:
## lm(formula = y ~ xx)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14154.6  -1521.4    339.8   1918.9   7714.8
##
## Coefficients: (4 not defined because of singularities)
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   2878.544    776.880   3.705 0.000212 ***
## xxGender                      -104.348     74.152  -1.407 0.159396
## xxAge                           58.685     28.833   2.035 0.041845 *
## xxOccupation0                  546.237    138.673   3.939 8.24e-05 ***
## xxOccupation1                  420.061    152.552   2.754 0.005906 **
## xxOccupation2                  339.352    171.113   1.983 0.047372 *
## xxOccupation3                  822.841    177.156   4.645 3.45e-06 ***
## xxOccupation4                  405.124    141.788   2.857 0.004282 **
## xxOccupation5                 1192.320    283.476   4.206 2.62e-05 ***
## xxOccupation6                  874.086    211.610   4.131 3.65e-05 ***
## xxOccupation7                  169.779    144.443   1.175 0.239860
## xxOccupation8                   22.374    433.319   0.052 0.958821
## xxOccupation9                  581.039    314.508   1.847 0.064711 .
## xxOccupation10                 671.689    201.787   3.329 0.000876 ***
## xxOccupation11                1128.677    229.827   4.911 9.21e-07 ***
## xxOccupation12                 613.636    163.213   3.760 0.000171 ***
## xxOccupation13                 749.939    266.665   2.812 0.004929 **
## xxOccupation14                 214.342    184.792   1.160 0.246116
## xxOccupation15                 798.557    240.982   3.314 0.000924 ***
## xxOccupation16                 154.407    177.043   0.872 0.383153
## xxOccupation17                 661.622    153.114   4.321 1.57e-05 ***
## xxOccupation18                -270.767    343.285  -0.789 0.430276
## xxOccupation19                -474.193    244.041  -1.943 0.052034 .
## xxCity_CategoryA              -548.112     77.984  -7.028 2.23e-12 ***
## xxCity_CategoryB              -385.258     74.621  -5.163 2.48e-07 ***
## xxStay_In_Current_City_Years    17.575     23.005   0.764 0.444919
## xxMarital_Status                -1.711     65.843  -0.026 0.979274
## xxProduct_Category_11        10313.006    466.147  22.124  < 2e-16 ***
## xxProduct_Category_12         8385.696    497.304  16.862  < 2e-16 ***
## xxProduct_Category_13         8089.121    530.430  15.250  < 2e-16 ***
## xxProduct_Category_14         -522.152    522.351  -1.000 0.317519
## xxProduct_Category_15         3234.801    457.798   7.066 1.70e-12 ***
## xxProduct_Category_16        12312.252    493.028  24.973  < 2e-16 ***
## xxProduct_Category_17        13472.135    565.560  23.821  < 2e-16 ***
## xxProduct_Category_18         4494.732    458.126   9.811  < 2e-16 ***
## xxProduct_Category_19        11557.326   1299.736   8.892  < 2e-16 ***
## xxProduct_Category_110       16491.462    554.181  29.758  < 2e-16 ***
## xxProduct_Category_111        1636.375    476.001   3.438 0.000589 ***
## xxProduct_Category_112       -1583.784    563.561  -2.810 0.004959 **
## xxProduct_Category_113       -2107.771    556.822  -3.785 0.000154 ***
## xxProduct_Category_114        9475.212    728.366  13.009  < 2e-16 ***
## xxProduct_Category_115       11988.521    536.099  22.363  < 2e-16 ***
## xxProduct_Category_116       11467.687    509.739  22.497  < 2e-16 ***
## xxProduct_Category_117        7565.948    938.349   8.063 8.30e-16 ***
## xxProduct_Category_20         -811.172    472.398  -1.717 0.085985 .
## xxProduct_Category_21               NA         NA      NA       NA
## xxProduct_Category_22         -316.693    494.503  -0.640 0.521910
## xxProduct_Category_23          224.194    825.896   0.271 0.786047
## xxProduct_Category_24        -1974.957    544.483  -3.627 0.000288 ***
## xxProduct_Category_25        -1043.561    507.649  -2.056 0.039840 *
## xxProduct_Category_26         -658.465    507.109  -1.298 0.194156
## xxProduct_Category_27         -245.179    947.194  -0.259 0.795758
## xxProduct_Category_28         -397.623    480.067  -0.828 0.407540
## xxProduct_Category_29         -889.737    554.281  -1.605 0.108479
## xxProduct_Category_210         778.413    657.249   1.184 0.236302
## xxProduct_Category_211        -528.555    510.085  -1.036 0.300129
## xxProduct_Category_212        -895.530    571.404  -1.567 0.117089
## xxProduct_Category_213        -790.690    525.078  -1.506 0.132137
## xxProduct_Category_214        -799.429    479.277  -1.668 0.095349 .
## xxProduct_Category_215        -650.729    486.893  -1.336 0.181419
## xxProduct_Category_216        -703.029    483.807  -1.453 0.146223
## xxProduct_Category_217         -23.684    507.798  -0.047 0.962801
## xxProduct_Category_30          462.519    375.781   1.231 0.218419
## xxProduct_Category_31               NA         NA      NA       NA
## xxProduct_Category_32               NA         NA      NA       NA
## xxProduct_Category_33         2121.246    859.008   2.469 0.013550 *
## xxProduct_Category_34        -2033.411    905.655  -2.245 0.024775 *
```

```
## xxProduct_Category_35           839.797    427.911   1.963 0.049726 *
## xxProduct_Category_36           -45.061    533.008  -0.085 0.932628
## xxProduct_Category_37                NA         NA      NA       NA
## xxProduct_Category_38          2110.076    426.265   4.950 7.54e-07 ***
## xxProduct_Category_39            31.289    436.976   0.072 0.942919
## xxProduct_Category_310          644.557    828.781   0.778 0.436754
## xxProduct_Category_311          461.611    658.213   0.701 0.483127
## xxProduct_Category_312          683.279    447.125   1.528 0.126505
## xxProduct_Category_313          -57.655    500.832  -0.115 0.908354
## xxProduct_Category_314          595.377    405.488   1.468 0.142055
## xxProduct_Category_315         -130.070    399.845  -0.325 0.744960
## xxProduct_Category_316           91.895    391.087   0.235 0.814235
## xxProduct_Category_317         1426.004    406.942   3.504 0.000460 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2959 on 9924 degrees of freedom
## Multiple R-squared:  0.6352, Adjusted R-squared:  0.6325
## F-statistic: 230.4 on 75 and 9924 DF,  p-value: < 2.2e-16
```

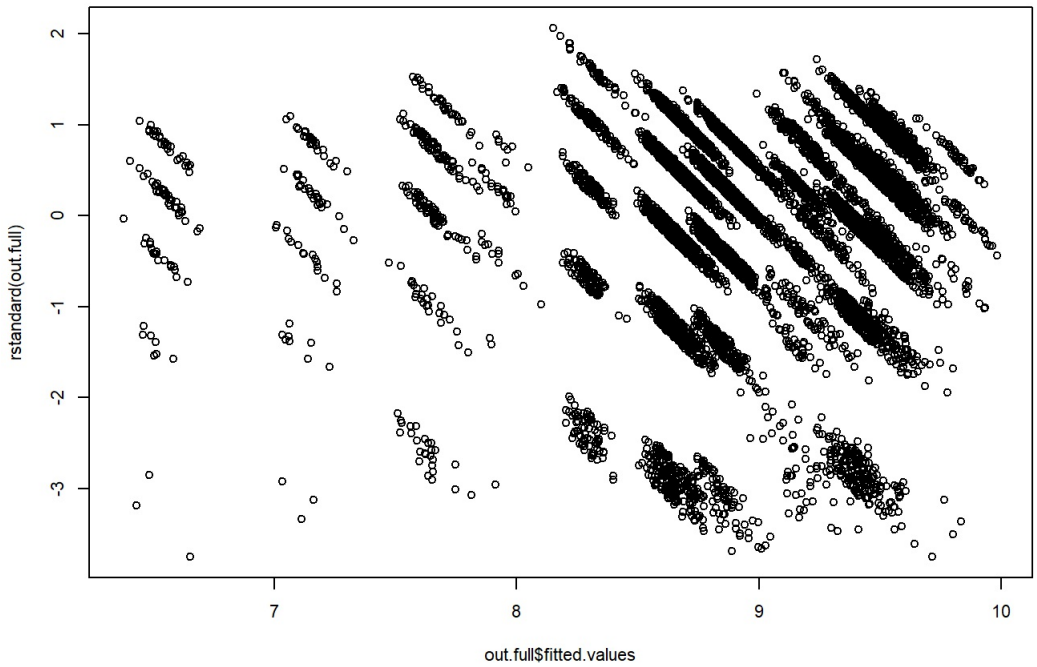
```
par(cex=0.7)
# Plot standardized residuals against fitted values.
plot(out$fitted.values,rstandard(out))
```



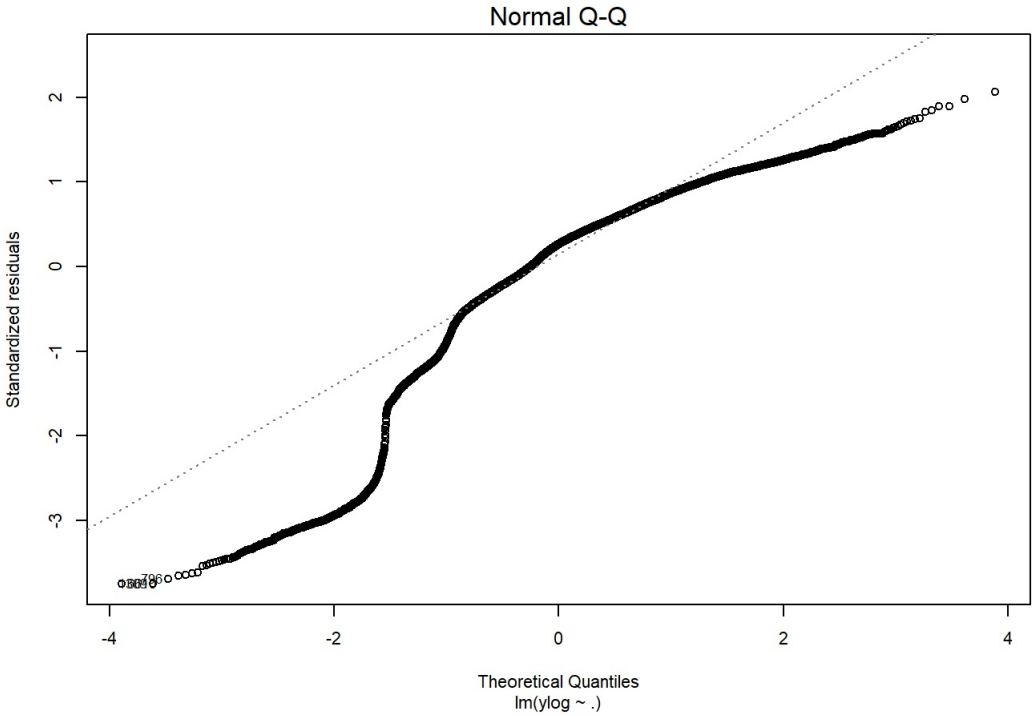

```
# QQ plot.
plot(out,2)
```



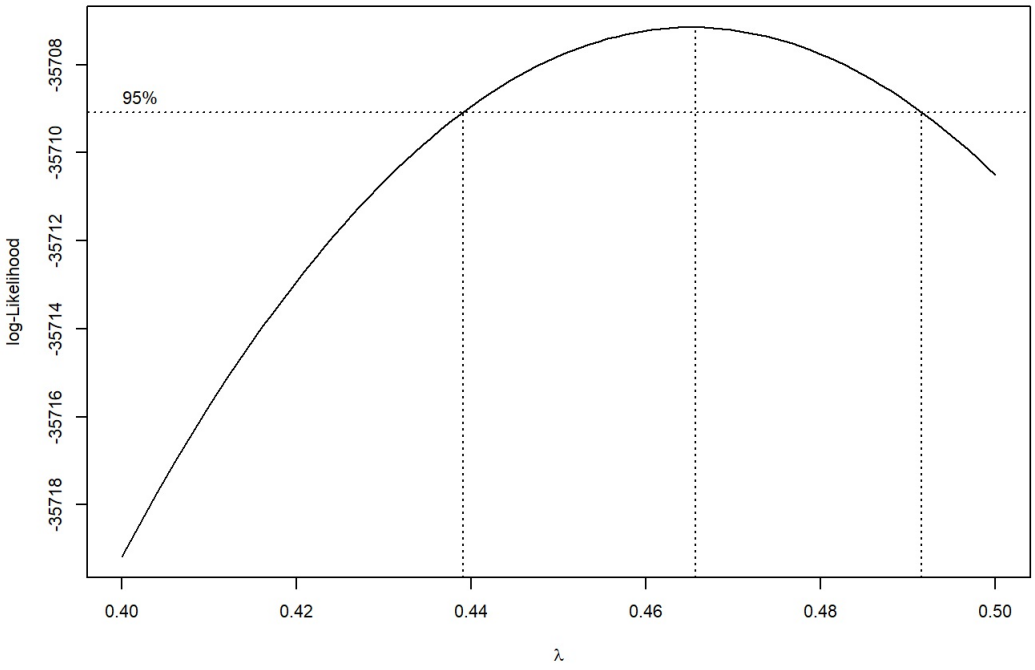

```
# Remove the variables which are zero in all observations.
xx<-xx[,-c(45,63,64,69)]

# Make log transformation to y.
ylog=log(y)
out.full=lm(ylog~.,data=as.data.frame(xx))
# Standardized residuals plot become acceptable.
plot(out.full$fitted.values,rstandard(out.full))
```



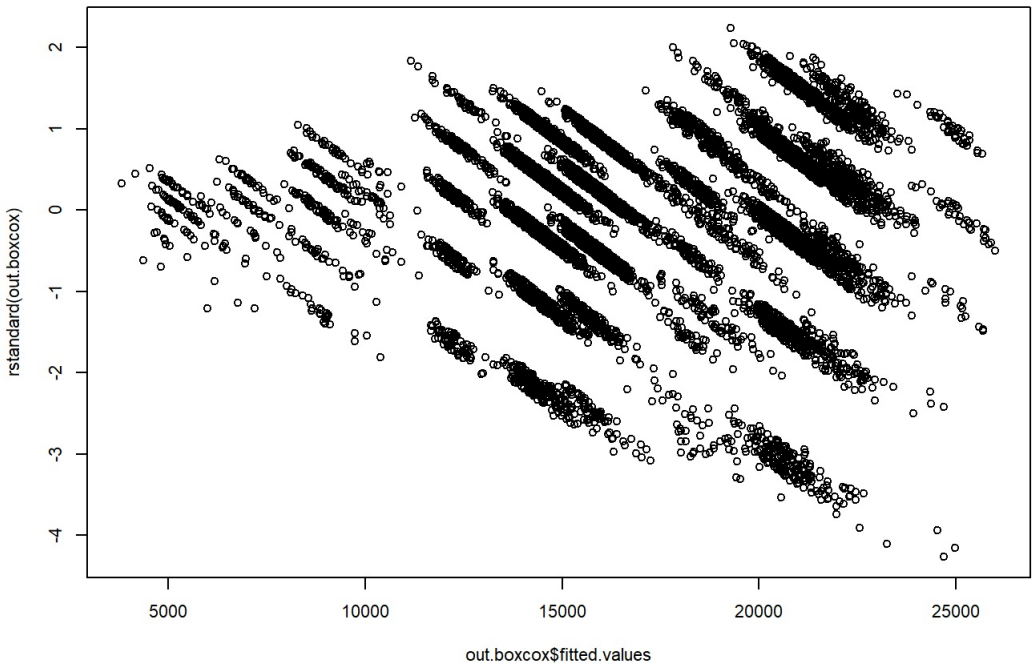

```
plot(out.full,2)
```



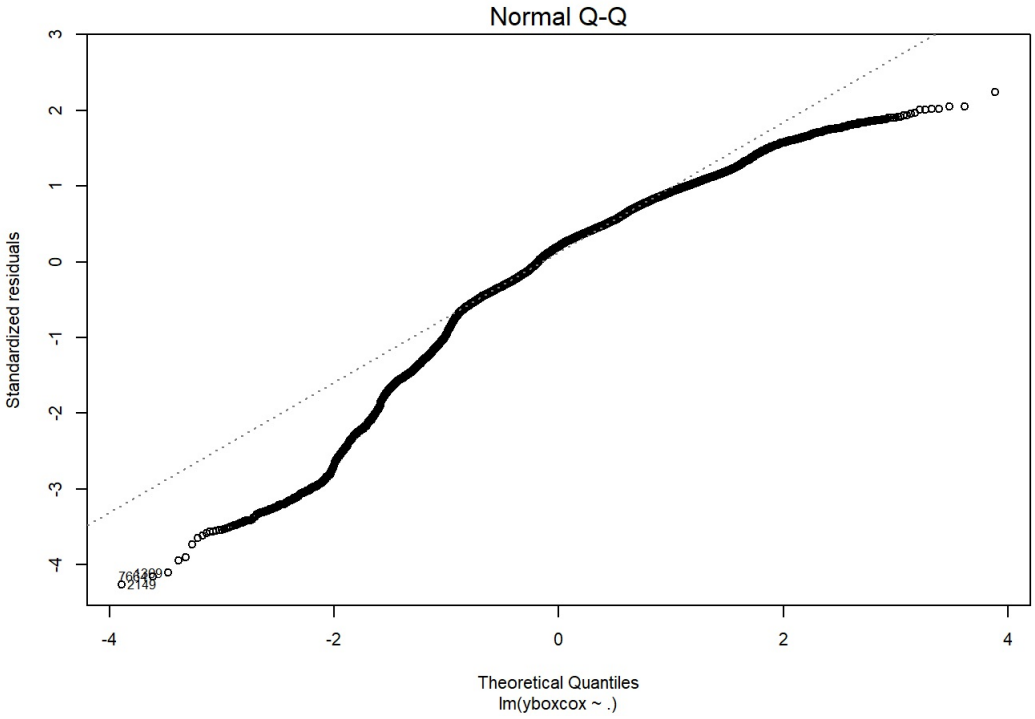

```
# try Box-Cox method.
library(MASS)
boxcox(out,lambda=seq(0.4,0.5,1/100),plotit=TRUE)
```




```
# lambda=0.465
lamb=0.465
yboxcox=(y^lamb-1)/lamb/prod(y^(1/10000))^(lamb-1)
out.boxcox=lm(yboxcox~.,data=as.data.frame(xx))
plot(out.boxcox$fitted.values,rstandard(out.boxcox))
```




```
plot(out.boxcox,2)
```



```
# Step-wise backward selection.
xx1=xx
repeat{out2=lm(ylog~xx1)
out22=summary(out2)
if(out22$coefficients[which.max(out22$coefficients[,4]),4]<0.05) break
j=as.numeric(which.max(out22$coefficients[,4])-1)
xx1<-xx1[,-j]
}

# Stepwise forward, backward and hybrid selection.(use AIC as criterion)
out.null=lm(ylog~1,data=as.data.frame(xx))
full=formula(lm(ylog~.,as.data.frame(xx)))
out.forward=step(out.null,scope=list(lower=~1,upper=full),direction="forward",
                 data=as.data.frame(xx),trace=F)

out.backward=step(out.full,scope=list(lower=~1,upper=full),direction="backward",
                 data=as.data.frame(xx),trace=F)

out.both=step(out.full,scope=list(lower=~1,upper=full),direction="both",
                 data=as.data.frame(xx),trace=F)

# Compare adj.r.squared of the four models
out22$adj.r.squared
```

```
## [1] 0.6471029
```

```
summary(out.forward)$adj.r.squared
```

```
## [1] 0.6474938
```

```
summary(out.backward)$adj.r.squared
```

```
## [1] 0.6474132
```

```
summary(out.both)$adj.r.squared
```

```
## [1] 0.6474132
```

```
# K-fold Cross validation
library(boot)
set.seed(123)
print(cv.glm(data=as.data.frame(xx),
             glm(ylog~xx1,data=as.data.frame(xx)),K=5)$delta[1])
```

```
## Warning: 'newdata' had 2000 rows but variables found have 10000 rows
```

```
## Warning: 'newdata' had 2000 rows but variables found have 10000 rows

## Warning: 'newdata' had 2000 rows but variables found have 10000 rows

## Warning: 'newdata' had 2000 rows but variables found have 10000 rows

## Warning: 'newdata' had 2000 rows but variables found have 10000 rows
```

```
## [1] 0.6810173
```

```
xx2<-cbind(Product_Category_11,Product_Category_113,
           Product_Category_14,Product_Category_112,Product_Category_15,
           Product_Category_111,Product_Category_18,Product_Category_24,
           Product_Category_110,Product_Category_16,Product_Category_17,
           Product_Category_116,Product_Category_115,Product_Category_20,
           City_CategoryA,Product_Category_12,Product_Category_13,
           Product_Category_114,Product_Category_117,Product_Category_19,
           Product_Category_38,City_CategoryB,Age , Product_Category_317 ,
           Occupation11,Occupation5 , Occupation19 , Occupation17 ,
           Product_Category_28 , Product_Category_217 , Occupation7 ,
           Product_Category_35 , Occupation6 , Occupation12 , Occupation15 ,
           Occupation3 , Occupation0 , Occupation10 , Product_Category_33 ,
           Occupation1 , Product_Category_210 , Occupation13 , Occupation4 ,
           Occupation2 , Product_Category_25 , Product_Category_34 ,
           Product_Category_312 , Product_Category_316 , Product_Category_315 ,
           Occupation14 , Occupation9 , Occupation16)
print(cv.glm(data=as.data.frame(xx),
             glm(ylog~xx2,data=as.data.frame(xx)),K=5)$delta[1])
```

```
## Warning: 'newdata' had 2000 rows but variables found have 10000 rows

## Warning: 'newdata' had 2000 rows but variables found have 10000 rows

## Warning: 'newdata' had 2000 rows but variables found have 10000 rows

## Warning: 'newdata' had 2000 rows but variables found have 10000 rows

## Warning: 'newdata' had 2000 rows but variables found have 10000 rows
```

```
## [1] 0.6800064
```

```
xx3<-cbind(Age , Occupation0 , Occupation1 , Occupation2 ,
             Occupation3 , Occupation4 , Occupation5 , Occupation6 , Occupation7 ,
             Occupation9 , Occupation10 , Occupation11 , Occupation12 ,
             Occupation13 , Occupation14 , Occupation15 , Occupation16 ,
             Occupation17 , City_CategoryA , City_CategoryB , Product_Category_11 ,
             Product_Category_12 , Product_Category_13 , Product_Category_14 ,
             Product_Category_15 , Product_Category_16 , Product_Category_17 ,
             Product_Category_18 , Product_Category_19 , Product_Category_110 ,
             Product_Category_111 , Product_Category_112 , Product_Category_113 ,
             Product_Category_114 , Product_Category_115 , Product_Category_116 ,
             Product_Category_117 , Product_Category_20 , Product_Category_24 ,
             Product_Category_25 , Product_Category_26 , Product_Category_210 ,
             Product_Category_211 , Product_Category_212 , Product_Category_213 ,
             Product_Category_214 , Product_Category_215 , Product_Category_216 ,
             Product_Category_217 , Product_Category_30 , Product_Category_33 ,
             Product_Category_34 , Product_Category_35 , Product_Category_38 ,
             Product_Category_312 , Product_Category_314 , Product_Category_317)
print(cv.glm(data=as.data.frame(xx),
             glm(ylog~xx3,data=as.data.frame(xx)),K=5)$delta[1])
```

```
## Warning: 'newdata' had 2000 rows but variables found have 10000 rows

## Warning: 'newdata' had 2000 rows but variables found have 10000 rows

## Warning: 'newdata' had 2000 rows but variables found have 10000 rows

## Warning: 'newdata' had 2000 rows but variables found have 10000 rows

## Warning: 'newdata' had 2000 rows but variables found have 10000 rows
```

```
## [1] 0.6851477
```

```
xx4<-cbind(Age , Occupation0 , Occupation1 , Occupation2 ,
           Occupation3 , Occupation4 , Occupation5 , Occupation6 , Occupation7 ,
           Occupation9 , Occupation10 , Occupation11 , Occupation12 ,
           Occupation13 , Occupation14 , Occupation15 , Occupation16 ,
           Occupation17 , City_CategoryA , City_CategoryB , Product_Category_11 ,
           Product_Category_12 , Product_Category_13 , Product_Category_14 ,
           Product_Category_15 , Product_Category_16 , Product_Category_17 ,
           Product_Category_18 , Product_Category_19 , Product_Category_110 ,
           Product_Category_111 , Product_Category_112 , Product_Category_113 ,
           Product_Category_114 , Product_Category_115 , Product_Category_116 ,
           Product_Category_117 , Product_Category_20 , Product_Category_24 ,
           Product_Category_25 , Product_Category_26 , Product_Category_210 ,
           Product_Category_211 , Product_Category_212 , Product_Category_213 ,
           Product_Category_214 , Product_Category_215 , Product_Category_216 ,
           Product_Category_217 , Product_Category_30 , Product_Category_33 ,
           Product_Category_34 , Product_Category_35 , Product_Category_38)
print(cv.glm(data=as.data.frame(xx),
             glm(ylog~xx4,data=as.data.frame(xx)),K=5)$delta[1])
```

```
## Warning: 'newdata' had 2000 rows but variables found have 10000 rows

## Warning: 'newdata' had 2000 rows but variables found have 10000 rows

## Warning: 'newdata' had 2000 rows but variables found have 10000 rows

## Warning: 'newdata' had 2000 rows but variables found have 10000 rows

## Warning: 'newdata' had 2000 rows but variables found have 10000 rows
```

```
## [1] 0.6726323
```

```
# Check Muticollinearity.
xxforward=xx[,-c(1,11,21,25,26,45,46,49,50,52,
                 54,55,56,57,58,59,61,65,67,68,69,71,72)]
mean(diag(solve(cor(xxforward))))
```

```
## [1] 4.99778
```

```
max(diag(solve(cor(xxforward))))
```

```
## [1] 48.6881
```

```
# Check influential observations
which(cooks.distance(out.forward)>1)
```

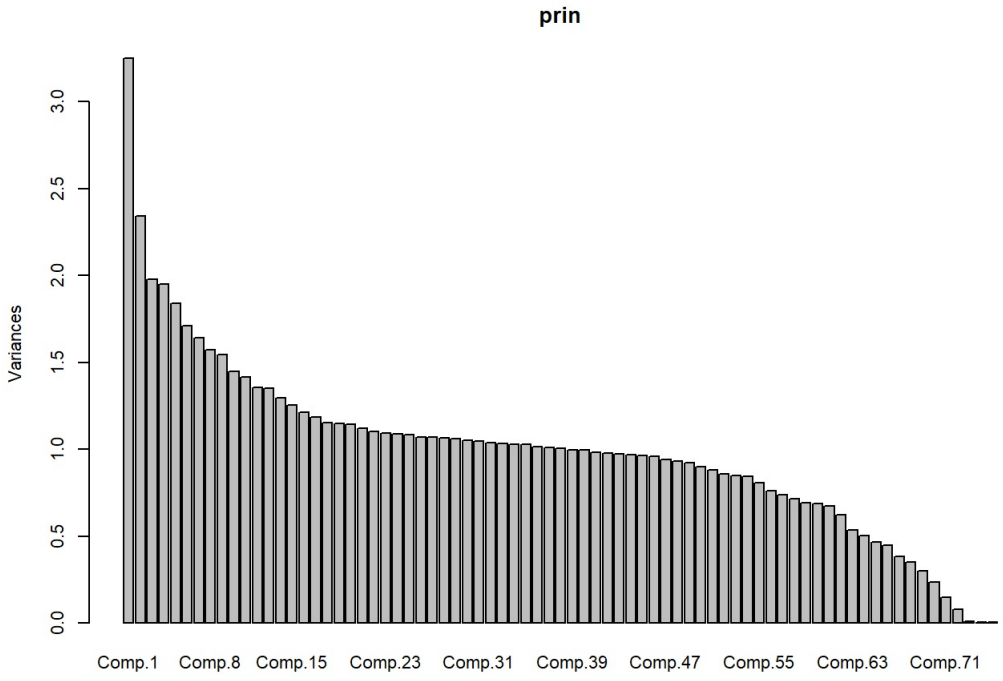
```
## named integer(0)
```

```
# PCA regression
#Standardization of observations in different scales
xxdata=xx
for(i in 1:75){
  xxdata[,i]=(xxdata[,i]-mean(xx[,i]))/sd(xx[,i])
}
prin=princomp(xxdata)

# Loadings
summary(prin)
```

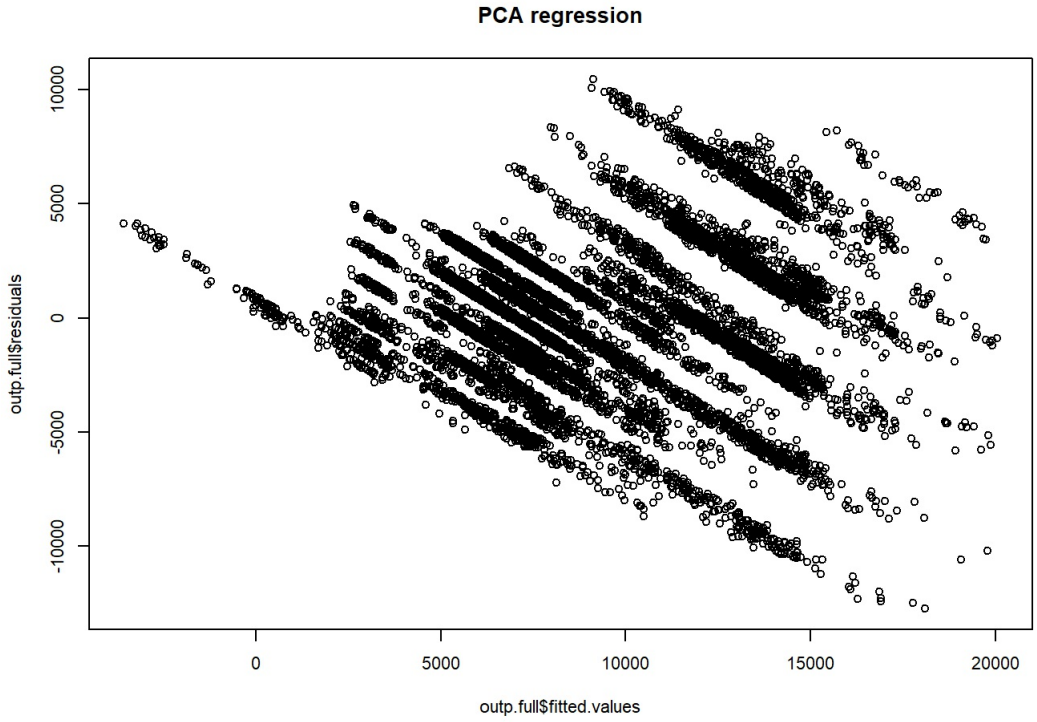
```
## Importance of components:
##                            Comp.1     Comp.2     Comp.3     Comp.4
## Standard deviation     1.80343803 1.53073299 1.40656415 1.39715766
## Proportion of Variance 0.04336952 0.03124504 0.02638161 0.02602993
## Cumulative Proportion  0.04336952 0.07461456 0.10099617 0.12702610
##                            Comp.5     Comp.6    Comp.7     Comp.8
## Standard deviation     1.3570495 1.30887828 1.2814166 1.25478849
## Proportion of Variance 0.0245569 0.02284445 0.0218959 0.02099536
## Cumulative Proportion  0.1515830 0.17442744 0.1963233 0.21731870
##                            Comp.9    Comp.10    Comp.11    Comp.12
```

```
## Standard deviation     1.24302872 1.20445471 1.19000175 1.16570685
## Proportion of Variance 0.02060367 0.01934475 0.01888328 0.01812011
## Cumulative Proportion  0.23792237 0.25726712 0.27615039 0.29427051
##                           Comp.13    Comp.14    Comp.15    Comp.16
## Standard deviation     1.16199140 1.13891490 1.12047798 1.10146624
## Proportion of Variance 0.01800479 0.01729676 0.01674129 0.01617799
## Cumulative Proportion  0.31227529 0.32957205 0.34631334 0.36249133
##                           Comp.17    Comp.18    Comp.19    Comp.20
## Standard deviation     1.08947159 1.07353955 1.07174698 1.07084716
## Proportion of Variance 0.01582756 0.01536803 0.01531675 0.01529104
## Cumulative Proportion  0.37831889 0.39368692 0.40900367 0.42429472
##                           Comp.21    Comp.22    Comp.23    Comp.24
## Standard deviation     1.05827966 1.04933093 1.04512132 1.04300541
## Proportion of Variance 0.01493424 0.01468274 0.01456517 0.01450625
## Cumulative Proportion  0.43922896 0.45391170 0.46847687 0.48298312
##                           Comp.25    Comp.26    Comp.27    Comp.28
## Standard deviation     1.04118378 1.03596543 1.03517934 1.03340714
## Proportion of Variance 0.01445563 0.01431109 0.01428938 0.01424049
## Cumulative Proportion  0.49743875 0.51174984 0.52603922 0.54027971
##                           Comp.29    Comp.30    Comp.31    Comp.32
## Standard deviation     1.0295247 1.02690041 1.02457141 1.01967305
## Proportion of Variance 0.0141337 0.01406173 0.01399802 0.01386449
## Cumulative Proportion  0.5544134 0.56847514 0.58247316 0.59633766
##                           Comp.33    Comp.34    Comp.35    Comp.36
## Standard deviation     1.01760734 1.01546617 1.01352061 1.00842467
## Proportion of Variance 0.01380838 0.01375033 0.01369769 0.01356029
## Cumulative Proportion  0.61014603 0.62389636 0.63759405 0.65115434
##                           Comp.37    Comp.38    Comp.39    Comp.40
## Standard deviation     1.00495450 1.00219902 0.99885006 0.99778699
## Proportion of Variance 0.01346713 0.01339338 0.01330402 0.01327571
## Cumulative Proportion  0.66462147 0.67801485 0.69131887 0.70459458
##                           Comp.41    Comp.42    Comp.43    Comp.44
## Standard deviation     0.99260277 0.98895554 0.98660827 0.98455919
## Proportion of Variance 0.01313812 0.01304174 0.01297991 0.01292605
## Cumulative Proportion  0.71773270 0.73077444 0.74375435 0.75668040
##                           Comp.45    Comp.46    Comp.47    Comp.48
## Standard deviation     0.98256747 0.97943485 0.97166582 0.96665546
## Proportion of Variance 0.01287381 0.01279185 0.01258972 0.01246022
## Cumulative Proportion  0.76955421 0.78234605 0.79493577 0.80739599
##                           Comp.49    Comp.50   Comp.51   Comp.52
## Standard deviation     0.96015963 0.94991592 0.9384865 0.9273775
## Proportion of Variance 0.01229332 0.01203241 0.0117446 0.0114682
## Cumulative Proportion  0.81968930 0.83172171 0.8434663 0.8549345
##                           Comp.53    Comp.54    Comp.55    Comp.56
## Standard deviation     0.92287883 0.91901095 0.89920159 0.87253222
## Proportion of Variance 0.01135721 0.01126221 0.01078192 0.01015185
## Cumulative Proportion  0.86629172 0.87755393 0.88833585 0.89848770
##                           Comp.57     Comp.58     Comp.59     Comp.60
## Standard deviation     0.860631948 0.847078419 0.831830953 0.828865916
## Proportion of Variance 0.009876819 0.009568181 0.009226826 0.009161166
## Cumulative Proportion  0.908364518 0.917932699 0.927159525 0.936320691
##                           Comp.61     Comp.62     Comp.63     Comp.64
## Standard deviation     0.821937140 0.788806650 0.731803225 0.709675536
## Proportion of Variance 0.009008643 0.008297042 0.007141194 0.006715863
## Cumulative Proportion  0.945329334 0.953626376 0.960767570 0.967483433
##                           Comp.65     Comp.66     Comp.67     Comp.68
## Standard deviation     0.682377251 0.671095723 0.620957215 0.591565071
## Proportion of Variance 0.006209137 0.006005527 0.005141686 0.004666456
## Cumulative Proportion  0.973692570 0.979698097 0.984839782 0.989506239
##                           Comp.69     Comp.70     Comp.71     Comp.72
## Standard deviation     0.547011021 0.488247509 0.383283970 0.282322401
## Proportion of Variance 0.003990013 0.003178793 0.001958951 0.001062852
## Cumulative Proportion  0.993496252 0.996675045 0.998633995 0.999696847
##                            Comp.73      Comp.74      Comp.75
## Standard deviation     0.1130887998 7.251066e-02 6.846382e-02
## Proportion of Variance 0.0001705381 7.011096e-05 6.250351e-05
## Cumulative Proportion  0.9998673855 9.999375e-01 1.000000e+00
```

```
# Screeplot
par(cex=0.7)
screeplot(prin,npcs=75,type="barplot")
```




```
# Make transformation on data upon the principle components.
xxdatap=predict(prin)

# Fit regression models with principle components
# whose accumulative variance proportion has just reach 0.9.
outp.full<-lm(y~.,data=as.data.frame(xxdatap[,1:57]))

# Plot residuals against fitted values.
plot(outp.full$fitted.values,outp.full$residuals,main="PCA regression")
```




```
# Make log transformation to y.
fullp=formula(lm(ylog~.,as.data.frame(xxdatap[,1:57])))

# Stepwise backward selection.(use AIC as criterion)
outp.backward=step(outp.full,scope=list(lower=~1,upper=fullp),direction="backward",
                   data=as.data.frame(xxdatap[,1:57]),trace=F)
summary(outp.backward)
```

```
##
```

```
## Call:
## lm(formula = y ~ Comp.1 + Comp.2 + Comp.3 + Comp.4 + Comp.5 +
##     Comp.6 + Comp.7 + Comp.8 + Comp.9 + Comp.10 + Comp.11 + Comp.12 +
##     Comp.14 + Comp.15 + Comp.16 + Comp.17 + Comp.18 + Comp.19 +
##     Comp.20 + Comp.21 + Comp.22 + Comp.23 + Comp.24 + Comp.25 +
##     Comp.26 + Comp.27 + Comp.29 + Comp.30 + Comp.31 + Comp.32 +
##     Comp.34 + Comp.35 + Comp.37 + Comp.38 + Comp.39 + Comp.40 +
##     Comp.41 + Comp.42 + Comp.43 + Comp.44 + Comp.45 + Comp.47 +
##     Comp.49 + Comp.50 + Comp.51 + Comp.52 + Comp.53 + Comp.54 +
##     Comp.55 + Comp.56 + Comp.57, data = as.data.frame(xxdatap[,
##     1:57]))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12700.8  -1915.8    154.6   1976.3  10448.0
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9152.49      32.35 282.885  < 2e-16 ***
## Comp.1       1190.14      17.94  66.339  < 2e-16 ***
## Comp.2       -563.61      21.14 -26.666  < 2e-16 ***
## Comp.3        118.14      23.00   5.136 2.86e-07 ***
## Comp.4       -238.51      23.16 -10.299  < 2e-16 ***
## Comp.5        131.84      23.84   5.530 3.29e-08 ***
## Comp.6        716.93      24.72  29.003  < 2e-16 ***
## Comp.7        119.71      25.25   4.741 2.15e-06 ***
## Comp.8        411.70      25.78  15.967  < 2e-16 ***
## Comp.9       -569.07      26.03 -21.863  < 2e-16 ***
## Comp.10       195.02      26.86   7.260 4.16e-13 ***
## Comp.11        76.33      27.19   2.808 0.005001 **
## Comp.12       224.49      27.75   8.088 6.76e-16 ***
## Comp.14       117.16      28.41   4.124 3.75e-05 ***
## Comp.15       532.79      28.88  18.451  < 2e-16 ***
## Comp.16      -197.41      29.37  -6.721 1.91e-11 ***
## Comp.17      -846.16      29.70 -28.493  < 2e-16 ***
## Comp.18       569.75      30.14  18.905  < 2e-16 ***
## Comp.19      -427.04      30.19 -14.146  < 2e-16 ***
## Comp.20       -47.77      30.21  -1.581 0.113919
## Comp.21       248.33      30.57   8.123 5.10e-16 ***
## Comp.22      -164.16      30.83  -5.324 1.04e-07 ***
## Comp.23      -422.50      30.96 -13.648  < 2e-16 ***
## Comp.24      -321.94      31.02 -10.379  < 2e-16 ***
## Comp.25       197.90      31.07   6.369 1.99e-10 ***
## Comp.26       201.82      31.23   6.462 1.08e-10 ***
## Comp.27       399.28      31.25  12.775  < 2e-16 ***
## Comp.29       325.35      31.43  10.353  < 2e-16 ***
## Comp.30        80.42      31.51   2.553 0.010707 *
## Comp.31       198.02      31.58   6.271 3.74e-10 ***
## Comp.32       130.37      31.73   4.109 4.01e-05 ***
## Comp.34      -243.34      31.86  -7.637 2.42e-14 ***
## Comp.35      -507.63      31.92 -15.902  < 2e-16 ***
## Comp.37      -153.87      32.19  -4.779 1.78e-06 ***
## Comp.38      -404.80      32.28 -12.539  < 2e-16 ***
## Comp.39       489.19      32.39  15.102  < 2e-16 ***
## Comp.40       429.01      32.43  13.230  < 2e-16 ***
## Comp.41      -220.95      32.60  -6.779 1.28e-11 ***
## Comp.42        97.80      32.72   2.989 0.002803 **
## Comp.43      -392.59      32.79 -11.972  < 2e-16 ***
## Comp.44        65.93      32.86   2.006 0.044850 *
## Comp.45      -143.68      32.93  -4.363 1.29e-05 ***
## Comp.47       221.31      33.30   6.647 3.16e-11 ***
## Comp.49       353.50      33.70  10.491  < 2e-16 ***
## Comp.50      -686.97      34.06 -20.169  < 2e-16 ***
## Comp.51       383.04      34.47  11.111  < 2e-16 ***
## Comp.52      -819.40      34.89 -23.487  < 2e-16 ***
## Comp.53      -702.90      35.06 -20.050  < 2e-16 ***
## Comp.54       509.47      35.21  14.471  < 2e-16 ***
## Comp.55       405.65      35.98  11.274  < 2e-16 ***
## Comp.56      -432.93      37.08 -11.675  < 2e-16 ***
## Comp.57       128.48      37.59   3.418 0.000634 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3235 on 9948 degrees of freedom
## Multiple R-squared:  0.563,  Adjusted R-squared:  0.5607
## F-statistic: 251.3 on 51 and 9948 DF,  p-value: < 2.2e-16
```