

```

1 ##### Econ 690 Computational Methods #####
2
3 ### Assignment 3 ###
4 # 2/7/2020
5 # by Bahar Zafer
6
7 ###=====
8 # Exercise 1 Basic Description
9 ###=====
10
11 install.packages("fastDummies")
12 library(fastDummies)
13 library(ggplot2)
14 library(AER)
15 data("Affairs")
16 dat = Affairs
17
18 # Finding correlations between affair and observed attributes in the data
19
20 # converting gender and children to numeric
21 dat1 = dat
22 dat1 = dummy_cols(dat1, remove_first_dummy = T )
23 # dummy variables, gender_male and children_yes are created.
24 dat1 = dat1 %>% remove_cols("gender", "children")
25
26 cor(dat1)
27
28 # Plotting the histogram of affairs
29 library(dplyr)
30 library(reshape2)
31
32 # by gender
33 g = dat %>%
34   ggplot( aes(x=affairs, fill=gender)) +
35   geom_histogram(color="#e9ecef", alpha=0.6, position = 'dodge') +
36   theme_ipsum() +
37   labs(fill="")
38
39 # by age groups
40 groups_age = vector()
41 for(i in 1:nrow(dat)) {
42   if(dat$age[i]>40) groups_age[i] = "above 40"
43   else groups_age[i] = "under 40"
44 }
45 dat = cbind(dat, groups_age)
46
47 a = dat %>%
48   ggplot( aes(x=affairs, fill=groups_age)) +
49   geom_histogram(color="#e9ecef", alpha=0.6, position = 'dodge') +
50   theme_ipsum() +
51   labs(fill="")
52
53
54 # by religiousness
55 groups_relig = vector()
56 for(i in 1:nrow(dat)) {
57   if(dat$religiousness[i]>3)
58     groups_relig[i] = "religious"
59   else groups_relig[i] = "not religious"
60 }
61 dat = cbind(dat, groups_relig)
62
63 r = dat %>%
64   ggplot( aes(x=affairs, fill=groups_relig)) +
65   geom_histogram(alpha=0.6, position = 'dodge') +
66   theme_ipsum() +
67   labs(fill="")
68
69

```

```

70  ###=====
71  # Exercise 2 Linear Regression
72  ###=====
73  regression1 = lm(affairs ~ education + gender_male + occupation + age + children_yes +
74                  religiousness + yearsmarried + rating ,
75                  data = dat1)
76
77
78  regression2 = lm(affairs ~ education*occupation +
79                  education + gender_male + occupation + age + children_yes +
80                  religiousness + yearsmarried + rating ,
81                  data = dat1)
82
83
84  regression3 = lm(affairs ~ education*children_yes +
85                  education + gender_male + occupation + age + children_yes +
86                  religiousness + yearsmarried + rating ,
87                  data = dat1)
88
89
90  ###=====
91  # Exercise 3 Probit
92  ###=====
93
94  install.packages("NORMT3")
95  library(NORMT3)
96
97  # recoding affairs as a dummy variable
98  dat1$affairs = as.numeric(dat1$affairs>0)
99
100 # creating a matrix of dependent and independent variables
101 xmat = cbind(1, dat1[-dat1[, "affairs"]])
102 xmat = as.matrix(Xmat)
103
104 ymat = as.matrix(dat1$affairs)
105
106 # Estimating the model using glm()
107 probitaffairs_glm = glm(formula = affairs ~ education + gender_male +
108                          occupation + age + children_yes +
109                          religiousness + yearsmarried + rating ,
110                          family = binomial(link = "probit"),
111                          data = dat1)
112
113 # extracting coefficients
114 betas = as.matrix(coef(probitaffairs_glm))
115
116 # likelihood function
117
118 affairs_probit = function(b, x, y){
119     L = vector()
120     for (i in 1:nrow(y)){
121         L[i] = erf(x[i,]%*%b)^y[i,]*
122               (1-erf(x[i,]%*%b))^(1-y[i,])
123     }
124     return(L)
125 }
126
127 # vector of likelihood for each observation
128 vec_L_i = affairs_probit(betas, xmat, ymat)
129
130 # likelihood of all sample
131 likelihood = prod(vec_L_i)
132
133
134  ###=====
135  # Exercise 4 Logit
136  ###=====
137  CDF_logit = function(xx, bb) {
138      exp(xx%*%bb)/(1 + exp(xx%*%bb))

```

```

139 }
140
141 affairs_logit = function(b, x, y){
142     L = vector()
143     for (i in 1:nrow(y)) {
144         L[i] = CDF_logit(x[i,], b)^y[i,]*
145             (1-CDF_logit(x[i,], b))^(1-y[i,])
146     }
147     return(L)
148 }
149
150 # vector of likelihood for each observation in affairs
151 vec_logit_L = affairs_probit(betas, xmat, ymat)
152
153 # likelihood of all sample
154 MLE_logit = prod(vec_logit_L)
155
156
157 ###=====
158 # Exercise 5 Multinomial Logit
159 ###=====
160
161 # Conditional Logit Model
162 p_cond_logit = function(xx, bb) {
163     sums = vector()
164     for(i in ncol(xx)) {
165         sums[i] = xx[,i]*bb
166     }
167     return(exp(xx*bb)/sum(sums))
168 }
169
170 affairs_cond_logit = function(b, x, y){
171     L = vector()
172     for (j in 1:ncol(x)) {
173
174         for (i in 1:nrow(y)) {
175             L[i] = p_cond_logit(x[i,j], b)^y[i,]*
176                 (1-p_cond_logit(x[i,j], b))^(1-y[i,])
177         }
178     }
179     return(L)
180 }
181
182 vec_cond_logit_L = affairs_cond_logit(betas, xmat, ymat)
183
184 MLE_cond_logit = prod(vec_cond_logit_L)
185
186 # Multinomial Logit Model
187 p_mult_logit = function(xx, bb) {
188     sums = vector()
189     for(i in ncol(xx)) {
190         sums[i] = xx*bb[i]
191     }
192     return(exp(xx*bb)/sum(sums))
193 }
194
195 affairs_mult_logit = function(b, x, y){
196     L = vector()
197     for (j in 1:nrow(b)) {
198         for (i in 1:nrow(y)) {
199             L[i] = p_cond_logit(x[i,], b[j])^y[i,]*
200                 (1-p_cond_logit(x[i,], b[j]))^(1-y[i,])
201         }
202     }
203     return(L)
204 }
205
206 vec_mult_logit_L = affairs_mult_logit(betas, xmat, ymat)
207

```

```
208 MLE_mult_logit = prod(vec_mult_logit_L)
209
210 # Mixed logit model
211
212
```