HW2

Erik Andersen

2023-10-13

Question 1

```r
# Load packages
pacman::p_load(tidyverse, haven, here, fixest, magrittr, margins, glmx)

# load data
df = read_dta(here("data", "Econ587_Field2010_data.dta"))

# Subset the data to only the rows that have values for our variable of interest
field_df = df |> filter(!is.na(HH_Income))
```

```r
# Estimate linear prob model
non_robust = field_df %>% feols(taken_new ~ Treated + Client_Age + Client_Married + Client_Education + H

summary(non_robust)
```

a)

```
## OLS estimation, Dep. Var.: taken_new
## Observations: 561
## Standard-errors: IID
##                       Estimate Std. Error   t value Pr(>|t|)
## (Intercept)         0.15532701 0.09641593  1.611010  0.10775
## Treated             0.04509724 0.03369204  1.338513  0.18128
## Client_Age          0.00055941 0.00188300  0.297088  0.76651
## Client_Married      0.03159795 0.04816394  0.656050  0.51207
## Client_Education   -0.00411242 0.00383822 -1.071441  0.28444
## HH_Size            -0.01055053 0.00911700 -1.157237  0.24768
## HH_Income           0.00000406 0.00000362  1.120184  0.26312
## muslim             -0.01635700 0.03577487 -0.457220  0.64769
## Hindu_SC_Kat       -0.02777064 0.05051246 -0.549778  0.58269
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## RMSE: 0.371582   Adj. R2: -0.004244
```

```r
# Reestimate but with robust se's
robust = field_df %>% feols(taken_new ~ Treated + Client_Age + Client_Married + Client_Education + HH_S

summary(robust)
```

b)

```
## OLS estimation, Dep. Var.: taken_new
## Observations: 561
## Standard-errors: Heteroskedasticity-robust
##                      Estimate Std. Error   t value Pr(>|t|)
## (Intercept)        0.15532701 0.08814458  1.762185 0.078592 .
## Treated            0.04509724 0.03240097  1.391849 0.164529
## Client_Age         0.00055941 0.00183202  0.305355 0.760211
## Client_Married     0.03159795 0.04591700  0.688154 0.491645
## Client_Education -0.00411242 0.00378242 -1.087247 0.277402
## HH_Size          -0.01055053 0.00907182 -1.163000 0.245332
## HH_Income         0.00000406 0.00000371  1.092368 0.275148
## muslim           -0.01635700 0.03539856 -0.462081 0.644205
## Hindu_SC_Kat     -0.02777064 0.04924740 -0.563901 0.573051
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## RMSE: 0.371582   Adj. R2: -0.004244
```

```r
non_robust$se > robust$se # Smaller SE's for all but HH_income
```

```
##      (Intercept)          Treated       Client_Age   Client_Married
##             TRUE             TRUE             TRUE             TRUE
## Client_Education          HH_Size        HH_Income           muslim
##             TRUE             TRUE            FALSE             TRUE
##     Hindu_SC_Kat
##             TRUE
```

```r
# Generate fitted values from both regressions
fitted_nonrobust = non_robust$fitted.values
fitted_robust = robust$fitted.values

# Check they're identical
sum(fitted_nonrobust == fitted_robust) - length(fitted_nonrobust == fitted_robust) # Roundabout way to
```

c)

```
## [1] 0
```

```r
# Find range of fitted values
max(fitted_robust); min(fitted_robust) # None outside of 0,1
```

```
## [1] 0.3207797

## [1] 0.03529123
```

```r
# the weights are just our residuals. Calculate again for robust and non-robust
# Define weights
non_weights = 1 / lm(abs(non_robust$residuals) ~ non_robust$fitted.values)$fitted.values^2
weights = 1 / lm(abs(robust$residuals) ~ robust$fitted.values)$fitted.values^2


non_robust_weighted = field_df %>% feols(taken_new ~ Treated + Client_Age + Client_Married + Client_Edu
summary(non_robust_weighted)
```

d)

```
## OLS estimation, Dep. Var.: taken_new
## Observations: 561
## Weights: non_weights
## Standard-errors: IID
##                      Estimate Std. Error   t value Pr(>|t|)
## (Intercept)       0.11319834 0.08380829  1.350682  0.17735
## Treated           0.04473244 0.03042988  1.470017  0.14213
## Client_Age        0.00146514 0.00173492  0.844496  0.39876
## Client_Married    0.02267607 0.04054471  0.559285  0.57619
## Client_Education -0.00238934 0.00346536 -0.689491  0.49080
## HH_Size          -0.01223611 0.00777249 -1.574284  0.11599
## HH_Income         0.00000667 0.00000386  1.727262  0.08468 .
## muslim           -0.01897740 0.03384038 -0.560791  0.57517
## Hindu_SC_Kat     -0.00712222 0.04518604 -0.157620  0.87481
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## RMSE: 1.36888   Adj. R2: 0.002701
```

```r
robust_weighted = field_df %>% feols(taken_new ~ Treated + Client_Age + Client_Married + Client_Educati
summary(robust_weighted)
```

```
## OLS estimation, Dep. Var.: taken_new
## Observations: 561
## Weights: weights
## Standard-errors: Heteroskedasticity-robust
##                      Estimate Std. Error   t value Pr(>|t|)
## (Intercept)       0.11319834 0.07308932  1.548767  0.12201
## Treated           0.04473244 0.03263838  1.370547  0.17107
## Client_Age        0.00146514 0.00178386  0.821331  0.41181
## Client_Married    0.02267607 0.04466980  0.507638  0.61191
## Client_Education -0.00238934 0.00318533 -0.750108  0.45351
## HH_Size          -0.01223611 0.00777922 -1.572922  0.11631
## HH_Income         0.00000667 0.00000415  1.605918  0.10886
## muslim           -0.01897740 0.03391415 -0.559572  0.57600
## Hindu_SC_Kat     -0.00712222 0.05267140 -0.135220  0.89249
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## RMSE: 1.36888   Adj. R2: 0.002701
```

```r
# Add interaction between age and Muslim.
interact = field_df %>% feols(taken_new ~ Treated + Client_Age + Client_Married + Client_Education + HH_

summary(interact)
```

f)

```
## OLS estimation, Dep. Var.: taken_new
## Observations: 561
## Standard-errors: IID
##                       Estimate Std. Error   t value Pr(>|t|)
## (Intercept)        0.20497477 0.10718056  1.912425  0.05634 .
## Treated            0.04475369 0.03368982  1.328404  0.18459
## Client_Age        -0.00081737 0.00228733 -0.357347  0.72097
## Client_Married     0.02521987 0.04853299  0.519644  0.60352
## Client_Education  -0.00399431 0.00383940 -1.040347  0.29864
## HH_Size           -0.01028644 0.00911938 -1.127976  0.25982
## HH_Income          0.00000433 0.00000363  1.192576  0.23355
## muslim            -0.15912698 0.13935685 -1.141867  0.25401
## Hindu_SC_Kat      -0.02574813 0.05054283 -0.509432  0.61065
## Client_Age:muslim  0.00417254 0.00393633  1.060008  0.28961
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## RMSE: 0.371204   Adj. R2: -0.004019
```

```r
# Calculate average partial effect of age
(sum(field_df$muslim)/nrow(field_df))*(coefficients(interact)[3]+coefficients(interact)[10])+(1-sum(fie
```

```
##   Client_Age
## 0.0004247225
```

Question 2)

```r
# Same regression, but with logit model
logit = df %>% glm(taken_new ~ Treated + Client_Age + Client_Married + Client_Education + HH_Size + HH_
summary(logit)
```

a)

```
##
## Call:
## glm(formula = taken_new ~ Treated + Client_Age + Client_Married +
##     Client_Education + HH_Size + HH_Income + muslim + Hindu_SC_Kat,
##     family = binomial(link = "logit"), data = .)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9566  -0.6352  -0.5808  -0.5015   2.1556
```

```
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.714e+00  7.234e-01  -2.369   0.0178 *
## Treated           3.392e-01  2.523e-01   1.345   0.1787
## Client_Age        4.059e-03  1.382e-02   0.294   0.7690
## Client_Married    2.545e-01  3.666e-01   0.694   0.4875
## Client_Education -3.067e-02  2.808e-02  -1.092   0.2748
## HH_Size          -8.008e-02  6.797e-02  -1.178   0.2387
## HH_Income         2.809e-05  2.465e-05   1.139   0.2546
## muslim           -1.196e-01  2.597e-01  -0.461   0.6450
## Hindu_SC_Kat     -2.109e-01  3.756e-01  -0.561   0.5745
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 507.14  on 560  degrees of freedom
## Residual deviance: 501.34  on 552  degrees of freedom
##   (36 observations deleted due to missingness)
## AIC: 519.34
##
## Number of Fisher Scoring iterations: 4
```

```r
# now probit
probit = df %>% glm(taken_new ~ Treated + Client_Age + Client_Married + Client_Education + HH_Size + HH_
summary(probit)
```

```
##
## Call:
## glm(formula = taken_new ~ Treated + Client_Age + Client_Married +
##     Client_Education + HH_Size + HH_Income + muslim + Hindu_SC_Kat,
##     family = binomial(link = "probit"), data = .)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9586  -0.6364  -0.5813  -0.5014   2.1646
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.043e+00  3.971e-01  -2.627  0.00862 **
## Treated           1.881e-01  1.382e-01   1.362  0.17332
## Client_Age        2.679e-03  7.638e-03   0.351  0.72580
## Client_Married    1.390e-01  1.998e-01   0.695  0.48681
## Client_Education -1.624e-02  1.554e-02  -1.045  0.29616
## HH_Size          -4.512e-02  3.739e-02  -1.207  0.22752
## HH_Income         1.659e-05  1.401e-05   1.184  0.23623
## muslim           -6.811e-02  1.440e-01  -0.473  0.63625
## Hindu_SC_Kat     -1.118e-01  2.061e-01  -0.543  0.58738
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 507.14  on 560  degrees of freedom
## Residual deviance: 501.27  on 552  degrees of freedom
##   (36 observations deleted due to missingness)
## AIC: 519.27
##
## Number of Fisher Scoring iterations: 4
```

```r
# We already have the predicted values stored as the fitted values in the two regression objects
logit_predict = logit$fitted.values
probit_predict = probit$fitted.values
lmp_predict = robust$fitted.values

# Get the correlation
cor(logit_predict, probit_predict) # Almost exactly perfectly correlated
```

b)

```
## [1] 0.9988347
```

```r
cor(logit_predict, lmp_predict)
```

```
## [1] 0.9915219
```

```r
cor(probit_predict, lmp_predict)
```

```
## [1] 0.9932232
```

```r
# calculate mean of ages
xbar = mean(df$Client_Age)
# Calculate beta for age
beta = coefficients(probit)[3]

# Define function for normal pdf for late use
g = function(x){
  1/(2*pi)*exp(-1/2*x^2)
}

# Plug into pdf for pdf for normal distribution
(partial = g(xbar*beta)*beta)
```

c)

```
##  Client_Age
## 0.000424746
```

```r
# We'll use the margins package which has similar functionality to stata's margins command
margins(probit, data = df, variables = c("Client_Age"))
```

d)

```
## Average marginal effects

## glm(formula = taken_new ~ Treated + Client_Age + Client_Married +     Client_Education + HH_Size + HH

##  Client_Age
##   0.0006637
```

```r
# Calculate partial effects
partial_means = sapply(1:nrow(df), function(i){
  # Plug into pdf
  out = g(df$Client_Age[i])%*%beta
})

# the row means of the above object are the mean of partial effects
(means = mean(partial_means))
```

e)

```
## [1] 2.999457e-05
```

```r
numerical_means = sapply(1:nrow(df), function(i){
  # Calculate numerical derivative for each observation
  (dnorm((df$Client_Age[i]-.001)*beta) - dnorm(df$Client_Age[i]*beta))/.001
})

mean(numerical_means)
```

f)

```
## [1] 9.236215e-05
```

Question 3

```r
# Get fitted values from lmp
fitted_lpm = robust$fitted.values
```

```r
# round to 0/1 based on 0.5 cutoff
predictions = round(fitted_lpm)

# To compare what is correct, we need to see where the prediction and the true outcome match. To do tha
vars_df = df |> select(taken_new, Treated, Client_Age, Client_Married, Client_Education, HH_Size, HH_Inc
complete = complete.cases(vars_df)
vars_df = vars_df[complete,]

# now we can see when the prediction was correct
sum(vars_df$taken_new == predictions)/length(predictions) # We predict no one accepts the lone, so this
```

a)

```
## [1] 0.8324421
```

```r
# Now we change the cutoff value to the mean of the loans ~0.16
predictions = fitted_lpm |> as_tibble() |>
  mutate(out = if_else(fitted_lpm<mean(vars_df$taken_new),0,1))

# Compare to true values
sum(vars_df$taken_new == predictions$out)/length(predictions$out) # 0.52
```

```
## [1] 0.5187166
```

```r
# Get fitted values from probit
fitted_probit = probit$fitted.values

# Round baed on 0.5 cutoff
predictions = round(fitted_probit)

# Compute prediction percentage
sum(vars_df$taken_new == predictions)/length(predictions) # Again we predict no one takes up loan
```

b)

```
## [1] 0.8324421
```

```r
# Round based on new cutoff
predictions = fitted_probit |> as_tibble() |>
  mutate(out = if_else(fitted_probit<mean(vars_df$taken_new),0,1))

# Compare to true values
sum(vars_df$taken_new == predictions$out)/length(predictions$out) # ~0.54
```

```
## [1] 0.543672
```

```r
# Filter such that imidlineid < 1400
df_1 = df |>  filter(imidlineid < 1400)

# Do the same thing to get only complete cases
vars_df1 = df_1 |> select(taken_new, Treated, Client_Age, Client_Married, Client_Education, HH_Size, HH_
complete = complete.cases(vars_df1)
vars_df1 = vars_df1[complete,]

# Rerun probit
probit_reduced = df_1 %>% glm(taken_new ~ Treated + Client_Age + Client_Married + Client_Education + HH_
summary(probit_reduced)
```

c)

```
##
## Call:
## glm(formula = taken_new ~ Treated + Client_Age + Client_Married +
##     Client_Education + HH_Size + HH_Income + muslim + Hindu_SC_Kat,
##     family = binomial(link = "probit"), data = .)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8333  -0.6404  -0.5791  -0.4193   2.3437
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.840e+00  6.759e-01  -2.723  0.00647 **
## Treated           4.111e-02  1.959e-01   0.210  0.83381
## Client_Age        1.083e-02  1.243e-02   0.871  0.38361
## Client_Married    2.327e-01  3.011e-01   0.773  0.43976
## Client_Education -3.265e-03  2.337e-02  -0.140  0.88891
## HH_Size           5.304e-02  5.107e-02   1.039  0.29900
## HH_Income         6.295e-06  1.969e-05   0.320  0.74914
## muslim           -4.495e-02  2.083e-01  -0.216  0.82917
## Hindu_SC_Kat     -5.572e-01  3.754e-01  -1.484  0.13771
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 250.16  on 279  degrees of freedom
## Residual deviance: 244.28  on 271  degrees of freedom
##   (2 observations deleted due to missingness)
## AIC: 262.28
##
## Number of Fisher Scoring iterations: 5
```

```r
# round output for 0.5
predictions = round(probit_reduced$fitted.values)

# Compute prediction percentage
sum(vars_df1$taken_new == predictions)/length(predictions)
```

```
## [1] 0.8357143
```

```r
# Round for cutoff at mean
predictions = probit_reduced$fitted.values |> as_tibble() |>
  mutate(out = if_else(probit_reduced$fitted.values<mean(vars_df$taken_new),0,1))

# Compute percentage correct
sum(vars_df1$taken_new == predictions$out)/length(predictions$out) # ~0.54
```

```
## [1] 0.5464286
```

Question 4)

```r
# Reestimate the lmp model
lmp_reg = vars_df %>% feols(taken_new ~ Treated + Client_Age + Client_Married + Client_Education + HH_Si

# Add residuals to the dataset
vars_df = vars_df |> mutate(lmp_resid = lmp_reg$residuals)

# Regress the residuals on the covariates from the original regression
resid_reg = vars_df %>% lm(lmp_resid ~ Treated + Client_Age + Client_Married + Client_Education + HH_Si
summary(resid_reg)
```

a)

```
##
## Call:
## lm(formula = lmp_resid ~ Treated + Client_Age + Client_Married +
##     Client_Education + HH_Size + HH_Income + muslim + Hindu_SC_Kat,
##     data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.3208 -0.1848 -0.1586 -0.1171  0.9124
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.393e-15  9.642e-02       0        1
## Treated          -1.753e-16  3.369e-02       0        1
## Client_Age        6.054e-19  1.883e-03       0        1
## Client_Married   -4.532e-16  4.816e-02       0        1
## Client_Education  9.204e-18  3.838e-03       0        1
## HH_Size           4.918e-17  9.117e-03       0        1
## HH_Income        -2.756e-20  3.623e-06       0        1
## muslim           -3.686e-17  3.577e-02       0        1
## Hindu_SC_Kat     -1.534e-16  5.051e-02       0        1
##
## Residual standard error: 0.3746 on 552 degrees of freedom
## Multiple R-squared:  4.167e-31,  Adjusted R-squared:  -0.01449
## F-statistic: 2.875e-29 on 8 and 552 DF,  p-value: 1
```

```r
# Now the same thing for squared residuals
vars_df = vars_df |> mutate(lmp_residsq = lmp_resid^2)

resid_regsq = vars_df %>% lm(lmp_residsq ~ Treated + Client_Age + Client_Married + Client_Education + HH
summary(resid_regsq)
```

b)

```
##
## Call:
## lm(formula = lmp_residsq ~ Treated + Client_Age + Client_Married +
##     Client_Education + HH_Size + HH_Income + muslim + Hindu_SC_Kat,
##     data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.15318 -0.11658 -0.10644 -0.08923  0.74442
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.208e-01  6.311e-02   1.913   0.0562 .
## Treated           3.050e-02  2.205e-02   1.383   0.1673
## Client_Age        5.785e-04  1.233e-03   0.469   0.6390
## Client_Married    2.210e-02  3.153e-02   0.701   0.4835
## Client_Education -2.371e-03  2.512e-03  -0.944   0.3456
## HH_Size          -7.708e-03  5.968e-03  -1.292   0.1970
## HH_Income         3.072e-06  2.371e-06   1.295   0.1957
## muslim           -1.160e-02  2.342e-02  -0.496   0.6204
## Hindu_SC_Kat     -1.699e-02  3.306e-02  -0.514   0.6075
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2452 on 552 degrees of freedom
## Multiple R-squared:  0.01144,    Adjusted R-squared:  -0.002883
## F-statistic: 0.7987 on 8 and 552 DF,  p-value: 0.6039
```

```r
# Now fit a heteroskadastic probit model using glmx package
hetprobit = df %>% hetglm(taken_new ~ Treated + Client_Age + Client_Married + Client_Education + HH_Size
summary(hetprobit)
```

c)

```
##
## Call:
## hetglm(formula = taken_new ~ Treated + Client_Age + Client_Married +
##     Client_Education + HH_Size + HH_Income + muslim + Hindu_SC_Kat, data = .,
##     family = binomial(link = "probit"))
##
```

```
## Deviance residuals:
##     Min      1Q  Median      3Q     Max
## -1.1194 -0.6666 -0.5559 -0.4051  2.2968
##
## Coefficients (binomial model with probit link):
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -0.9692947  0.3976941  -2.437   0.0148 *
## Treated          -0.1447791  0.2599118  -0.557   0.5775
## Client_Age        0.0263268  0.0154478   1.704   0.0883 .
## Client_Married   -0.1528424  0.2910579  -0.525   0.5995
## Client_Education -0.0031370  0.0241726  -0.130   0.8967
## HH_Size          -0.0878543  0.1052161  -0.835   0.4037
## HH_Income        -0.0001253  0.0001023  -1.225   0.2207
## muslim           -0.0414596  0.3536557  -0.117   0.9067
## Hindu_SC_Kat      0.6502275  0.4466820   1.456   0.1455
##
## Latent scale model coefficients (with log link):
##                   Estimate Std. Error z value Pr(>|z|)
## Treated           4.108e-01  2.430e-01   1.691  0.09086 .
## Client_Age       -2.931e-02  1.009e-02  -2.906  0.00366 **
## Client_Married    5.140e-01  3.035e-01   1.694  0.09035 .
## Client_Education -1.202e-02  2.689e-02  -0.447  0.65495
## HH_Size           4.807e-04  6.627e-02   0.007  0.99421
## HH_Income         1.343e-04  3.424e-05   3.922 8.78e-05 ***
## muslim           -5.694e-02  3.081e-01  -0.185  0.85337
## Hindu_SC_Kat     -9.353e-01  3.641e-01  -2.569  0.01020 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-likelihood: -246.5 on 17 Df
## LR test for homoscedasticity: 8.361 on 8 Df, p-value: 0.3991
## Dispersion: 1
## Number of iterations in nlminb optimization: 34
```