Problem Set 3

Tessie Dong, Derek Li, Andi Liu

Due Jan 18th, 2024

Part 1: Describe the Data

Combine the NSW data in nswre74_control.csv and nswre74_treated.csv and complete Table 1. Note that variables 1-10 are *predetermined*, i.e., capture characteristics determined at or before treatment assignment; some of these variables are background characteristics (e.g., edu), others capture a subject's pre-RCT labor market experience (e.g., u75). re78 is the observed outcome variable. treat is the indicator of treatment status.

```
# load and combine the data sets into one dataframe
df1 <- utils::read.csv(file = "nswre74_control.csv")
df2 <- utils::read.csv(file = "nswre74_treated.csv")
df <- rbind(df1, df2)
```

```
# count units in each sample
dplyr::tally(dplyr::group_by(df, treat))
```

```
## # A tibble: 2 x 2
##   treat     n
##   <int> <int>
## 1     0   260
## 2     1   185
```

```
# generate mean summary statistics for each variable and treatment
dplyr::summarise_all((dplyr::group_by(df, treat)), list(mean))
```

```
## # A tibble: 2 x 12
##   treat   age   edu black   hisp married nodegree  re74  re75  re78   u74   u75
##   <int> <dbl> <dbl> <dbl>  <dbl>   <dbl>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1     0  25.1  10.1 0.827 0.108    0.154    0.835 2107. 1267. 4555. 0.75  0.685
## 2     1  25.8  10.3 0.843 0.0595   0.189    0.708 2096. 1532. 6349. 0.708 0.6
```

A completed version of Table 1 is provided on the following page.

Variable	Variable	Variable	Sample	Average
Counter	Name	Definition	Treated	Control
1	age	Age in years	25.8	25.1
2	edu	Education in years	10.3	10.1
3	nodegree	1 if education < 12	0.708	0.835
4	black	1 if Black	0.843	0.827
5	hisp	1 if Hispanic	0.0595	0.108
6	married	1 if married	0.189	0.154
7	u74	1 if unemployed in '74	0.708	0.75
8	u75	1 if unemployed in '75	0.6	0.685
9	re74	Real earnings in '74 (in '82 $)	2096	2107
10	re75	Real earnings in '75 (in '82 $)	1532	1267
11	re78	Real earnings in '78 (in '82 $)	6349	4555
12	treat	1 if received offer of training	1	0
Sample Size			185	260

Table 1: Descriptive statistics for the NSW data by group.

Part 2: Test Balance

1. Test balance for each of the 10 OPVs in Table (1), i.e., test that each variable's mean is the same in the control and treated groups. Do so by running 10 simple linear regressions (SLR) specifications. Use a 5% significance level and look at the relevant 10 t-tests, comment on your findings. **Hint**: Each OPV is the dependent variable in its regression equation. All 10 SLR models have the same covariates.

```r
ols_p_values <- list()
ols_coefficients <- list()
ols_se <- list()
vars <- names(df)[2:12]
for (var in vars) {
    formula <- stats::formula(paste(var, "~treat"))
    lm_model <- lm(formula = formula, data = df)
    ols_p_values[[var]] <- summary(lm_model)$coefficients[2, 4]
    ols_coefficients[[var]] <- summary(lm_model)$coefficients[2, 1]
    ols_se[[var]] <- summary(lm_model)$coefficients[2, 2]
}
# print(ols_p_values)
```

Table 2 shows the p-values for the t-tests of balance for each OPV.
We can see that for most of our variables, the p-value is greater than 0.05, so we fail to reject the null hypothesis that the means are the same in the control and treated groups. However, for nodegree we reject the null hypothesis that the means are the same in the control and treated groups, and we cannot claim that these groups are balanced for this variable.

	Variable	p-value
1	age	0.264764
2	edu	0.135411
3	nodegree	0.001398
4	black	0.649493
5	hisp	0.076474
6	married	0.327408
7	u74	0.326209
8	u75	0.065469
9	re74	0.982318
10	re75	0.382254

Table 2: p-values for the t-tests of balance for each OPV.

2. Testing balance as done in Q1 suffers from the so called “multiple comparisons” or “multiple testing” problem which occurs when one considers a set of statistical inferences simultaneously. The problem emerges because as more variables are compared, it becomes more likely that the treatment and control groups appear to differ on at least one attribute *by random chance alone*. To deal with this problem we use an estimation methodology called SUR estimation and then test just one hypothesis, the *joint* hypothesis that all OPVs are balanced, i.e., their means are the same in the two groups. SUR stands for seemingly unrelated regression and it is a special case of feasible Generalized Least Squares (GLS) estimation. Instead of estimating the coefficients *equation-by-equation* by OLS (and done in Q1 you combine the 10 equations in a system of equations and estimate the coefficients present in all the equations jointly, accounting for the fact that the unobservables may be correlated across equations within an individual (we continue to assume that they are uncorrelated across units). After estimation, you use standard testing procedures to test the *joint* hypothesis that the slope coefficients in all the equations of the SUR system are zero. This joint test is a test of covariate balance that does not suffer from the “multiple testing’ ’ problem.

a. Estimate the SUR system. Are the estimated coefficients and their SEs different from those obtained in Q1? Comment. **Hint**: In 2 situations there is no efficiency payoff to GLS versus OLS: 1) the unobservables are uncorrelated across equations within an individual; and 2) the equations have identical covariates.

```r
# create a list of formulas for each equation
formulas <- list()
vars <- names(df)[2:12]
for (var in vars) {
  formulas[[var]] <- formula(paste(var, "~treat"))
}
```

```r
# estimate the SUR model
sur_fit <- systemfit::systemfit(formula = formulas, data = df, method = "SUR")
```

```r
# print the coefficients and SE of the SUR model
i <- 1
while (i <= 11){
  print(sprintf("%s: %f, %f",
                vars[i],
                summary(sur_fit)$coefficients[i * 2, 1],
                summary(sur_fit)$coefficients[i * 2, 2]))
  i <- (i + 1)
}
```

```
## [1] "age: 0.762370, 0.682751"
## [1] "edu: 0.257484, 0.172135"
## [1] "black: 0.016320, 0.035886"
## [1] "hisp: -0.048233, 0.027163"
## [1] "married: 0.035343, 0.036048"
## [1] "nodegree: -0.126507, 0.039345"
## [1] "re74: -11.452958, 516.477971"
## [1] "re75: 265.146299, 303.155497"
## [1] "re78: 1794.342382, 632.853392"
## [1] "u74: -0.041892, 0.042622"
## [1] "u75: -0.084615, 0.045822"
```

```
# print the coefficients and SE of the OLS model
for (var in vars)
{
  print(sprintf("%s: %f, %f ", var, ols_coefficients[var], ols_se[var]))
}
```

```
## [1] "age: 0.762370, 0.682751 "
## [1] "edu: 0.257484, 0.172135 "
## [1] "black: 0.016320, 0.035886 "
## [1] "hisp: -0.048233, 0.027163 "
## [1] "married: 0.035343, 0.036048 "
## [1] "nodegree: -0.126507, 0.039345 "
## [1] "re74: -11.452958, 516.477971 "
## [1] "re75: 265.146299, 303.155497 "
## [1] "re78: 1794.342382, 632.853392 "
## [1] "u74: -0.041892, 0.042622 "
## [1] "u75: -0.084615, 0.045822 "
```

b. Test *joint* balance by testing the joint hypothesis that the coefficients of `treat` are zero in all the equations of the system. Spell out null and alternative hypotheses, comment on findings, and verify the test's value and p-value “manually.’ ’ **Hint**: Use the Likelihood Ratio (LR) test where the unrestricted model is the system of equations with `treat` as covariate, and the restricted model is the system with no regression covariates, i.e., with only the constant term. To verify the value of the test use its algebraic expression. Recall that the test has a χ^2_{df} distribution with df = number of restrictions.

```
# create a list of formulas null_system with only the constant
null_formulas <- list()
for (var in vars) {
  null_formulas[[var]] <- formula(paste(var, "~1"))
}

# pass the null_system to systemfit
null_fit <- systemfit::systemfit(formula = null_formulas, data = df, method = "SUR")

# calculate the LR test statistic
lrtest_obj <- lmtest::lrtest(null_fit, sur_fit)
lr_statistic <- lrtest_obj$Chisq[2]
lr_df <- lrtest_obj$Df[2]

p_value <- stats::pchisq(lr_statistic, df = lr_df, lower.tail = FALSE)
print(sprintf("LR test statistic: %f, p-value: %f", lr_statistic, p_value))
```

```
## [1] "LR test statistic: 27.021898, p-value: 0.004560"
```

3. Test that the OPVs do not predict treatment assignment. Spell out null and alternative hypotheses, comment on findings, and verify the test's value and p-value “manually.’ ’ Why would scientists carry out this test? **Hint:** Use a MLRM and the F-test for the overall significance of the regression. **Programming guidance**: Use `summary()` after estimation, e.g., `summary(lm_fit)$fstatistic` returns the test's value and degrees of freedom. Use `stats::pf()` to verify the test's p-value.

   We want to test that, given the covariates, the treatment assignment is random. We can do this by testing that the coefficients of the covariates are zero in the regression of treatment assignment on the covariates.

   Given (Y, X^{OPV}) where Y represents the treatment assignment, and X is a vector of our OPV covariates, we can test the following model: $Y = \beta' X^{OPV} + \epsilon$ with these hypotheses:

$$\begin{cases} H_0 : \beta = 0 \\ H_1 : \beta \neq 0 \end{cases}$$

   In which β is a vector of coefficients for the OPV covariates. Thus, we want $\beta = 0$, i.e the OPV covariates all have no correlation to the treatment Y.

```r
# create a list of formulas for each equation

lm_fit <- lm(formula = formula(paste("treat", "~", paste(vars, collapse = "+"))), data = df)
# summary(lm_fit)
# summary(lm_fit)$fstatistic
```