SU Econometrics I Spring 2024

Problem Set 3: Marek Chadim[1]

1. (a)

$$\begin{aligned}
E\left[E(y|x_1)|x_1,x_2\right] &= E\left[E(y|x_1)|x_1\right] \quad (\text{because } E(y|x_1) \perp\!\!\!\perp x_2|x_1) \\
&= \left(\int_{\mathbb{R}^{k_1}} E(y|x_1) f(x_1)\, dx_1\right)|x_1 \\
&= E(y|x_1)\int_{\mathbb{R}^{k_1}} f(x_1)\, dx_1 \quad (\text{because } E(y|x_1) \text{ is a function of } X_1) \\
&= E(y|x_1)
\end{aligned}$$

$$\begin{aligned}
E\left[E(y|x_1,x_2)|x_1\right] &= \int_{\mathbb{R}^{k_2}} E(y|x_1,x_2) f(x_2|x_1)\, dx_2 \\
&= \int_{\mathbb{R}^{k_2}} \left(\int_{\mathbb{R}} y f(y|x_1,x_2)\, dy\right) f(x_2|x_1)\, dx_2 \\
&= \int_{\mathbb{R}^{k_2}} \left(\int_{\mathbb{R}} y f(y|x_1,x_2) f(x_2|x_1)\, dy\right) dx_2 \\
&\left[f(y|x_1,x_2)f(x_2|x_1) = \frac{f(y,x_1,x_2)}{f(x_1,x_2)}\frac{f(x_1,x_2)}{f(x_1)} = f(y,x_2|x_1)\right] \\
&= \int_{\mathbb{R}^{k_2}} \int_{\mathbb{R}} y f(y,x_2|x_1)\, dy\, dx_2 \\
&= \int_{\mathbb{R}} y \left(\int_{\mathbb{R}^{k_2}} f(y,x_2|x_1)\, dx_2\right) dy \quad (\text{because } E|Y| < \infty) \\
&= \int_{\mathbb{R}} y f(y|x_1)\, dy = E(y|x_1)
\end{aligned}$$

(b) i.

```
np.random.seed(0)
u0 = np.random.normal(0, sigma2, 500)
u1 = np.random.normal(0, 1, 500)
u2 = np.random.normal(0, 1, 500)
x1 = u0 + u1, x2 = u0 + u2
epsilon = np.random.normal(0, 1, 500)
sigma2 = 1
beta2 = 5
y = x1 + beta2*x2 + epsilon
correlation_x1_x2 = np.corrcoef(x1, x2)[0, 1]
correlation_x1_x2
0.48745413341163624
```

ii.

```
model = LinearRegression()
model.fit(x1.reshape(-1, 1), y)
y_hat_1 = model.predict(x1.reshape(-1, 1))
mse = mean_squared_error(y, y_hat_1)
mse
35.35939031894467
```

[1] 42624@student.hhs.se

iii.
```
X = np.column_stack((x1, x2))
model_full = LinearRegression()
model_full.fit(X, y)
x2_mean = np.mean(x2)
beta_0 = model_full.intercept_
beta_1, beta_2 = model_full.coef_
y_hat_2 = beta_0 + beta_1*x1 + beta_2*x2_mean
mse_y_hat_2 = mean_squared_error(y, y_hat_2)
mse_y_hat_2
46.087549635205555
```

iv.
```
model_x2 = LinearRegression()
model_x2.fit(x1.reshape(-1, 1), x2)
x2_hat = model_x2.predict(x1.reshape(-1, 1))
y_hat_3 = beta_0 + beta_1*x1 + beta_2*x2_hat
mse_y_hat_3 = mean_squared_error(y, y_hat_3)
mse_y_hat_3
35.35939031894467
```

v.
```
corr_y_yhat1 = np.corrcoef(y, y_hat_1)[0, 1]
corr_y_yhat2 = np.corrcoef(y, y_hat_2)[0, 1]
corr_y_yhat3 = np.corrcoef(y, y_hat_3)[0, 1]
corr_y_yhat1, corr_y_yhat2, corr_y_yhat3
(0.6138527807909712, 0.613852780790971, 0.6138527807909712)
```

In line with the result from the Law of Iterated Expectation, given available information on x1, the correlation of the fitted and true values stays the same.

```
x1_quad = x1**2
X_quad = np.column_stack((x1, x1_quad))
model_quad = LinearRegression()
model_quad.fit(X_quad, y)
y_pred_quad = model_quad.predict(X_quad)
mse_quad = mean_squared_error(y, y_pred_quad)
mse_quad
35.31789866407726

spline_transformer =
SplineTransformer(degree=3, n_knots=4, include_bias=False)
spline_pipeline =
make_pipeline(spline_transformer, LinearRegression())
spline_pipeline.fit(X1, y)
y_pred_spline = spline_pipeline.predict(X1)
mse_spline = mean_squared_error(y, y_pred_spline)
mse_spline
35.226184698933984
```

Higher order polynomials of x_1 provide in sample improvement by reducing bias of the estimates. However, their performance is likely to be worse when tested on new data due to higher variance of the estimates associated with overfitting.

2. (a)
```
np.random.seed(42)
earnings = np.random.normal(19, 1, 500)
capital_gains = np.random.normal(1, 1, 500)
u = np.random.normal(0, 1, 500)
e = np.random.normal(0, 1, 500)
occupational_status = earnings + u
child_outcomes = earnings - capital_gains + e
income = earnings + capital_gains
fraction_earnings = np.mean(earnings / income)
fraction_earnings
0.950501580683676
```

(b)
```
average_effect= 2*(fraction_earnings - (1-fraction_earnings))
average_effect
1.8020063227347038
```

(c)
```
Y = child_outcomes
X = add_constant(np.column_stack((earnings, capital_gains)))
model = OLS(Y, X).fit(), coef= model.params[:]
X_os = add_constant(np.column_stack((earnings, capital_gains,
occupational_status)))
model_os = OLS(Y, X_os).fit(), coef_os = model_os.params[:]
coef, coef_os
(array([ 0.96576913,  0.9499424 , -1.01915512]),
 array([ 1.01153196,  0.99865047, -1.01970993, -0.05101403]))

X_income = add_constant(income.reshape(-1, 1))
model_income = OLS(Y, X_income).fit()
coefficient_income = model_income.params[1]
coefficient_income
0.030418938549663543
```

Earnings and capital gains effect cancel each other out in the regression on pooled income. Income is uncorrelated with structural error thus not endogenous.

(d)
```
cor_os_income = np.corrcoef(occupational_status, income)[0, 1]
cor_os_income
0.498705133916934
```

(f) Variance-weighted average/signal-to-noise ratios
OV"B" $= \hat{\beta}_{os,income} \times \hat{\beta}_{os,co} = \frac{1}{1+1} \times 1 \times \frac{1}{1+1} \times 1 = .25$

(g)
```
X_income_os =
add_constant(np.column_stack((income, occupational_status)))
model_income_os = OLS(child_outcomes, X_income_os).fit()
coef_income_os= model_income_os.params[1]
coef_income_os
-0.3081210399949755
```

From zero to a substantial negative effect of income controlling for occupational status indicates the presence of omitted variable "bias" in terms of correlation with both the independent variable and the dependent variable. Yet, occupational status is a bad control as it screens off the earnings effect of income on child outcomes.

3. (a)

$$\hat{\beta} = \frac{\text{cov}(\tilde{x}, y)}{\text{var}(\tilde{x})} = \frac{E[(x+\nu)(\beta x+\epsilon)]}{\text{var}(x+\nu)} = \frac{E[x^2\beta + x\epsilon + \nu\beta x + \nu\epsilon]}{\text{var}(x) + \text{var}(\nu) + 2\text{cov}(x,\nu)}$$

$$plim\,\hat{\beta} = \frac{\beta\sigma_x^2 + 0 + \beta 0 + 0}{\sigma_x^2 + \sigma_\nu^2 + 2\times 0} = \frac{\sigma_x^2}{\sigma_x^2 + \sigma_\nu^2}\beta$$

Since $\frac{\sigma_x^2}{\sigma_x^2+\sigma_\nu^2} < 0$ the coefficient $\hat{\beta}$ will be biased towards zero.

(b)

$$\hat{\beta} = \frac{\text{cov}(x, \tilde{y})}{\text{var}(x)} = \frac{E[x(\beta x + \epsilon + \nu)]}{\text{var}(x)} = \frac{E[x^2\beta + x\epsilon + x\nu]}{\text{var}(x)}$$

$$plim\,\hat{\beta} = \frac{\beta\sigma_x^2 + 0 + 0}{\sigma_x^2} = \beta$$

Since ν is uncorrelated with x we can estimate β consistently by OLS in this case.

(c)

$$\hat{\beta} = \frac{\text{cov}(\tilde{x}, y)}{\text{var}(\tilde{x})} = \frac{E[(x+\nu)(\beta x+\epsilon)]}{\text{var}(x+\nu)} = \frac{E[x^2\beta + x\epsilon + \nu\beta x + \nu\epsilon]}{\text{var}(x) + \text{var}(\nu) + 2\text{cov}(x,\nu)}$$

$$plim\,\hat{\beta} = \frac{\beta\sigma_x^2 + 0 + \beta 0 + \sigma_{\nu\epsilon}}{\sigma_x^2 + \sigma_\nu^2 + 2\times 0} = \frac{1}{\sigma_x^2 + \sigma_\nu^2}(\sigma_x^2\beta + \sqrt{\sigma_\nu^2}\sqrt{\sigma_\epsilon^2}\rho)$$

The classical error is a special case where $\rho = 0$. Increasing σ_ν^2 impacts $\hat{\beta}$ by the attenuation factor $\frac{\sigma_x^2}{\sigma_x^2+\sigma_\nu^2}$ and gives more weight to the endogeneity bias from $\rho \neq 0$.

(d)

$$\hat{\beta}_{IV} = \frac{\text{cov}(y, z)}{\text{cov}(\tilde{x}, z)} = \frac{\text{cov}(\beta x + \epsilon, z)}{\text{cov}(x + \nu, z)} = \frac{E[z(\beta x + \epsilon)]}{E[z(x+\nu)]} = \frac{E[xz\beta + z\epsilon]}{E[xz + z\nu]}$$

$$plim\,\hat{\beta}_{IV} = \frac{\beta\cdot\sigma_{xz} + 0}{\sigma_{xz} + 0} = \beta$$

We get consistent estimate as long as z is correlated with x but not with ϵ and ν.