STAT230A – HW3 (Due 2/27 @5pm PT)

1. Problem 6.7 - Unbiased sandwich variance estimator under the Gauss–Markov model

Under the Gauss–Markov model with $\sigma_i^2 = \sigma^2$ for $i = 1, \ldots, n$, show that the HC0 version of $\hat{V}_{\text{EHW}}$ is biased but the HC2 version of $\tilde{V}_{\text{EHW}}$ is unbiased for $\text{cov}(\hat{\beta})$.

2. Problem 7.3 – Projection matrices

Prove Lemma 7.2.

Hint: Use Lemma 7.1.

Lemma 7.2 *We have*

$$H_1 \tilde{H}_2 = \tilde{H}_2 H_1 = 0, \quad H = H_1 + \tilde{H}_2.$$

Lemma 7.2 is purely algebraic. I leave the proof as Problem 7.3. The first two identities implies that the column space of $\tilde{X}_2$ is orthogonal to the column space of X_1. The last identity $H = H_1 + \tilde{H}_2$ has a clear geometric interpretation. For any vector $v \in \mathbb{R}^n$, we have $Hv = H_1 v + \tilde{H}_2 v$, so the projection of v onto the column space of X equals the summation of the projection of v onto the column space of X_1 and the projection of v onto the column space of $\tilde{X}_2$. Importantly, $H \neq H_1 + H_2$ in general.

Second, we can obtain $\hat{\beta}_2$ from (7.3) or (7.4), which corresponds to the partial regression of Y on $\tilde{X}_2$ or the partial regression of $\tilde{Y}$ on $\tilde{X}_2$. We can verify that the residual vector from the second partial regression equals the residual vector from the full regression.

3. Problem 7.7 - QR decomposition of X and the computation of OLS

Verify that the R matrix equals

$$R = \begin{pmatrix} Q_1^{\mathrm{T}} X_1 & Q_1^{\mathrm{T}} X_2 & \cdots & Q_1^{\mathrm{T}} X_p \\ 0 & Q_2^{\mathrm{T}} X_2 & \cdots & Q_2^{\mathrm{T}} X_p \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & Q_p^{\mathrm{T}} X_p \end{pmatrix}.$$

Based on the QR decompostion of X, show that

$$H = QQ^{\mathrm{T}},$$

and h_{ii} equals the squared length of the i-th row of Q.

4. Problem 8.5 - Formula of the partial correlation coefficient

Prove Theorem 8.1.

Theorem 8.1 *For $Y, X, W \in \mathbb{R}^n$, we have*

$$\hat{\rho}_{yx|w} = \frac{\hat{\rho}_{yx} - \hat{\rho}_{yw}\hat{\rho}_{xw}}{\sqrt{1-\hat{\rho}_{yw}^2}\sqrt{1-\hat{\rho}_{xw}^2}}.$$

Its proof is purely algebraic, so I leave it as Problem 8.5. Theorem 8.1 states that we can obtain the sample partial correlation coefficient based on the three pairwise correlation coefficients. Figure 8.1 illustrates the interplay among three variables. In particular, the correlation between x and y is due to two "pathways": the one acting through w and the one acting independent of w. The first path way is related to the product term $\hat{\rho}_{yw}\hat{\rho}_{xw}$, and the second pathway is related to $\hat{\rho}_{yx|w}$. This gives some intuition for Theorem 8.1.

Based on data $(y_i, x_i, w_i)_{i=1}^n$, we can compute the sample correlation matrix

$$\hat{R} = \begin{pmatrix} 1 & \hat{\rho}_{yx} & \hat{\rho}_{yw} \\ \hat{\rho}_{xy} & 1 & \hat{\rho}_{xw} \\ \hat{\rho}_{wy} & \hat{\rho}_{wx} & 1 \end{pmatrix},$$

which is symmetric and positive semi-definite. Simpson's paradox happens if and only if

$$\hat{\rho}_{yx}(\hat{\rho}_{yx} - \hat{\rho}_{yw}\hat{\rho}_{xw}) < 0 \Longleftrightarrow \hat{\rho}_{yx}^2 < \hat{\rho}_{yx}\hat{\rho}_{yw}\hat{\rho}_{xw}.$$

5. Problem 8.6 - Examples of Simpson's Paradox

Give three numerical examples of (Y, X, W) which causes Simpson's Paradox. Report the mean and covariance matrix for each example.

Note: you only need to give one numerical example and report the mean and covariance matrix

6. A simplified version of Problem 6.5

Empirical comparison of the standard errors Read Long and Ervin (2000). Conduct a Monte Carlo simulation study to explore the finite sample performance of the OLS estimator of the standard error and the four versions of the heteroskedasticity-robust standard error estimators. Follow the data generating process in Long and Ervin (2000) with heteroskedasticity, which is described in Equation (5) with the error structure 2 in Table 2. Replicate this simulation study and replicate their Figure 2. Briefly discuss what you learn from this simulation study.

1. Problem 6.7 - Unbiased sandwich variance estimator under the Gauss–Markov model

Under the Gauss–Markov model with $\sigma_i^2 = \sigma^2$ for $i = 1, \ldots, n$, show that the HC0 version of $\hat{V}_{\text{EHW}}$ is biased but the HC2 version of $\tilde{V}_{\text{EHW}}$ is unbiased for $\text{cov}(\hat{\beta})$.

Under Gauss Markov Model: $\quad \text{cov}(\hat{\beta}) = \sigma^2 (X^T X)^{-1}$

$$\hat{V}_{EHW,k} = (X^T X)^{-1} (X^T \hat{\Omega}_k X)(X^T X)^{-1}$$

where $\hat{\Omega}_k := \text{diag}\{\hat{\varepsilon}_{1,k}^2, \hat{\varepsilon}_{2,k}^2, \ldots, \hat{\varepsilon}_{n,k}^2\}$

$k=0 \Rightarrow HC_0 : \quad \hat{\varepsilon}_{i,0} = \hat{\varepsilon}_i$

$k=2 \Rightarrow HC_2 : \quad \hat{\varepsilon}_{i,2} = \hat{\varepsilon}_i / \sqrt{1-h_{jj}} \quad$ where $h_{jj} = \text{diag}[H]$

$$\mathbb{E}[\hat{V}_{EHW,0}] = \mathbb{E}\left[(X^T X)^{-1}(X^T \hat{\Omega}_0 X)(X^T X)^{-1}\right]$$

$$= (X^T X)^{-1}(X^T \underline{\mathbb{E}[\hat{\Omega}_0]} X)(X^T X)^{-1}$$

$$= \mathbb{E}\left[\begin{bmatrix} \hat{\varepsilon}_1^2 & & \\ & \hat{\varepsilon}_2^2 & \\ & & \ddots \\ & & & \hat{\varepsilon}_n^2 \end{bmatrix}\right] = \mathbb{E}\left[\begin{bmatrix} \ddots & & \\ & \{\hat{\varepsilon}_i - \cancelto{0}{\mathbb{E}[\hat{\varepsilon}_i]}\}^2 & \\ & & \ddots \end{bmatrix}\right]$$

$$= \begin{bmatrix} \ddots & & \\ & \mathbb{E}[\{\hat{\varepsilon}_i - \mathbb{E}[\hat{\varepsilon}_i]\}^2] & \\ & & \ddots \end{bmatrix} = \begin{bmatrix} \ddots & & \\ & \text{Var}[\hat{\varepsilon}_i] & \\ & & \ddots \end{bmatrix} = \left[\text{diag}\{\text{cov}[\hat{\varepsilon}]\}\right]$$

$$\text{cov}[\hat{\varepsilon}] = \text{cov}[Y - \hat{Y}] = \text{cov}[\{I_n - H\} Y]$$
$$= \sigma^2 \{I_n - H\}\{I_n - H\}^T = \sigma^2 \{I_n - H\}$$

$$= \begin{bmatrix} \ddots & & \\ & \sigma^2(1-h_{ii}) & \\ & & \ddots \end{bmatrix} = \sigma^2 I_n - \sigma^2 \text{diag}\{h_{ii}\}$$

$$= \sigma^2 (X^T X)^{-1}(X^T X)(X^T X)^{-1} + \underline{\underline{\sigma^2 (X^T X)^{-1} X^T \text{diag}^{\mathbb{R}^{n \times n}}\{h_{ii}\} X (X^T X)^{-1}}}$$

$$\mathbb{E}[\hat{V}_{EHW,0}] = \sigma^2 (X^T X)^{-1} + \boxed{\text{some non zero thing else}} \neq \text{cov}(\hat{\beta}) \Rightarrow \boxed{\text{BIASED.}}$$

$$\mathbb{E}\left[\hat{V}_{EHW,2}\right] = \mathbb{E}\left[(X^T X)^{-1}(X^T \hat{\Omega}_2 X)(X^T X)^{-1}\right]$$

$$= (X^T X)^{-1}\left(X^T \underline{\mathbb{E}\left[\hat{\Omega}_2\right]} X\right)(X^T X)^{-1}$$

$$= \mathbb{E}\left[\begin{bmatrix} \hat{\varepsilon}_1^2/\sqrt{1-h_{11}}^2 & & & \\ & \hat{\varepsilon}_2^2/\sqrt{1-h_{22}}^2 & & \\ & & \ddots & \\ & & & \hat{\varepsilon}_n^2/\sqrt{1-h_{nn}}^2 \end{bmatrix}\right]$$

$$= \mathbb{E}\left[\begin{bmatrix} \ddots & & \\ & \{\hat{\varepsilon}_i - \mathbb{E}[\varepsilon_i]\}^2/(1-h_{ii}) & \\ & & \ddots \end{bmatrix}\right]$$

$$= \left[\mathrm{diag}\left\{\mathrm{cov}[\hat{\varepsilon}]\right\}/(1-h_{ii})\right]$$

$$= \begin{bmatrix} \ddots & & \\ & \sigma^2(1-h_{ii})/(1-h_{ii}) & \\ & & \ddots \end{bmatrix}$$

$$= \sigma^2 I_n$$

$$= \sigma^2 (X^T X)^{-1}(X^T X)(X^T X)^{-1}$$

$$= \sigma^2 (X^T X)^{-1}$$

$$= \mathrm{cov}(\hat{\beta}) \Rightarrow \boxed{\text{UNBIASED}}$$

2. Problem 7.3 – Projection matrices

Prove Lemma 7.2.

Hint: Use Lemma 7.1.

Lemma 7.2 *We have*

$$H_1\tilde{H}_2 = \tilde{H}_2 H_1 = 0, \quad H = H_1 + \tilde{H}_2.$$

Lemma 7.2 is purely algebraic. I leave the proof as Problem 7.3. The first two identities implies that the column space of $\tilde{X}_2$ is orthogonal to the column space of X_1. The last identity $H = H_1 + \tilde{H}_2$ has a clear geometric interpretation. For any vector $v \in \mathbb{R}^n$, we have $Hv = H_1 v + \tilde{H}_2 v$, so the projection of v onto the column space of X equals the summation of the projection of v onto the column space of X_1 and the projection of v onto the column space of $\tilde{X}_2$. Importantly, $H \neq H_1 + H_2$ in general.

Second, we can obtain $\hat{\beta}_2$ from (7.3) or (7.4), which corresponds to the partial regression of Y on $\tilde{X}_2$ or the partial regression of $\tilde{Y}$ on $\tilde{X}_2$. We can verify that the residual vector from the second partial regression equals the residual vector from the full regression.

$$H_1\tilde{H}_2 = X_1(X_1^TX_1)^{-1}X_1^T\,\tilde{X}_2(\tilde{X}_2^T\tilde{X}_2)^{-1}\tilde{X}_2^T$$

$$= X_1(X_1^TX_1)^{-1}\underbrace{\cancel{X_1^T(I_n-H_1)}}_{\vec{0}^T}X_2\left(X_2^T(I_n-H_1)^2X_2\right)^{-1}X_2^T(I_n-H_1) = \boxed{\vec{0}}$$

$$\tilde{H}_2H_1 = (I_n-H_1)X_2\left(X_2^T(I_n-H_1)^2X_2\right)^{-1}X_2^T\underbrace{\cancel{(I_n-H_1)X_1}}_{\vec{0}}(X_1^TX_1)^{-1}X_1^T = \boxed{\vec{0}}$$

$$H = X(X^TX)^{-1}X^T$$

$$= [X_1 \,\vdots\, X_2]\left\{\begin{bmatrix}X_1^T\\ \hdashline X_2^T\end{bmatrix}[X_1 \,\vdots\, X_2]\right\}^{-1}\begin{bmatrix}X_1^T\\ \hdashline X_2^T\end{bmatrix}$$

$$= [X_1 \,\vdots\, X_2]\left\{\underline{\underline{\begin{bmatrix}X_1^TX_1 & X_1^TX_2\\ X_2^TX_1 & X_2^TX_2\end{bmatrix}}}\right\}^{-1}\begin{bmatrix}X_1^T\\ \hdashline X_2^T\end{bmatrix}$$

We invert this block matrix with:

$$\begin{bmatrix}A & B\\ C & D\end{bmatrix}^{-1} = \begin{bmatrix}A^{-1}+A^{-1}B(D-CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D-CA^{-1}B)^{-1}\\ -(D-CA^{-1}B)^{-1}CA^{-1} & (D-CA^{-1}B)^{-1}\end{bmatrix}$$

$H =$ NEXT PAGE

$$H = \begin{bmatrix} X_1 & X_2 \end{bmatrix} \begin{bmatrix} (X_1^TX_1)^{-1} + (X_1^TX_1)^{-1}X_1^TX_2\left(X_2^TX_2 - X_2^TX_1(X_1^TX_1)^{-1}X_1^TX_2\right)^{-1}X_2^TX_1(X_1^TX_1)^{-1} & -(X_1^TX_1)^{-1}X_1^TX_2\left(X_2^TX_2 - X_2^TX_1(X_1^TX_1)^{-1}X_1^TX_2\right)^{-1} \\ -\left(X_2^TX_2 - X_2^TX_1(X_1^TX_1)^{-1}X_1^TX_2\right)^{-1}X_2^TX_1(X_1^TX_1)^{-1} & \left(X_2^TX_2 - X_2^TX_1(X_1^TX_1)^{-1}X_1^TX_2\right)^{-1} \end{bmatrix} \begin{bmatrix} X_1^T \\ X_2^T \end{bmatrix}$$

$$\downarrow \quad \left(X_2^TX_2 - X_2^TX_1(X_1^TX_1)^{-1}X_1^TX_2\right)^{-1} = \left[X_2^T\{I_n - H_1\}X_2\right]^{-1} = \left[\tilde{X}_2^T\tilde{X}_2\right]^{-1}$$

$$= \begin{bmatrix} X_1 & X_2 \end{bmatrix} \begin{bmatrix} (X_1^TX_1)^{-1} + (X_1^TX_1)^{-1}X_1^TX_2\left[\tilde{X}_2^T\tilde{X}_2\right]^{-1}X_2^TX_1(X_1^TX_1)^{-1} & -(X_1^TX_1)^{-1}X_1^TX_2\left[\tilde{X}_2^T\tilde{X}_2\right]^{-1} \\ -\left[\tilde{X}_2^T\tilde{X}_2\right]^{-1}X_2^TX_1(X_1^TX_1)^{-1} & \left[\tilde{X}_2^T\tilde{X}_2\right]^{-1} \end{bmatrix} \begin{bmatrix} X_1^T \\ X_2^T \end{bmatrix}$$

$$= \begin{bmatrix} X_1 & X_2 \end{bmatrix} \begin{bmatrix} (X_1^TX_1)^{-1}X_1^T + (X_1^TX_1)^{-1}X_1^TX_2\left[\tilde{X}_2^T\tilde{X}_2\right]^{-1}X_2^TX_1(X_1^TX_1)^{-1}X_1^T - (X_1^TX_1)^{-1}X_1^TX_2\left[\tilde{X}_2^T\tilde{X}_2\right]^{-1}X_2^T \\ -\left[\tilde{X}_2^T\tilde{X}_2\right]^{-1}X_2^TX_1(X_1^TX_1)^{-1}X_1^T + \left[\tilde{X}_2^T\tilde{X}_2\right]^{-1}X_2^T \end{bmatrix}$$

$$\downarrow \quad X_1(X_1^TX_1)^{-1}X_1^T = H_1$$

$$= \begin{bmatrix} X_1 & X_2 \end{bmatrix} \begin{bmatrix} (X_1^TX_1)^{-1}X_1^T + (X_1^TX_1)^{-1}X_1^TX_2\left[\tilde{X}_2^T\tilde{X}_2\right]^{-1}X_2^TH_1 - (X_1^TX_1)^{-1}X_1^TX_2\left[\tilde{X}_2^T\tilde{X}_2\right]^{-1}X_2^T \\ -\left[\tilde{X}_2^T\tilde{X}_2\right]^{-1}X_2^TH_1 + \left[\tilde{X}_2^T\tilde{X}_2\right]^{-1}X_2^T \end{bmatrix}$$

$$= X_1(X_1^TX_1)^{-1}X_1^T + X_1(X_1^TX_1)^{-1}X_1^TX_2\left[\tilde{X}_2^T\tilde{X}_2\right]^{-1}X_2^TH_1 - X_1(X_1^TX_1)^{-1}X_1^TX_2\left[\tilde{X}_2^T\tilde{X}_2\right]^{-1}X_2^T - X_2\left[\tilde{X}_2^T\tilde{X}_2\right]^{-1}X_2^TH_1 + X_2\left[\tilde{X}_2^T\tilde{X}_2\right]^{-1}X_2^T$$

$$= H_1 + H_1X_2\left[\tilde{X}_2^T\tilde{X}_2\right]^{-1}X_2^TH_1 - H_1X_2\left[\tilde{X}_2^T\tilde{X}_2\right]^{-1}X_2^T - X_2\left[\tilde{X}_2^T\tilde{X}_2\right]^{-1}X_2^TH_1 + X_2\left[\tilde{X}_2^T\tilde{X}_2\right]^{-1}X_2^T$$

$$= H_1 + \{I_n - H_1\}X_2\left[\tilde{X}_2^T\tilde{X}_2\right]^{-1}X_2^T\{I_n - H_1\}$$

$$= H_1 + \tilde{X}_2\left[\tilde{X}_2^T\tilde{X}_2\right]^{-1}\tilde{X}_2$$

$$= \boxed{H_1 + \tilde{H}_2}$$

3. Problem 7.7 - QR decomposition of X and the computation of OLS

Verify that the R matrix equals

$$R = \begin{pmatrix} Q_1^{\mathrm{T}} X_1 & Q_1^{\mathrm{T}} X_2 & \cdots & Q_1^{\mathrm{T}} X_p \\ 0 & Q_2^{\mathrm{T}} X_2 & \cdots & Q_2^{\mathrm{T}} X_p \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & Q_p^{\mathrm{T}} X_p \end{pmatrix}.$$

Based on the QR decompostion of X, show that

$$H = QQ^{\mathrm{T}},$$

and h_{ii} equals the squared length of the i-th row of Q.

By QR decomposition:

$$\vec{u}_1 := \vec{X}_1$$

$$\vec{u}_2 := \vec{X}_2 - \hat{\beta}_{x_2|u_1}\vec{u}_1 = \vec{X}_2 - \frac{\vec{X}_2^T \vec{u}_1}{\vec{u}_1^T \vec{u}_1}\vec{u}_1$$

$$\vdots$$

$$\vec{u}_k := \vec{X}_k - \sum_{j=1}^{k-1}\hat{\beta}_{x_k|u_j}\vec{u}_j = \vec{X}_k - \sum_{j=1}^{k-1}\frac{\vec{X}_k^T \vec{u}_j}{\vec{u}_j^T \vec{u}_j}\vec{u}_j$$

$$\vdots$$

$$\hat{\beta}_{x_i|u_j} := \frac{\vec{x}_i^T \vec{u}_j}{\|\vec{u}_j\|^2}$$

$$X = \begin{bmatrix} | & | & & | \\ \vec{X}_1 & \vec{X}_2 & \cdots & \vec{X}_p \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{n\times p}$$

$$X = \underbrace{\begin{bmatrix} | & | & & | \\ \vec{u}_1 & \vec{u}_2 & \cdots & \vec{u}_p \\ | & | & & | \end{bmatrix}}_{\in\mathbb{R}^{n\times p}} \underbrace{\begin{bmatrix} 1 & \hat{\beta}_{x_2|u_1} & \hat{\beta}_{x_3|u_1} & \cdots & \hat{\beta}_{x_p|u_1} \\ 0 & 1 & \hat{\beta}_{x_3|u_2} & \cdots & \hat{\beta}_{x_p|u_2} \\ \vdots & \vdots & 1 & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}}_{\in\mathbb{R}^{p\times p}}$$

$$\vec{Q}_j := \frac{\vec{u}_j}{\|\vec{u}_j\|}$$

$$X = \underbrace{\begin{bmatrix} | & | & & | \\ \vec{Q}_1 & \vec{Q}_2 & \cdots & \vec{Q}_p \\ | & | & & | \end{bmatrix}}_{\in\mathbb{R}^{n\times p}} \underbrace{\underbrace{\begin{bmatrix} \|\vec{u}_1\| & & & \\ & \|\vec{u}_2\| & & \\ & & \ddots & \\ & & & \|\vec{u}_p\| \end{bmatrix}}_{\in\mathbb{R}^{p\times p}} \underbrace{\begin{bmatrix} 1 & \hat{\beta}_{x_2|u_1} & \hat{\beta}_{x_3|u_1} & \cdots & \hat{\beta}_{x_p|u_1} \\ 0 & 1 & \hat{\beta}_{x_3|u_2} & \cdots & \hat{\beta}_{x_p|u_2} \\ \vdots & \vdots & 1 & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}}_{\in\mathbb{R}^{p\times p}}}_{R}$$

$$X = \underbrace{\begin{bmatrix} | & | & & | \\ \vec{Q}_1 & \vec{Q}_2 & \cdots & \vec{Q}_p \\ | & | & & | \end{bmatrix}}_{\in\mathbb{R}^{n\times p}} [R]$$

Since $\vec{u}_i \perp \vec{u}_j$,

$$\left\{\frac{\vec{X}_k^T \vec{u}_i}{\|\vec{u}_i\|^2}\vec{u}_i\right\}^T \left\{\frac{\vec{X}_k^T \vec{u}_j}{\|\vec{u}_j\|^2}\vec{u}_j\right\} = \begin{cases} \{\ldots\}\,\vec{u}_i^T \vec{u}_j\,\{\ldots\} = 0 & \forall\ i \neq j \\ \dfrac{\vec{X}_k^T \vec{u}_i}{\|\vec{u}_i\|^2}\cancel{\vec{u}_i^T \vec{u}_i}\dfrac{\vec{X}_k^T \vec{u}_i}{\cancel{\|\vec{u}_i\|^2}} = \vec{X}_k^T \dfrac{\vec{X}_k^T \vec{u}_i}{\|\vec{u}_i\|^2}\vec{u}_i & \forall\ i = j \end{cases}$$

$$\hat{\beta}_{x_k|u_k} = \frac{\vec{X}_k^T\left\{\vec{X}_k - \sum_{j=1}^{k-1}\frac{\vec{X}_k^T \vec{u}_j}{\|\vec{u}_j\|^2}\vec{u}_j\right\}}{\left\|\vec{X}_k - \sum_{j=1}^{k-1}\frac{\vec{X}_k^T \vec{u}_j}{\|\vec{u}_j\|^2}\vec{u}_j\right\|^2} = \frac{\vec{X}_k^T \vec{X}_k^T - \sum_{i=1}^{k-1}\vec{X}_k^T \frac{\vec{X}_k^T \vec{u}_j}{\|\vec{u}_j\|^2}\vec{u}_j}{\left\|\vec{X}_k - \sum_{j=1}^{k-1}\frac{\vec{X}_k^T \vec{u}_j}{\|\vec{u}_j\|^2}\vec{u}_j\right\|^2}$$

$$\hat{\beta}_{x_k|u_k} = \frac{\vec{X}_k^T \vec{X}_k^T - \sum_{i=1}^{k-1}\vec{X}_k^T \frac{\vec{X}_k^T \vec{u}_j}{\|\vec{u}_j\|^2}\vec{u}_j - \sum_{i=1}^{k-1}\vec{X}_k^T \frac{\vec{X}_k^T \vec{u}_j}{\|\vec{u}_j\|^2}\vec{u}_j + \sum_{i=1}^{k-1}\vec{X}_k^T \frac{\vec{X}_k^T \vec{u}_j}{\|\vec{u}_j\|^2}\vec{u}_j}{\left\|\vec{X}_k - \sum_{j=1}^{k-1}\frac{\vec{X}_k^T \vec{u}_j}{\|\vec{u}_j\|^2}\vec{u}_j\right\|^2} = \frac{\left\{\vec{X}_k - \sum_{j=1}^{k-1}\frac{\vec{X}_k^T \vec{u}_j}{\|\vec{u}_j\|^2}\vec{u}_j\right\}^T\left\{\vec{X}_k - \sum_{j=1}^{k-1}\frac{\vec{X}_k^T \vec{u}_j}{\|\vec{u}_j\|^2}\vec{u}_j\right\}}{\left\|\vec{X}_k - \sum_{j=1}^{k-1}\frac{\vec{X}_k^T \vec{u}_j}{\|\vec{u}_j\|^2}\vec{u}_j\right\|^2} = 1$$

$$R = \underbrace{\begin{bmatrix} \|\vec{u}_1\| & & & \\ & \|\vec{u}_2\| & & \\ & & \ddots & \\ & & & \|\vec{u}_p\| \end{bmatrix}}_{\in \mathbb{R}^{p\times p}} \underbrace{\begin{bmatrix} 1 & \hat{\beta}_{x_2|u_1} & \hat{\beta}_{x_3|u_1} & \cdots & \hat{\beta}_{x_p|u_1} \\ 0 & 1 & \hat{\beta}_{x_3|u_2} & \cdots & \hat{\beta}_{x_p|u_2} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}}_{\in \mathbb{R}^{p\times p}}$$

$$= \begin{bmatrix} \|\vec{u}_1\| & \|\vec{u}_1\| \vec{\beta}_{x_2|u_1} & \|\vec{u}_1\| \vec{\beta}_{x_3|u_1} & \cdots & \|\vec{u}_1\| \vec{\beta}_{x_p|u_1} \\ 0 & \|\vec{u}_2\| & \|\vec{u}_2\| \vec{\beta}_{x_3|u_2} & \cdots & \|\vec{u}_2\| \vec{\beta}_{x_p|u_2} \\ 0 & 0 & \|\vec{u}_3\| & \cdots & \|\vec{u}_3\| \vec{\beta}_{x_p|u_3} \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \|\vec{u}_p\| \end{bmatrix}$$

$$= \begin{bmatrix} \|\vec{u}_1\| \frac{\vec{u}_1^T \vec{u}_1}{\|\vec{u}_1\|^2} & \|\vec{u}_1\| \frac{\vec{X}_2^T \vec{u}_1}{\|\vec{u}_1\|^2} & \|\vec{u}_1\| \frac{\vec{X}_3^T \vec{u}_1}{\|\vec{u}_1\|^2} & \cdots & \cdots \\ 0 & \|\vec{u}_2\| \frac{\vec{X}_2^T \vec{u}_2}{\|\vec{u}_2\|^2} & \|\vec{u}_2\| \frac{\vec{X}_3^T \vec{u}_2}{\|\vec{u}_2\|^2} & \cdots & \cdots \\ 0 & 0 & \|\vec{u}_3\| \frac{\vec{X}_3^T \vec{u}_3}{\|\vec{u}_3\|^2} & \cdots \|\vec{u}_i\| \frac{\vec{X}_i^T \vec{u}_i}{\|\vec{u}_i\|^2} & \cdots \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \ddots \end{bmatrix}$$

$$= \begin{bmatrix} & \vec{X}_j^T \frac{\vec{u}_i}{\|\vec{u}_i\|} \\ \text{zeros.} & \end{bmatrix} = \begin{bmatrix} & \vec{X}_j^T \vec{Q}_i = \vec{Q}_i^T \vec{X}_j \\ \text{zeros.} & \end{bmatrix}$$

$$R = \begin{bmatrix} \vec{Q}_1^T \vec{X}_1 & \vec{Q}_1^T X_2 & \cdots & \vec{Q}_1^T \vec{X}_p \\ & & \ddots & \vdots \\ & \text{zeros.} & & \vec{Q}_p^T \vec{X}_p \end{bmatrix}$$

$$H = X(X^TX)^{-1}X^T = QR\left((QR)^TQR\right)^{-1}(QR)^T$$

$$= QR\{R^TQ^TQR\}^{-1}R^TQ^T$$

$$= QR(R^TR)^{-1}R^TQ^T$$

NOTE HERE: each column of R : $\vec{R_i} \perp \vec{R_j}$ thus $R^{p \times p}$ is invertible.

$$= Q\,RR^{-1}R^{-1^T}R^TQ^T$$

$$= Q\,I\,I^TQ^T$$

$$H = QQ^T$$

$$\begin{bmatrix} \ddots & & H_{ij}\text{'s} \\ & H_{ii} & \\ H_{ij}\text{'s} & & \ddots \end{bmatrix} = \begin{bmatrix} \text{---} \; Q_{1,:} \; \text{---} \\ \text{---} \; Q_{2,:} \; \text{---} \\ \vdots \\ \text{---} \; Q_{n,:} \; \text{---} \end{bmatrix} \begin{bmatrix} | & | & & | \\ Q_{1,:}^T & Q_{2,:}^T & \cdots & Q_{n,:} \\ | & | & & | \end{bmatrix}$$

where $Q_{i,:} \in \mathbb{R}^{1 \times n}$

$$\Downarrow$$

$$\boxed{H_{ii} = \underbrace{Q_{i,:}}_{\mathbb{R}^{1\times n}}\;\underbrace{Q_{i,:}^T}_{\mathbb{R}^{1\times n^T} = \mathbb{R}^{n\times 1}} = \|Q_{i,:}\|^2}$$

4. Problem 8.5 - Formula of the partial correlation coefficient

Prove Theorem 8.1.

Theorem 8.1 *For $Y, X, W \in \mathbb{R}^n$, we have*

$$\hat{\rho}_{yx|w} = \frac{\hat{\rho}_{yx} - \hat{\rho}_{yw}\hat{\rho}_{xw}}{\sqrt{1-\hat{\rho}_{yw}^2}\sqrt{1-\hat{\rho}_{xw}^2}}.$$

Its proof is purely algebraic, so I leave it as Problem 8.5.

$$\hat{\rho}_{yx|w} = \frac{\sum \hat{\varepsilon}_{xi}\hat{\varepsilon}_{yi}}{\sqrt{\sum \hat{\varepsilon}_{xi}^2}\sqrt{\sum \hat{\varepsilon}_{yi}^2}} \quad ; \quad \{\vec{X}\} \sim \{\vec{1}, \vec{W}\} \Rightarrow \begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{pmatrix} := \begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_{xw} \end{pmatrix}$$

$\Updownarrow$ take out the intercept.

$$\{\vec{X} - \bar{X}\} \sim \{\vec{W} - \bar{W}\} \Rightarrow \hat{\beta}_1 := \hat{\beta}_{xw}$$

$$\hat{\rho}_{yw} = \frac{\sum (y_i - \bar{y})(w_i - \bar{w})}{\sqrt{\sum (y_i - \bar{y})^2}\sqrt{\sum (w_i - \bar{w})^2}}$$

$$\hat{\rho}_{xw} = \frac{\sum (x_i - \bar{x})(w_i - \bar{w})}{\sqrt{\sum (x_i - \bar{x})^2}\sqrt{\sum (w_i - \bar{w})^2}}$$

$$\hat{\rho}_{yx} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2}\sqrt{\sum (y_i - \bar{y})^2}}$$

$$\begin{cases} \hat{\beta}_{xw} = \left\{\{\vec{W} - \bar{W}\}^T\{\vec{W} - \bar{W}\}\right\}^{-1}\{\vec{W} - \bar{W}\}\{\vec{X} - \bar{X}\} = \dfrac{\sum (w_i - \bar{w})(x_i - \bar{x})}{\sum (w_i - \bar{w})^2} \\ \hat{\beta}_{yw} = \dfrac{\sum (w_i - \bar{w})(y_i - \bar{y})}{\sum (w_i - \bar{w})^2} \end{cases}$$

$$\hat{\beta}_{yx} = \frac{\sum (y_i - \bar{y})(x_i - \bar{x})}{\sum (x_i - \bar{x})^2} = \frac{\sum (y_i - \bar{y})(x_i - \bar{x})}{\hat{\sigma}_x^2}$$

$$\Rightarrow \hat{\beta}_{xw} = \hat{\rho}_{xw} \cdot \frac{\sqrt{\sum (x_i - \bar{x})^2}}{\sqrt{\sum (w_i - \bar{w})^2}} = \hat{\rho}_{xw} \cdot \frac{\hat{\sigma}_x}{\hat{\sigma}_w}$$

$$\Rightarrow \hat{\beta}_{yw} = \hat{\rho}_{yw} \cdot \frac{\hat{\sigma}_y}{\hat{\sigma}_w}$$

$$\Rightarrow \hat{\beta}_{yx} = \hat{\rho}_{yx} \cdot \frac{\hat{\sigma}_y}{\hat{\sigma}_x}$$

⇒ WE CAN DERIVE SOME USEFUL IDENTITIES.

$$\boxed{\hat\rho_{xw} = \frac{\hat\sigma_w}{\hat\sigma_x}\hat\beta_{xw}} \quad (1) \qquad \boxed{\hat\rho_{yw} = \frac{\hat\sigma_w}{\hat\sigma_y}\hat\beta_{yw}} \quad (2) \qquad \boxed{\hat\rho_{yx} = \frac{\hat\sigma_x}{\hat\sigma_y}\hat\beta_{yw}} \quad (3)$$

(1) & (2):

$$\hat\rho_{yw}\,\hat\rho_{xw} = \frac{\hat\sigma_w^2}{\hat\sigma_x\,\hat\sigma_y}\hat\beta_{xw}\,\hat\beta_{yw}$$

$$\boxed{\hat\rho_{yw}\,\hat\rho_{xw} = \frac{\Sigma(w_i-\bar w)^2}{\hat\sigma_x\,\hat\sigma_y}\hat\beta_{xw}\,\hat\beta_{yw}} \quad (4)$$

(3):

$$\hat\rho_{yx} = \frac{\hat\sigma_x}{\hat\sigma_y}\,\frac{\Sigma(y_i-\bar y)(x_i-\bar x)}{\hat\sigma_x^2}$$

$$\boxed{\hat\rho_{yx} = \frac{\Sigma(y_i-\bar y)(x_i-\bar x)}{\hat\sigma_y\,\hat\sigma_x}} \quad (5)$$

$(1)^2$:

$$\hat\rho_{xw}^2 = \frac{\Sigma(w_i-\bar w)^2}{\Sigma(x_i-\bar x)^2}\left[\frac{\Sigma(w_i-\bar w)(x_i-\bar x)}{\Sigma(w_i-\bar w)^2}\right]^2$$

$$\boxed{\hat\rho_{xw}^2 = \frac{\left[\Sigma(w_i-\bar w)(x_i-\bar x)\right]^2}{\Sigma(x_i-\bar x)^2\,\Sigma(w_i-\bar w)^2}} \quad (6)$$

$(2)^2$:

$$\boxed{\hat\rho_{yw}^2 = \frac{\left[\Sigma(w_i-\bar w)(y_i-\bar y)\right]^2}{\Sigma(y_i-\bar y)^2\,\Sigma(w_i-\bar w)^2}} \quad (7)$$

$$\hat{\rho}_{yx|w} = \frac{\sum \hat{\varepsilon}_{x_i}\,\hat{\varepsilon}_{y_i}}{\sqrt{\sum \hat{\varepsilon}_{x_i}^2}\;\sqrt{\sum \hat{\varepsilon}_{y_i}^2}}$$

$$\begin{cases} \hat{\varepsilon}_{x_i} = x_i - \bar{x} - \hat{\beta}_{xw}(w_i - \bar{w}) \\ \hat{\varepsilon}_{y_i} = y_i - \bar{y} - \hat{\beta}_{yw}(w_i - \bar{w}) \end{cases}$$

$$= \frac{\sum\left(x_i - \bar{x} - \hat{\beta}_{xw}(w_i - \bar{w})\right)\left(y_i - \bar{y} - \hat{\beta}_{yw}(w_i - \bar{w})\right)}{\sqrt{\sum\left(x_i - \bar{x} - \hat{\beta}_{xw}(w_i - \bar{w})\right)^2}\;\sqrt{\sum\left(y_i - \bar{y} - \hat{\beta}_{yw}(w_i - \bar{w})\right)}} = \frac{\text{numerator}}{\text{denominator}}$$

$$\text{numerator} = \sum\left[(x_i - \bar{x}) - \hat{\beta}_{xw}(w_i - \bar{w})\right]\left[(y_i - \bar{y}) - \hat{\beta}_{yw}(w_i - \bar{w})\right]$$

$$= \sum\left[(x_i - \bar{x})(y_i - \bar{y}) - \hat{\beta}_{xw}(w_i - \bar{w})(y_i - \bar{y}) - \hat{\beta}_{yw}(x_i - \bar{x})(w_i - \bar{w}) + \hat{\beta}_{xw}\hat{\beta}_{yw}(w_i - \bar{w})^2\right]$$

$$= \sum\left[(x_i - \bar{x})(y_i - \bar{y})\right] - \hat{\beta}_{xw}\sum\left[(y_i - \bar{y})(w_i - \bar{w})\right]$$
$$- \hat{\beta}_{yw}\sum\left[(x_i - \bar{x})(w_i - \bar{w})\right] + \hat{\beta}_{xw}\hat{\beta}_{yw}\sum\left[(w_i - \bar{w})^2\right]$$

$$= \sum\left[(x_i - \bar{x})(y_i - \bar{y})\right] - \frac{\sum(w_i - \bar{w})(x_i - \bar{x})}{\sum(w_i - \bar{w})^2}\sum\left[(y_i - \bar{y})(w_i - \bar{w})\right]$$
$$- \frac{\sum(w_i - \bar{w})(y_i - \bar{y})}{\sum(w_i - \bar{w})^2}\sum\left[(x_i - \bar{x})(w_i - \bar{w})\right]$$
$$+ \frac{\sum(w_i - \bar{w})(y_i - \bar{y})}{\sum(w_i - \bar{w})^2}\;\frac{\sum(w_i - \bar{w})(x_i - \bar{x})}{\sum(w_i - \bar{w})^2}\sum\left[(w_i - \bar{w})^2\right]$$

$$\text{numerator} = \sum\left[(x_i - \bar{x})(y_i - \bar{y})\right] - \boxed{\frac{\sum(w_i - \bar{w})(x_i - \bar{x})\;\sum(w_i - \bar{w})(y_i - \bar{y})}{\sum(w_i - \bar{w})^2}}$$
$$- \frac{\sum(w_i - \bar{w})(x_i - \bar{x})\;\sum(w_i - \bar{w})(y_i - \bar{y})}{\sum(w_i - \bar{w})^2} + \boxed{\frac{\sum(w_i - \bar{w})(x_i - \bar{x})\;\sum(w_i - \bar{w})(y_i - \bar{y})}{\sum(w_i - \bar{w})^2}}$$

0

$$\text{numerator} = \sum\left[(x_i-\bar{x})(y_i-\bar{y})\right] - \frac{\sum(w_i-\bar{w})(x_i-\bar{x})\ \sum(w_i-\bar{w})(y_i-\bar{y})}{\sum(w_i-\bar{w})^2}$$

$$= \sum\left[(x_i-\bar{x})(y_i-\bar{y})\right] - \frac{\sum(w_i-\bar{w})(x_i-\bar{x})\ \sum(w_i-\bar{w})(y_i-\bar{y})}{\sum(w_i-\bar{w})^2\ \sum(w_i-\bar{w})^2}\sum(w_i-\bar{w})^2$$

$$= \hat{\rho}_{yx}\cdot\hat{\sigma}_y\cdot\hat{\sigma}_x - \hat{\beta}_{xw}\hat{\beta}_{yw}\sum(w_i-\bar{w})^2$$

④&⑤ $\quad \hat{\rho}_{yw}\hat{\rho}_{xw} = \dfrac{\sum(w_i-\bar{w})^2}{\hat{\sigma}_x\hat{\sigma}_y}\hat{\beta}_{xw}\hat{\beta}_{yw}, \qquad \hat{\rho}_{yx} = \dfrac{\sum(y_i-\bar{y})(x_i-\bar{x})}{\hat{\sigma}_y\hat{\sigma}_x}$

$$\boxed{\text{numerator} = \hat{\rho}_{yx}\cdot\hat{\sigma}_y\cdot\hat{\sigma}_x - \hat{\rho}_{yw}\hat{\rho}_{xw}\hat{\sigma}_y\hat{\sigma}_x}$$

$$\text{denominator} = \sqrt{\sum\left[(x_i-\bar{x})-\hat{\beta}_{xw}(w_i-\bar{w})\right]^2}\sqrt{\sum\left[(y_i-\bar{y})-\hat{\beta}_{yw}(w_i-\bar{w})\right]^2}$$

$$= \sqrt{\sum\left[(x_i-\bar{x})^2-2\hat{\beta}_{xw}(x_i-\bar{x})(w_i-\bar{w})+\hat{\beta}_{xw}^2(w_i-\bar{w})^2\right]}$$
$$\cdot\sqrt{\sum\left[(y_i-\bar{y})^2-2\hat{\beta}_{yw}(y_i-\bar{y})(y_i-\bar{y})+\hat{\beta}_{yw}^2(w_i-\bar{w})^2\right]}$$

$$= \sqrt{\sum(x_i-\bar{x})^2-2\hat{\beta}_{xw}\sum(x_i-\bar{x})(w_i-\bar{w})+\hat{\beta}_{xw}^2\sum(w_i-\bar{w})^2}$$
$$\cdot\sqrt{\sum(y_i-\bar{y})^2-2\hat{\beta}_{yw}\sum(y_i-\bar{y})(w_i-\bar{w})+\hat{\beta}_{yw}\sum(w_i-\bar{w})^2}$$

$$\text{denominator} = \sqrt{\hat{\sigma}_x^2-2\frac{\sum(w_i-\bar{w})(x_i-\bar{x})}{\sum(w_i-\bar{w})^2}\sum(x_i-\bar{x})(w_i-\bar{w})+\left(\frac{\sum(w_i-\bar{w})(x_i-\bar{x})}{\sum(w_i-\bar{w})^2}\right)^2\sum(w_i-\bar{w})^2}$$
$$\cdot\sqrt{\hat{\sigma}_y^2-2\frac{\sum(w_i-\bar{w})(y_i-\bar{y})}{\sum(w_i-\bar{w})^2}\sum(y_i-\bar{y})(w_i-\bar{w})+\left(\frac{\sum(w_i-\bar{w})(y_i-\bar{y})}{\sum(w_i-\bar{w})^2}\right)^2\sum(w_i-\bar{w})^2}$$

$$\text{denominator} = \sqrt{\hat\sigma_x^2 - \frac{\left[\sum(x_i-\bar x)(w_i-\bar w)\right]^2}{\sum(w_i-\bar w)^2}} \cdot \sqrt{\hat\sigma_y^2 - \frac{\left[\sum(y_i-\bar y)(w_i-\bar w)\right]^2}{\sum(w_i-\bar w)^2}}$$

$$= \sqrt{\hat\sigma_x^2 - \frac{\left[\sum(x_i-\bar x)(w_i-\bar w)\right]^2}{\sum(w_i-\bar w)^2}\cdot\frac{\hat\sigma_x^2}{\sum(x_i-\bar x)^2}} \cdot \sqrt{\hat\sigma_y^2 - \frac{\left[\sum(y_i-\bar y)(w_i-\bar w)\right]^2}{\sum(w_i-\bar w)^2}\cdot\frac{\hat\sigma_y^2}{\sum(y_i-\bar y)^2}}$$

$$= \hat\sigma_x\hat\sigma_y\sqrt{1-\frac{\left[\sum(x_i-\bar x)(w_i-\bar w)\right]^2}{\sum(w_i-\bar w)^2\sum(x_i-\bar x)^2}}\sqrt{1-\frac{\left[\sum(y_i-\bar y)(w_i-\bar w)\right]^2}{\sum(w_i-\bar w)^2\sum(y_i-\bar y)^2}}$$

6 & 7

$$\hat\rho_{xw}^2 = \frac{\left[\sum(w_i-\bar w)(x_i-\bar x)\right]^2}{\sum(x_i-\bar x)^2\sum(w_i-\bar w)^2} \qquad \hat\rho_{yw}^2 = \frac{\left[\sum(w_i-\bar w)(y_i-\bar y)\right]^2}{\sum(y_i-\bar y)^2\sum(w_i-\bar w)^2}$$

$$\boxed{\text{denominator} = \hat\sigma_x\hat\sigma_y\sqrt{1-\hat\rho_{xw}^2}\sqrt{1-\hat\rho_{yw}^2}}$$

$$\hat\rho_{yx|w} = \frac{\sum\hat\epsilon_{x_i}\hat\epsilon_{y_i}}{\sqrt{\sum\hat\epsilon_{x_i}^2}\sqrt{\sum\hat\epsilon_{y_i}^2}}$$

$$= \frac{\sum\left(x_i-\bar x-\hat\rho_{xw}(w_i-\bar w)\right)\left(y_i-\bar y-\hat\rho_{yw}(w_i-\bar w)\right)}{\sqrt{\sum\left(x_i-\bar x-\hat\rho_{xw}(w_i-\bar w)\right)^2}\sqrt{\sum\left(y_i-\bar y-\hat\rho_{yw}(w_i-\bar w)\right)}} = \frac{\text{numerator}}{\text{denominator}}$$

$$= \frac{\hat\rho_{yx}\cdot\cancel{\hat\sigma_y\hat\sigma_x} - \hat\rho_{yw}\hat\rho_{xw}\cancel{\hat\sigma_y\hat\sigma_x}}{\cancel{\hat\sigma_x\hat\sigma_y}\sqrt{1-\hat\rho_{xw}^2}\sqrt{1-\hat\rho_{yw}^2}}$$

$$\boxed{\hat\rho_{yx|w} = \frac{\hat\rho_{yx}-\hat\rho_{yw}\hat\rho_{xw}}{\sqrt{(1-\hat\rho_{yw}^2)(1-\hat\rho_{xw}^2)}}} \Longrightarrow$$

For $Y, X, W \in \mathbb{R}^n$, we have

$$\hat\rho_{yx|w} = \frac{\hat\rho_{yx}-\hat\rho_{yw}\hat\rho_{xw}}{\sqrt{1-\hat\rho_{yw}^2}\sqrt{1-\hat\rho_{xw}^2}}.$$

∎

5. Problem 8.6 - Examples of Simpson's Paradox

Give three numerical examples of (Y, X, W) which causes Simpson's Paradox. Report the mean and covariance matrix for each example.

Note: you only need to give $\boxed{\text{one numerical example}}$ and report the mean and covariance matrix

EXAMPLE 1

sorry, my R kernel broke down and I could not run R, so I used python.

SORRY!!!

$$\vec{W} \in \mathbb{R}^n \sim \overrightarrow{Bin(n, p)}$$

$$\vec{h}_1 \in \mathbb{R}^n \sim \overrightarrow{N_{\mathbb{R}^n}(0, 1)}$$

$$\vec{h}_2 \in \mathbb{R}^n \sim 5 + \overrightarrow{N_{\mathbb{R}^n}(0, 1)}$$

$$\vec{X} \in \mathbb{R}^n \sim \vec{W} \cdot \vec{h}_1 + (\vec{1} - \vec{W}) \cdot \vec{h}_2$$

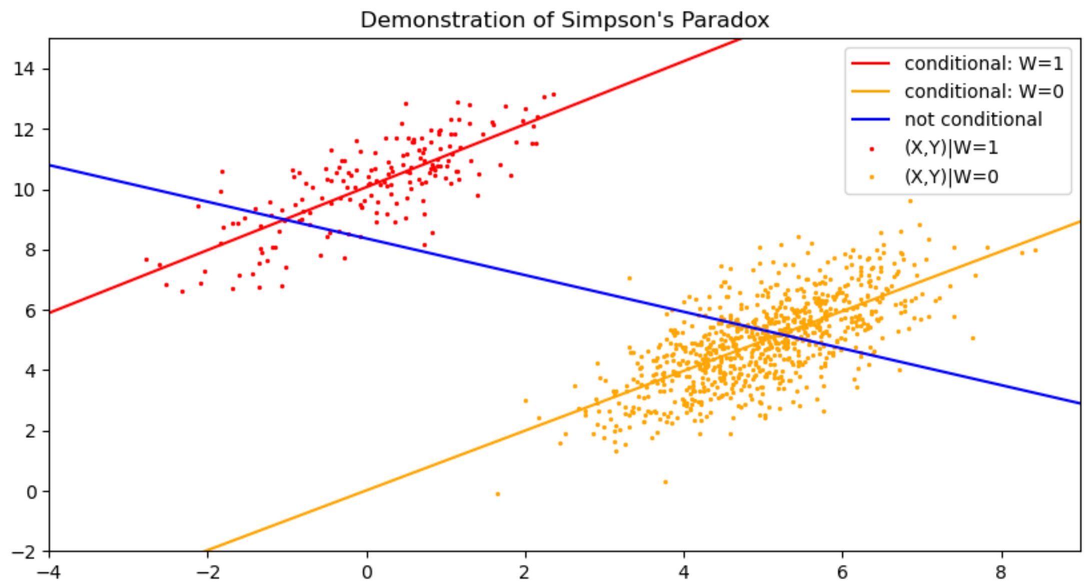
$$\vec{Y} \in \mathbb{R}^n \sim \vec{X} + 10\vec{W} + \overrightarrow{N_{\mathbb{R}^n}(0, 1)}$$

```
mean for all data: (X,Y)
(4.092116058970064, 5.87772014880355)
coviariance matrix for all data
[[ 4.53156547 -2.75446155]
 [-2.75446155  6.10800084]]

mean for data given W==1: (X,Y)
(0.08926614991163011, 10.160435686322744)
coviariance matrix for data given W==1
[[1.16654895 1.21692904]
 [1.21692904 2.13611601]]

mean for data given W==0: (X,Y)
(4.970790429251184, 4.937611860079825)
coviariance matrix for data given W==0
[[0.97804643 0.96895603]
 [0.96895603 2.06753566]]
```

In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

n = 1000
p = 0.2

W = np.random.binomial(1, p, n)
h1 = np.random.normal(0, 1, n)
h2 = 5 + np.random.normal(0, 1, n)

X = W*h1 + (1-W)*h2
Y = X + 10*W + np.random.normal(0, 1, n)

X_given_W = X[W==1]
Y_given_W = Y[W==1]

X_given_not_W = X[W==0]
Y_given_not_W = Y[W==0]

# covariance matrix
mean_all = np.mean(X), np.mean(Y)
print('mean for all data: (X,Y)')
print(mean_all)
cov_all = np.cov(X, Y)
print('coviariance matrix for all data')
print(cov_all)

mean_given_W = np.mean(X_given_W), np.mean(Y_given_W)
print('\nmean for data given W==1: (X,Y)')
print(mean_given_W)
cov_given_W = np.cov(X_given_W, Y_given_W)
print('coviariance matrix for data given W==1')
print(cov_given_W)

mean_given_not_W = np.mean(X_given_not_W), np.mean(Y_given_not_W)
print('\nmean for data given W==0: (X,Y)')
print(mean_given_not_W)
cov_given_not_W = np.cov(X_given_not_W, Y_given_not_W)
print('coviariance matrix for data given W==0')
print(cov_given_not_W)
```

```
mean for all data: (X,Y)
(4.092116058970064, 5.87772014880355)
coviariance matrix for all data
[[ 4.53156547 -2.75446155]
 [-2.75446155  6.10800084]]

mean for data given W==1: (X,Y)
(0.08926614991163011, 10.160435686322744)
coviariance matrix for data given W==1
[[1.16654895 1.21692904]
 [1.21692904 2.13611601]]

mean for data given W==0: (X,Y)
(4.970790429251184, 4.937611860079825)
coviariance matrix for data given W==0
[[0.97804643 0.96895603]
 [0.96895603 2.06753566]]
```

In [ ]:
```
lin_mod_1 = LinearRegression().fit(X_given_W.reshape(-1, 1), Y_given_W)
lin_mod_2 = LinearRegression().fit(X_given_not_W.reshape(-1, 1), Y_given_not_W)
lin_mod_all = LinearRegression().fit(X.reshape(-1, 1), Y)
x_range = np.linspace(-5, 10, 1000)
lin_mod_1_pred = lin_mod_1.predict(x_range.reshape(-1, 1))
lin_mod_2_pred = lin_mod_2.predict(x_range.reshape(-1, 1))
lin_mod_all_pred = lin_mod_all.predict(x_range.reshape(-1, 1))
plt.figure(figsize=(10, 5))
plt.xlim(-4, 9)
plt.ylim(-2, 15)
plt.plot(x_range, lin_mod_1_pred, label='conditional: W=1', color='red')
plt.plot(x_range, lin_mod_2_pred, label='conditional: W=0', color='orange')
plt.plot(x_range, lin_mod_all_pred, label='not conditional', color='blue')
plt.title('Demonstration of Simpson\'s Paradox')
plt.scatter(X_given_W, Y_given_W, color='red', label='(X,Y)|W=1', s=2)
plt.scatter(X_given_not_W, Y_given_not_W, color='orange', label='(X,Y)|W=0', s=2)
plt.legend()
plt.show()
```



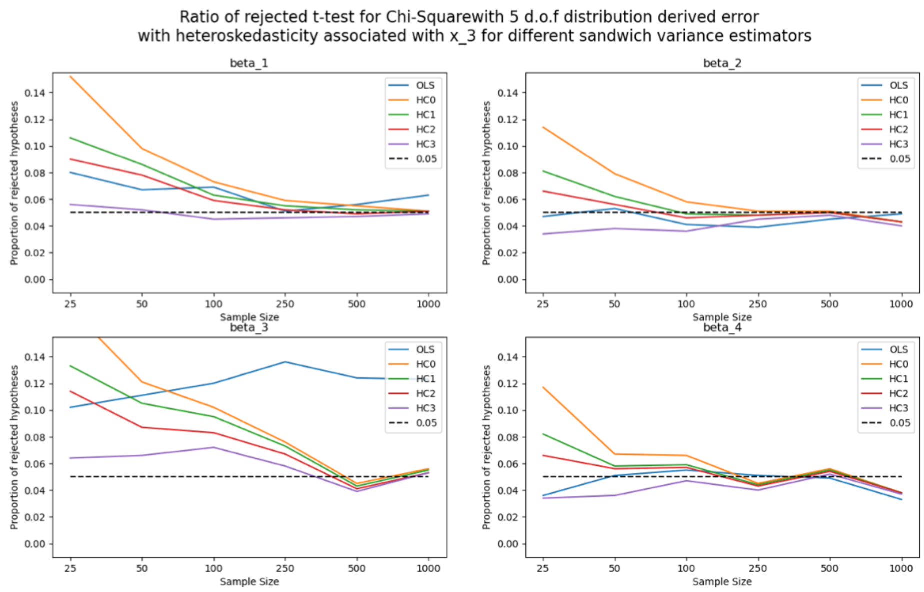
6. A simplified version of Problem 6.5

Empirical comparison of the standard errors Read Long and Ervin (2000). Conduct a Monte Carlo simulation study to explore the finite sample performance of the OLS estimator of the standard error and the four versions of the heteroskedasticity-robust standard error estimators. Follow the data generating process in Long and Ervin (2000) with heteroskedasticity, which is described in Equation (5) with the error structure 2 in Table 2. Replicate this simulation study and replicate their Figure 2. Briefly discuss what you learn from this simulation study.

→ replicated figure 2.

Sorry, my R kernel broke down and I could not run R, so I used python.

SORRY!!!



→ I have learned that when there's heteroskedasticity in data, we really need some kind of sandwich estimator to correct for the variance. Or else, we will be rejecting more than what we intend when hypothesis testing.

In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import statsmodels.api as sm
np.random.seed(0)
```

In [ ]:
```
# generate 100,000 observations
n = 100_000
delta_1 = np.random.uniform(0, 1, n)
delta_2 = np.random.normal(0, 1, n)
delta_3 = np.random.chisquare(1, n)
delta_4 = np.random.normal(0, 1, n)
delta_5 = np.random.uniform(0, 1, n)

# generate the dependent variable
x_1 = 1 + delta_1
x_2 = 3*delta_1 + 0.6*delta_2
x_3 = 2*delta_1 + 0.6*delta_3
x_4 = 0.1*delta_1 + 0.9*delta_3 - 0.8*delta_4 + 4*delta_5
x_D = np.where(x_2 > 1.6, 1, 0)
```

In [ ]:
```
# make sure the data is not too different from the paper
x_1_stats = [x_1.min(), x_1.max(), x_1.mean(), x_1.std()]
x_2_stats = [x_2.min(), x_2.max(), x_2.mean(), x_2.std()]
x_3_stats = [x_3.min(), x_3.max(), x_3.mean(), x_3.std()]
x_4_stats = [x_4.min(), x_4.max(), x_4.mean(), x_4.std()]
x_D_stats = [x_D.min(), x_D.max(), x_D.mean(), x_D.std()]

stats_df = pd.DataFrame([x_1_stats, x_2_stats, x_3_stats, x_4_stats, x_D_stats],
                        columns=['min', 'max', 'mean', 'std'],
                        index=['x_1', 'x_2', 'x_3', 'x_4', 'x_D'])
stats_df
```

Out[ ]:

	min	max	mean	std
x_1	1.000003	1.999978	1.499500	0.289036
x_2	-2.052631	5.233223	1.504764	1.054939
x_3	0.001704	13.660338	1.599156	1.029669
x_4	-2.309826	22.455186	2.950662	1.894655
x_D	0.000000	1.000000	0.467860	0.498966

In [ ]:
```
# measure their corrolations, again, make sure they are not too different from the paper
corr_df = pd.DataFrame(np.corrcoef([x_1, x_2, x_3, x_4, x_D]),
                        columns=['x_1', 'x_2', 'x_3', 'x_4', 'x_D'],
                          index=['x_1', 'x_2', 'x_3', 'x_4', 'x_D'])
# show only the lower triangle, since the upper triangle is the same
mask = np.tril(np.ones_like(corr_df, dtype=bool))
corr_df = corr_df.mask(~mask)
corr_df
```

Out[ ]:

	x_1	x_2	x_3	x_4	x_D
x_1	1.000000	NaN	NaN	NaN	NaN
x_2	0.823291	1.000000	NaN	NaN	NaN
x_3	0.560612	0.462098	1.000000	NaN	NaN
x_4	0.014476	0.013202	0.565141	1.00000	NaN
x_D	0.728986	0.825077	0.407091	0.01198	1.0

In [ ]:
```
# generate the error terms
epsilon_Ni = np.random.normal(0, 1, n)
epsilon_Xi = np.random.chisquare(5, n)
epsilon_ti = np.random.standard_t(5, n)

# 6 types of error structures from the paper
Error_Structure_1N = np.sqrt(x_1) * epsilon_Ni
Error_Structure_1X = np.sqrt(x_1) * epsilon_Xi

Error_Structure_3N = np.sqrt(x_3 + 1.6) * epsilon_Ni
Error_Structure_3X = np.sqrt(x_3 + 1.6) * epsilon_Xi

Error_Structure_34N = np.sqrt(x_3) * np.sqrt(x_4 + 2.5) * epsilon_Ni
Error_Structure_34X = np.sqrt(x_3) * np.sqrt(x_4 + 2.5) * epsilon_Xi

Error_Structure_123N = np.sqrt(x_1) * np.sqrt(x_2 + 2.5) * np.sqrt(x_3) * epsilon_Ni
Error_Structure_123X = np.sqrt(x_1) * np.sqrt(x_2 + 2.5) * np.sqrt(x_3) * epsilon_Xi

Error_Structure_DsmN = np.where(x_D == 1, 1.5 * epsilon_Ni, epsilon_Ni)
Error_Structure_DbigN = np.where(x_D == 1, 4 * epsilon_Ni, 0.5 * epsilon_Ni)

Error_Structure_DsmX = np.where(x_D == 1, 1.5 * epsilon_Xi, epsilon_Xi)
Error_Structure_DbigX = np.where(x_D == 1, 4 * epsilon_Xi, 0.5 * epsilon_Xi)
```

In [ ]:
```python
# error structures asiciated with x_3
Error_Structure_List = [Error_Structure_3X, Error_Structure_34X, Error_Structure_123X]
# number of samples to sameple from the population
N_list = [25, 50, 100, 250, 500, 1000]
# correction methods for the sandwich estimator
HC_list = ['HC0', 'HC1', 'HC2', 'HC3']
# tau chosen such that R^2 in the population was approximately 0.4, this works
tau = 0.8

# store the results
rejected_all_all_all = np.zeros((len(Error_Structure_List), len(HC_list)+1, len(N_list), 4))
for ES_idx, Error_Struture in enumerate(Error_Structure_List):
    rejected_all_all = np.zeros((len(HC_list)+1, len(N_list), 4))
    # generate the dependent variable
    y = 1 + x_1 + x_2 + x_3 + 0 * x_4 + tau * Error_Struture
    # estimate the population parameters by OLS
    X = np.column_stack((x_1, x_2, x_3, x_4))
    X = sm.add_constant(X)
    model = sm.OLS(y, X)
    results = model.fit()
    # these are the population parameters to be compared to as H0
    beta_1_star = results.params[1]
    beta_2_star = results.params[2]
    beta_3_star = results.params[3]
    beta_4_star = results.params[4]
    # print R^2 to make sure it is approximately 0.4
    print('Error Structure ', ES_idx, ', with:')
    print('R^2 = ', results.rsquared)
    for sample_idx in range(1000):
        rejected_all = np.zeros((len(HC_list)+1, len(N_list), 4))
        for N_idx, N in enumerate(N_list):
            rejected = np.zeros((len(HC_list)+1, len(N_list), 4))
            sample = np.random.choice(n, N, replace=False)
            y_sample = y[sample]
            X_sample = X[sample]
            # estimate the parameters by OLS, ie, no correction for heteroskedasticity
            model_og = sm.OLS(y_sample, X_sample)
            results_og = model_og.fit()
            beta_1_og = results_og.conf_int()[1]
            beta_2_og = results_og.conf_int()[2]
            beta_3_og = results_og.conf_int()[3]
            beta_4_og = results_og.conf_int()[4]
            # record the instances where the confidence intervals do not contain beta_star
            if beta_1_og[0] > beta_1_star or beta_1_og[1] < beta_1_star:
                rejected[0, N_idx, 0] += 1
            if beta_2_og[0] > beta_2_star or beta_2_og[1] < beta_2_star:
                rejected[0, N_idx, 1] += 1
            if beta_3_og[0] > beta_3_star or beta_3_og[1] < beta_3_star:
                rejected[0, N_idx, 2] += 1
            if beta_4_og[0] > beta_4_star or beta_4_og[1] < beta_4_star:
                rejected[0, N_idx, 3] += 1
            # now do the same for the other correction methods for heteroskedasticity
            for HC_idx, HC in enumerate(HC_list):
                model = sm.OLS(y_sample, X_sample)
                results = model.fit(cov_type=HC)
                beta_1_ci = results.conf_int()[1]
                beta_2_ci = results.conf_int()[2]
                beta_3_ci = results.conf_int()[3]
                beta_4_ci = results.conf_int()[4]
                # same here, record the rejected instances
                if beta_1_ci[0] > beta_1_star or beta_1_ci[1] < beta_1_star:
                    rejected[HC_idx+1, N_idx, 0] += 1
                if beta_2_ci[0] > beta_2_star or beta_2_ci[1] < beta_2_star:
                    rejected[HC_idx+1, N_idx, 1] += 1
                if beta_3_ci[0] > beta_3_star or beta_3_ci[1] < beta_3_star:
                    rejected[HC_idx+1, N_idx, 2] += 1
                if beta_4_ci[0] > beta_4_star or beta_4_ci[1] < beta_4_star:
                    rejected[HC_idx+1, N_idx, 3] += 1
            rejected_all += rejected
        rejected_all_all += rejected_all
    # divide by 1000 to get the proportion of rejected instances
    rejected_all_all /= 1000
    rejected_all_all_all[ES_idx] = rejected_all_all
# average over the error structures
rejected_all = rejected_all_all_all.mean(axis=0)
```

```
Error Structure  0 , with:
R^2 =  0.30655517325142034
Error Structure  1 , with:
R^2 =  0.43644960957900514
Error Structure  2 , with:
R^2 =  0.4485023393350809
```

In [ ]:
```python
# rejection rate data from beta_1, beta_2, beta_3, beta_4
b1_data = rejected_all_all[:, :, 0]
b2_data = rejected_all_all[:, :, 1]
b3_data = rejected_all_all[:, :, 2]
b4_data = rejected_all_all[:, :, 3]
```

In [ ]:
```python
# convert N_list to string for plotting
N_list = [str(N) for N in N_list]
# plot results: beta_1, beta_2, beta_3, beta_4 into 4 subplots
fig, axs = plt.subplots(2, 2, figsize=(16, 9))
fig.suptitle('Ratio of rejected t-test for Chi-Squarewith 5 d.o.f distribution derived error ' +
             '\n with heteroskedasticity associated with x_3 for different sandwich variance estimators', fon
axs[0, 0].plot(N_list, b1_data[0], label='OLS')
axs[0, 0].plot(N_list, b1_data[1], label='HC0')
axs[0, 0].plot(N_list, b1_data[2], label='HC1')
axs[0, 0].plot(N_list, b1_data[3], label='HC2')
axs[0, 0].plot(N_list, b1_data[4], label='HC3')
axs[0, 0].plot(N_list, [0.05]*len(N_list), '--', color='black', label='0.05')
axs[0, 0].set_title('beta_1')
axs[0, 0].set_xlabel('Sample Size')
axs[0, 0].set_ylabel('Proportion of rejected hypotheses')
axs[0, 0].set_ylim(-0.01, 0.155)
axs[0, 0].legend()
axs[0, 1].plot(N_list, b2_data[0], label='OLS')
axs[0, 1].plot(N_list, b2_data[1], label='HC0')
axs[0, 1].plot(N_list, b2_data[2], label='HC1')
axs[0, 1].plot(N_list, b2_data[3], label='HC2')
axs[0, 1].plot(N_list, b2_data[4], label='HC3')
axs[0, 1].plot(N_list, [0.05]*len(N_list), '--', color='black', label='0.05')
axs[0, 1].set_title('beta_2')
axs[0, 1].set_xlabel('Sample Size')
axs[0, 1].set_ylabel('Proportion of rejected hypotheses')
axs[0, 1].set_ylim(-0.01, 0.155)
axs[0, 1].legend()
axs[1, 0].plot(N_list, b3_data[0], label='OLS')
axs[1, 0].plot(N_list, b3_data[1], label='HC0')
axs[1, 0].plot(N_list, b3_data[2], label='HC1')
axs[1, 0].plot(N_list, b3_data[3], label='HC2')
axs[1, 0].plot(N_list, b3_data[4], label='HC3')
axs[1, 0].plot(N_list, [0.05]*len(N_list), '--', color='black', label='0.05')
axs[1, 0].set_title('beta_3')
axs[1, 0].set_xlabel('Sample Size')
axs[1, 0].set_ylabel('Proportion of rejected hypotheses')
axs[1, 0].set_ylim(-0.01, 0.155)
axs[1, 0].legend()
axs[1, 1].plot(N_list, b4_data[0], label='OLS')
axs[1, 1].plot(N_list, b4_data[1], label='HC0')
axs[1, 1].plot(N_list, b4_data[2], label='HC1')
axs[1, 1].plot(N_list, b4_data[3], label='HC2')
axs[1, 1].plot(N_list, b4_data[4], label='HC3')
axs[1, 1].plot(N_list, [0.05]*len(N_list), '--', color='black', label='0.05')
axs[1, 1].set_title('beta_4')
axs[1, 1].set_xlabel('Sample Size')
axs[1, 1].set_ylabel('Proportion of rejected hypotheses')
axs[1, 1].set_ylim(-0.01, 0.155)
axs[1, 1].legend()
plt.show()
```