Computer Programs for the Numerical Results in "On Optimal Inference in the Linear IV Model" by Andrews, Marmer and Yu

1. Matlab code **Prob_infty_CI.m** simulates the probabilities of infinite-length confidence intervals for different tests: AR, CLR, POIS_∞, POIS2_∞ and CLR2_n, where $n = 100, 500, 1000$. It produces results in Table SM-I (for AR, CLR, POIS_∞, POIS2_∞) and Table SM-VI (for CLR2_n and CLR). Table I is a summary of Table SM-I. **Prob_infty_CI.m** currently sets the value of rho (corresponding to ρ_{uv} in the paper) as 0. Results for $\rho_{uv} = .3, .5, .7$, and $.9$ can be obtained by changing the input of rho. **Prob_infty_CI.m** calls user-defined functions **CLR_pv.m** and **BD_pv.m** to calculate the p-values for the CLR and POIS_∞ by numerical integration.

2. Matlab code **Power_comp_max_beta0.m** computes maximum power differences between POIS2 and CLR tests over β_0 values with $|\beta_0| \in \{.25, .5, ..., 3.75, 4, 5, 7.5, 10, 50, 100, 1000, 10000\}$. The current code sets $\rho_{uv} = 0$ and $\lambda = 10$. Results for other (ρ_{uv}, λ)'s can be obtained by changing the inputs of rho and con (corresponding to λ in the paper).

   Power_comp_max_beta0.m generates Table SM-II. Table II is a summary of Table SM-II.

   Matlab code **Power_comp_max_beta0.m** calls the user-defined function **CLR_pv.m**.

   Matlab code **Power_comp_max_beta0_AR_LM.m** adds powers for AR and LM tests. It produces results that are summarized in Table SM-III and Table SM-IV.

3. Matlab codes **WAP.m** and **WAP_Hrho.m** produce results that are summarized in Table III, Table IV and Table SM-V.

   WAP.m simulates the average (over λ) power differences for $\lambda \in \{2.5, 5.0, ..., 90.0\}$ between the WAP2 and CLR tests, for $\rho_{uv} = .0, .3, .5, .7, .9$. **WAP_Hrho.m** simulates the same average power differences for $\rho_{uv} = .95, .99$. For very large $\rho_{uv} = .95, .99$, the calculation of $\psi(Q;\beta_0,\beta,\lambda)$ and $\psi_2(Q_T;\beta_0,\beta,\lambda)$ in the WAP2 test [1] needs extra steps. This is because large Q_T associated with large ρ_{uv} sometimes causes the modified Bessel functions of the first kind $I_{(k-2)/2}(\sqrt{\lambda\xi_\beta(Q)})$ and $I_{(k-2)/2}(\sqrt{\lambda d_\beta^2 Q_T})$ to exceed the maximum real number allowed in Matlab. To deal with this, we first approximate the Bessel function $I_{(k-2)/2}(x)$ by $e^x/\sqrt{2pi \cdot x}$ (see AMS06, p728), so that we can combine the $I_{(k-2)/2}(\cdot)$ term with the first multiplier on the right hand side of ψ and ψ_2. We then divide the denominator and numerator of WAP2

[1] The WAP2(Q,β_0,β_*) test of the form $\sum_{j=1}^J(\psi(Q;\beta_0,\beta_{*j},\lambda_j) + \psi(Q;\beta_0,\beta_{2*j},\lambda_{2j}))/\sum_{j=1}^J 2\psi_2(Q_T;\beta_0,\beta_*,\lambda_j)$, where $\psi(Q;\beta_0,\beta,\lambda) = \exp(-\lambda(c_\beta^2+d_\beta^2)/2)(\lambda\xi_\beta(Q))^{-(k-2)/4}I_{(k-2)/2}(\sqrt{\lambda\xi_\beta(Q)})$ and $\psi_2(Q_T;\beta_0,\beta,\lambda) = \exp(-\lambda d_\beta^2/2)(\lambda d_\beta^2 Q_T)^{-(k-2)/4}I_{(k-2)/2}(\sqrt{\lambda d_\beta^2 Q_T})$.

by a number. These extra steps are conducted by calling user-defined functions **psi_a.m**, **psi2_a.m**, **psi_h.m** and **psi2_h.m**.

WAP.m and **WAP_Hrho.m** currently sets $k = 5$. Results for $k = 2, 10, 20, 40$ can be obtained by changing the input of k.

Both **WAP.m** and **WAP_Hrho.m** call the user-defined function **CLR_pv.m**.

4. Matlab code **plots_power.m** generates Figure I. It uses (some of) outputs from Matlab codes

   Power_comp_max_beta0.m and **Power_comp_max_beta0_AR_LM.m**, which are provided as **output_XXX.mat** in the corresponding folder. **plots_power.m** is currently set for the figure with $\rho_{uv} = 0$ (Figure I(a)). Figures for $\rho_{uv} = .5$ and .9 can be obtained by changing the input data **output_XXX.mat**.

5. Matlab code **graphs_AMS.m** generates Figure SM-I, which provides graphs that are the same as in AMS06, but with $\rho_\Omega = 0$, rather than $\rho_\Omega = .5$ or .95. The current code sets $(k, \lambda) = (5, 20)$, which can be changed to other values. **graphs_AMS.m** calls the user-defined function **CLR_pv.m**.