```
* NOTES:
* My code fricking CRAAAWLS. It is extremely slow. I guess trying to jerry rig how
* I did this in R into the weird world of stata was not a great idea.


clear all
set more off, perm

global dir "c:\Users\Nmath_000\Documents\Code\courses\econ 675\PS_2_tex\"

cap log close
log using $pset2 stata log.smcl, replace
**********************************
*********** Question 1 ************
**********************************

 global hvalues .5 .6 .7 .8 0.8199 .9 1 1.1 1.2 1.3 1.4 1.5
* global hvalues .5
local h = .5
local i = 1
global n = 1000

* I need to only do 10 simulations because of how slow this thing is
global m = 10
set obs $n

* replace with for loop eventually
forvalues i = 1/10{
di `i'
* start loop
clear
set obs $n
* generate random data
gen z_o = uniform()
gen xi = rnormal(-1.5, sqrt(1.5)) if z_o < .5
replace xi = rnormal(1,1) if z_o >= .5

* drop zero one var
drop z_o

* gen constaant for merge
gen const = 1

* save as temp file for merge
tempfile rand_i
 save "`rand_i'"


 * rename variable
 rename xi x

* try merging this with teacher level enr_staff file
joinby const using `rand_i'

* now loop over h values
foreach h in $hvalues {

    di `h'
    * make h for file names
    local h_n: subinstr local h "." "", all

    *preserve data before I mess with is
    preserve

    * gnerate u
    gen u =  (xi-x )/`h'

    * calculate kernal for pairs
    gen kern = (.75*(1-u^2)*(abs(u)<=1))/`h'

```

```
71
72      * get means
73      replace x = round(x,.00001)
74      bys x: egen fhats = mean(kern)
75      egen tag = tag(x)
76      keep if tag == 1
77      drop xi const u kern tag
78
79      * add in f_x
80      gen f_x = .5*normalden(x, -1.5, sqrt(1.5)) + .5*normalden(x, 1, 1)
81
82      * find sq error
83      gen sq_er = (fhats-f_x)^2
84
85      * now get  imse_li
86      egen imse_li = mean(sq_er)
87      egen tag2 = tag(imse_li)
88      keep if tag2 == 1
89
90      keep imse_li
91
92      * fill in sum info
93      gen sim = `i'
94      gen h = `h'
95
96
97      * save temp data
98      tempfile imseli_`h_n'_`i'
99       save "imseli_`h_n'_`i'", replace
100
101     * restore data, preserve it for next thing
102     restore
103
104     preserve
105     * now do the leave on out, drop columns with the same x xi
106     * this is bad coding but STATA is terrible so this is what it deserves
107     keep if x != xi
108
109     * gnerate u
110     gen u =  (xi-x )/`h'
111
112     * calculate kernal for pairs
113     gen kern = (.75*(1-u^2)*(abs(u)<=1))/`h'
114
115     * collaps data to get means \* collapse data
116     replace x = round(x,.00001)
117     bys x: egen fhats = mean(kern)
118     egen tag = tag(x)
119     keep if tag == 1
120     drop xi const u kern tag
121
122     * add in f_x
123     gen f_x = .5*normalden(x, -1.5, sqrt(1.5)) + .5*normalden(x, 1, 1)
124
125     * find sq error
126     gen sq_er = (fhats-f_x)^2
127
128     * now get  imse_li
129     egen imse_lo = mean(sq_er)
130     egen tag2 = tag(imse_lo)
131     keep if tag2 == 1
132
133     keep imse_lo
134
135
136     * fill in sum info
137     gen sim = `i'
138     gen h = `h'
139
140
```

```
        * save temp data
        tempfile imselo_`h_n'_`i'
        quietly save "imselo_`h_n'_`i'" , replace

        * restore data for next h
        restore

    }
    }
    * now, because I dont think stata has lists we just load all that back in and stack it
    * clear out data
    clear

    forvalues i = 1/$m{
    foreach h in $hvalues {

        * make h for file names
        local h n: subinstr local h "." "", all

        append using "imseli_`h_n'_`i'.dta"
        append using "imselo_`h_n'_`i'.dta"

    }
    }

    * Now collapse data to get mean leave on in and out across iteratiosn by h
        bys h: egen m_imse_li = mean(imse_li)
        bys h: egen m_imse_lo = mean(imse_lo)
        egen tag = tag(h)
        keep if tag == 1
        keep h m_imse_li m_imse_lo


    * graph this stuff
    line m_imse_li m_imse_lo h

    graph export "$dir\stata plot 1 3 b.png", replace

    dataout, save($dir\stata_table_1_3_b.tex) tex replace


    **************
    **** Problem 2
    **************
    **************
    **** Problem 2.5.b
    ***************
    clear all
    set obs 1000
    * Define cross validation function: CV(list, i): vars=variable list, i = max polynomial
    mata
        void CV(vars, i) {
            st_view(y=., ., "y")
            st_view(X=., ., tokens(vars))
            XpX  = cross(X, X)
            XpXinv  = invsym(XpX)
            b  = XpXinv*cross(X, y)
            w = diagonal(X*XpXinv*X')
            muhat = X*b
            num = (y - muhat):*(y - muhat)
            den= (J(1000,1,1) - w):*(J(1000,1,1) - w)
            div = num:/den
            CV = mean(div)
            CV
            st_numscalar("mCV"+strofreal(i), CV)
        }
    end
    * Program which runs the monte-carlo experiment
    program CVsim, rclass
        drop _all
```

```
211         set obs 1000
212         forvalues i = 0/20 {
213             gen CV`i' = 0
214         }
215         gen x = runiform(-1,1)
216         gen e = x^2*(rchi2(5)-5)
217         gen y = exp(-0.1*(4*x-1)^2)*sin(5*x)+e
218         forvalues i = 0/20 {
219             gen x`i' = x^`i'
220         }
221         forvalues i = 0/20 {
222             global xlist = "x0-x`i'"
223             di "$xlist"
224             mata CV("$xlist", `i')
225             replace CV`i' = mCV`i'
226         }
227     end
228     * Run the experiment
229     set seed 12345
230     simulate CV0=CV0 CV1=CV1 CV2=CV2 CV3=CV3 CV4=CV4 CV5=CV5 CV6=CV6 CV7=CV7 CV8=CV8 ///
231         CV9=CV9 CV10=CV10 CV11=CV11 CV12=CV12 CV13=CV13 CV14=CV14 CV15=CV15 ///
232         CV16=CV16 CV17=CV17 CV18=CV18 CV19=CV19 CV20=CV20, reps(100) nodots: CVsim
233     collapse *
234     gen i = 1
235     reshape long CV, i(i) j(k)
236     sort CV
237     local min = k[1]
238     twoway scatter CV k, ytitle("Mean CV") xtitle("K") xlabel(0(2)20) xmtick(0(1)20) xline(`min'
        ) title("Average CV(K), across 1000 simulations")
239     graph export "$dir\stata_plot_2_5_b.png", replace
240
241     ****************
242     ***Problem 2.5.c
243     ****************
244
245     * Program which runs the monte-carlo experiment for mu_0
246     program muhatsim, rclass
247         drop _all
248         set obs 1000
249         gen x = runiform(-1,1)
250         gen e = x^2*(rchi2(5)-5)
251         gen y = exp(-0.1*(4*x-1)^2)*sin(5*x)+e
252         forvalues p = 0/7 {
253             gen x`p' = x^`p'
254         }
255         reg y x0-x7, nocons
256         clear
257         set obs 11
258         gen n = _n
259         gen foo = 1
260         gen x = -1+(_n-1)/5
261         forvalues p = 0/7 {
262             gen x`p' = x^`p'
263         }
264         predict muhat
265         predict se, stdp
266         generate lb = muhat - invnormal(0.975)*se
267         generate ub = muhat + invnormal(0.975)*se
268
269
270         keep n muhat foo lb ub
271         reshape wide muhat lb ub, i(foo) j(n)
272     end
273     set seed 12345
274     simulate muhat1=muhat1 muhat2=muhat2 muhat3=muhat3 muhat4=muhat4 muhat5=muhat5 ///
275         muhat6=muhat6 muhat7=muhat7 muhat8=muhat8 muhat9=muhat9 muhat10=muhat10 muhat11=muhat11
        ///
276         ub1=ub1 ub2=ub2 ub3=ub3 ub4=ub4 ub5=ub5 ub6=ub6 ub7=ub7 ub8=ub8 ub9=ub9 ub10=ub10 ub11=
        ub11 ///
277         lb1=lb1 lb2=lb2 lb3=lb3 lb4=lb4 lb5=lb5 lb6=lb6 lb7=lb7 lb8=lb8 lb9=lb9 lb10=lb10 lb11=
```

```
    lb11, reps(1000) nodots: muhatsim
gen i = _n
reshape long muhat ub lb, i(i) j(grid)
collapse muhat ub lb, by(grid)
gen x = -1+ (grid-1)/5
twoway (function y = exp(-0.1*(4*x-1)^2)*sin(5*x), range(-1 1) lcolor(red)) ///
    (line muhat x, lcolor(gs6)) (line lb x, lcolor(gs6) lpattern(dash)) (line ub x, lcolor(gs6) lpattern(dash)), ///
    legend(order(1 "DGP" 2 "Prediction" 3 "Confidence Interval") rows(1)) ytitle(Y) xtitle(X) title("Mu_hat(x) across 1000 simulations")
graph export "$dir\stata_plot_2_5_c.png", replace


***************
* poblem 2.5.d
****************


* Program which runs the monte-carlo experiment for mu_1
program dmuhatsim, rclass
    drop _all
    set obs 1000
    gen x = runiform(-1,1)
    gen e = x^2*(rchi2(5)-5)
    gen y = exp(-0.1*(4*x-1)^2)*((0.8-3.2*x)*sin(5*x)+5*cos(5*x)) + e
    forvalues p = 0/7 {
        gen x`p' = x^`p'
    }
    reg y x0-x7, nocons
    clear
    set obs 11
    gen n = _n
    gen foo = 1
    gen x = -1+(_n-1)/5
    forvalues p = 0/7 {
        gen x`p' = x^`p'
    }
    predict dmuhat
    predict se, stdp
    generate lb = dmuhat - invnormal(0.975)*se
    generate ub = dmuhat + invnormal(0.975)*se


    keep n dmuhat foo lb ub
    reshape wide dmuhat lb ub, i(foo) j(n)
end
set seed 12345
simulate dmuhat1=dmuhat1 dmuhat2=dmuhat2 dmuhat3=dmuhat3 dmuhat4=dmuhat4 dmuhat5=dmuhat5 ///
    dmuhat6=dmuhat6 dmuhat7=dmuhat7 dmuhat8=dmuhat8 dmuhat9=dmuhat9 dmuhat10=dmuhat10 dmuhat11=dmuhat11 ///
    ub1=ub1 ub2=ub2 ub3=ub3 ub4=ub4 ub5=ub5 ub6=ub6 ub7=ub7 ub8=ub8 ub9=ub9 ub10=ub10 ub11=ub11 ///
    lb1=lb1 lb2=lb2 lb3=lb3 lb4=lb4 lb5=lb5 lb6=lb6 lb7=lb7 lb8=lb8 lb9=lb9 lb10=lb10 lb11=lb11, reps(1000) nodots: dmuhatsim
gen i = _n
reshape long dmuhat ub lb, i(i) j(grid)
collapse dmuhat ub lb, by(grid)
gen x = -1+ (grid-1)/5
twoway (function y = exp(-0.1*(4*x-1)^2)*((0.8-3.2*x)*sin(5*x)+5*cos(5*x)), range(-1 1) lcolor(red)) ///
    (line dmuhat x, lcolor(gs6)) (line lb x, lcolor(gs6) lpattern(dash)) (line ub x, lcolor(gs6) lpattern(dash)), ///
    legend(order(1 "DGP" 2 "Prediction" 3 "Confidence Interval") rows(1)) ytitle(Y) xtitle(X) title("(d/dx)*Mu_hat(x) across 1000 simulations")
graph export "$dir\stata_plot_2_5_d.png", replace
```

```
339
340   ****************************
341   ******** Question 3 ********
342   ****************************
343   * DGP
344   clear all
345   drop _all
346   local theta = 1
347   local d = 5
348   local n = 500
349
350   set obs 1000
351
352   forvalues p = 1/14 {
353   gen se_hat`p' = .
354   gen theta_hat`p' = .
355
356   }
357   mata:
358   void polyloop(i) {
359
360   X   = uniform(`n',`d'):*2 :-1
361   ep  = invnormal(uniform(`n',1)):*0.3637899:*(1 :+ rowsum(X:^2))
362   gx  = exp(rowsum(X:^2))
363   T   = invnormal(uniform(`n',1)) + rowsum(X:^2):^.5 :>= 0
364   Y   = T + gx + ep
365   cons= J(500,1,1)
366
367   /*Raising to single powers */
368   X2  = X:^2
369   X3  = X:^3
370   X4  = X:^4
371   X5  = X:^5
372   X6  = X:^6
373   X7  = X:^7
374   X8  = X:^8
375   X9  = X:^9
376   X10 = X:^10
377
378   /*Kronekering, but this creates some duplicates*/
379   X1k = X#X
380   X2k = X2#X2
381   X3k = X3#X3
382   X4k = X4#X4
383
384   /* Manually removing duplicates...might be a better way to do this */
385   X1k = X1k[1::`n',2::5], X1k[1::`n', 8::10], X1k[1::`n',14::15], X1k[1::`n', 20]
386   X2k = X2k[1::`n',2::5], X2k[1::`n', 8::10], X2k[1::`n',14::15], X2k[1::`n', 20]
387   X3k = X3k[1::`n',2::5], X3k[1::`n', 8::10], X3k[1::`n',14::15], X3k[1::`n', 20]
388   X4k = X4k[1::`n',2::5], X4k[1::`n', 8::10], X4k[1::`n',14::15], X4k[1::`n', 20]
389
390   A = asarray_create("real",1)
391   asarray(A,1,X)
392   asarray(A,2,(asarray(A,1),X2))
393   asarray(A,3,(asarray(A,2),X1k))
394   asarray(A,4,(asarray(A,3),X3))
395   asarray(A,5,(asarray(A,4),X2k))
396   asarray(A,6,(asarray(A,5),X4))
397   asarray(A,7,(asarray(A,6),X3k))
398   asarray(A,8,(asarray(A,7),X5))
399   asarray(A,9,(asarray(A,8),X4k))
400   asarray(A,10,(asarray(A,9),X6))
401   asarray(A,11,(asarray(A,10),X7))
402   asarray(A,12,(asarray(A,11),X8))
403   asarray(A,13,(asarray(A,12),X9))
404   asarray(A,14,(asarray(A,13),X10))
405   theta_hat = I(1,14):*0
406   se_hat = I(1,14):*0
407   k_hat = I(1,14):*0
408
```

```
409   for (j=1; j<=14; j++) {
410   Z = qrsolve(cons,(T,asarray(A,j)))
411   ZZ  = Z*Z'
412   Yhat = ZZ*Y
413   W = diag(ZZ)
414   ZQ = (cons,asarray(A,j))*invsym((cons,asarray(A,j))'*(cons,asarray(A,j)))*(cons,asarray(A,j
      ))'
415   M = I(`n') - ZQ
416   YM = M*Y
417   TM = M*T
418   theta_hat[1,j] = (TM'*YM) / (TM'*TM)
419   sigma = diag(ZQ*(Y-T*theta hat[1,j]))
420   se_hat[1,j] = sqrt(invsym(T'*ZQ*T)*(T'*ZQ*sigma*ZQ*T)*invsym(T'*ZQ*T))
421   st_store(i, "se_hat"+strofreal(j), se_hat[1,j])
422   st_store(i, "theta_hat"+strofreal(j), theta_hat[1,j])
423   }
424
425   }
426   end
427
428   forvalues i = 1/1000 {
429   mata polyloop(`i')
430   }
431
432   gen theta = _n
433
434   reshape long se_hat theta_hat, i(theta) j(K)
435
436   replace theta = 1
437
438   gen bias = theta_hat - theta
439   gen cov = ((theta_hat - invnormal(.975)*abs(se_hat) <= 1) & (theta_hat + invnormal(.975)*abs
      (se_hat) >= 1))
440
441   collapse se_hat theta_hat bias cov (sd) svar = theta_hat, by(K)
442
443
444   label var se_hat "SE"
445   label var theta_hat "Thetahat"
446   label var bias "Bias"
447   label var cov "Coverage"
448   label var svar "Sample Standard Dev."
449
450   order theta_hat se_hat bias cov svar
451   dataout, save($dir\stata_table_3_4_d.png) tex replace
452
```