

```

1  """
2  FULL REPLICATION SCRIPT
3  Educational Advantage, Savings Deficit: Survey-Weighted Evidence on Gender,
4  Income, and Financial Fragility Among Working-Age Adults in Kazakhstan
5  NHSJS, final version (v3.1, corrected at proof stage)
6
7  Regenerates every figure reported in the manuscript (Tables 1-5 and the
8  in-text estimates) and in the supplementary robustness document
9  (Tables R1-R8) from the raw Kazakhstan country file.
10
11  Run: python3 index_v3_full_replication.py [path/to/Findex_Microdata_2025_updateKazakhstan.csv]
12  Software: Python 3.13, pandas, numpy, statsmodels, scipy, patsy.
13
14  ESTIMATOR NOTE. statsmodels' GLM `freq_weights` argument applies frequency
15  (count) semantics, giving a robust meat matrix of  $\sum(w_i e_i^2 x_i x_i')$ .
16  Probability weights require  $\sum(w_i^2 e_i^2 x_i x_i')$ , because the score
17  contribution of unit  $i$  is  $w_i * s_i$  and therefore squares. `var_weights`
18  supplies the required  $w^2$  meat. The GLM path also returns identical
19  covariances for HC0/HC1/HC2/HC3, so the finite-sample correction is applied
20  explicitly.  $\text{var\_weights} + \text{HC0} + \sqrt{n/(n-1)}$  reproduces Stata's
21  `svyset [pw=wt]` + `svy: logit` to machine precision (max relative
22  difference 3.6e-16; see estimator_verification.py).
23  Only Wald tests are reported: deviance, AIC and likelihood-ratio statistics
24  are not valid likelihood quantities under pseudo-likelihood estimation.
25
26  v3.1 CORRECTION. Weighted Firth fits now use the same linearized (sandwich)
27  variance as the maximum-likelihood fits, evaluated at the penalized
28  estimates:  $\text{bread} = (X'WX)^{-1}$  with  $W = w * \mu * (1 - \mu)$ ,  $\text{meat} = \sum(w_i^2 e_i^2$ 
29   $x_i x_i')$ , times  $n/(n-1)$ . Version 3 used the model-based inverse
30  information for these fits, which is not a design-based variance. The
31  change moves several Firth p-values and intervals and changes no
32  conclusion. Unweighted Firth fits (diagnostic only) keep the model-based
33  variance. The weight-trimming check now trims at the 5th/95th
34  percentiles, because the released weights are already capped: the 1st and
35  99th percentiles equal the minimum and maximum, so trimming there changes
36  nothing.
37  """
38  import sys
39  import pandas as pd, numpy as np, statsmodels.api as sm, statsmodels.formula.api as smf
40  import numpy.linalg as la, patsy
41  from scipy import stats
42  import warnings; warnings.filterwarnings('ignore')
43
44  RNG = np.random.default_rng(20260819)
45  SRC = sys.argv[1] if len(sys.argv) > 1 else 'Findex_Microdata_2025_updateKazakhstan.csv'
46  MBAR = 10 # assumed average cluster size for the sensitivity band
47  SEP = ' '*78
48
49  def hdr(t): print('\n'+SEP+'\n'+t+'\n'+SEP)
50
51  # ----- 1. BUILD
52  raw = pd.read_csv(SRC)
53  d = raw.copy()
54  # Primary outcome. fin24 codes 1-6 name a substantive source; 7 = could not
55  # come up with the money; 8/9 = don't know / refused -> EXCLUDED, not zero.
56  d['vuln'] = np.where(d.fin24.isin([1,2,3,4,5,6]), 0.,
57                      np.where(d.fin24 == 7, 1., np.nan))
58  # Composite robustness outcome: unable OR would find it very difficult.
59  d['fragile'] = np.where(d.fin24 == 7, 1.,
60                        np.where(d.fin24a == 1, 1.,
61                                np.where(d.fin24a.isin([2,3]), 0., np.nan)))
62  d['sav'] = d.fin17a.map({1:1., 2:0., 3:np.nan, 4:np.nan})
63  d['female_d'] = (d.female == 1).astype(int) # codebook: 1 female, 2 male
64  d['digital'] = d.dig_account
65  d['tert'] = np.where(d.educ.isna(), np.nan, (d.educ == 3).astype(float))
66  d['urban'] = (d.urbanicity == 2).astype(int) # 1 rural, 2 urban
67  d['employed'] = (d.emp_in == 1).astype(int)
68  d['ageg'] = pd.cut(d.age, [17,35,55], labels=['18-35', '36-55'])
69  d['old'] = (d.ageg == '36-55').astype(int)
70  d['inc3'] = pd.cut(d.inc_q, [0,2,3,5], labels=['Q1-2', 'Q3', 'Q4-5'])
71
72  A = d[d.ageg.notna() & d.vuln.notna()].copy(); A['w'] = A.wgt/A.wgt.mean()
73  S = d[d.ageg.notna() & d.sav.notna()].copy(); S['w'] = S.wgt/S.wgt.mean()
74  FR = d[d.ageg.notna() & d.fragile.notna()].copy(); FR['w'] = FR.wgt/FR.wgt.mean()
75
76  # ----- 2. TOOLS

```

```

77 def w_prop(y, w):
78     y=np.asarray(y,float); w=np.asarray(w,float)
79     m=~np.isnan(y); y,w=y[m],w[m]; n=len(y)
80     p=(w*y).sum()/w.sum()
81     se=np.sqrt((n/(n-1))*np.sum(w**2*(y-p)**2)/w.sum()**2)
82     if 0<p<1 and se>0: # logit-transformed interval
83         g=np.log(p/(1-p)); sg=se/(p*(1-p))
84         lo,hi=[1/(1+np.exp(-(g+z*1.96*sg))) for z in (-1,1)]
85     else: lo,hi=max(0,p-1.96*se),min(1,p+1.96*se)
86     return dict(p=p,se=se,lo=lo,hi=hi,n=n,num=int(y.sum()),wd=w.sum(),
87               deff=n*np.sum(w**2)/w.sum()**2)
88
89 def w_gap(a,b):
90     dd=a['p']-b['p']; se=np.hypot(a['se'],b['se']); z=dd/se
91     return dict(d=dd,se=se,z=z,p=2*(1-stats.norm.cdf(abs(z))),
92               lo=dd-1.96*se,hi=dd+1.96*se,
93               h=2*np.arcsin(np.sqrt(a['p']))-2*np.arcsin(np.sqrt(b['p'])))
94
95 def _tab(b,se,names):
96     z=b/se
97     return pd.DataFrame({'coef':b,'se':se,'z':z,
98                       'p':2*(1-stats.norm.cdf(np.abs(z))),'OR':np.exp(b),
99                       'lo':np.exp(b-1.96*se),'hi':np.exp(b+1.96*se)},index=names)
100
101 def wml(f,dat):
102     """Pseudo-ML logistic regression with probability weights (see header)."""
103     dat=dat.dropna(subset=[v for v in ['tert','inc3'] if v in f])
104     m=smf.glm(f,dat,family=sm.families.Binomial(),
105              var_weights=dat.w.values).fit(cov_type='HC0')
106     n=int(m.nobs); se=m.bse*np.sqrt(n/(n-1))
107     return _tab(m.params.values,se.values,m.params.index), n
108
109 def sandwich(X,y,w,b):
110     """Linearized (sandwich) variance for probability weights, with n/(n-1)."""
111     n=len(y); mu=1/(1+np.exp(-(X@b)))
112     Ai=la.inv(X.T@(X*(w*mu*(1-mu))[:,None]))
113     B=X.T@(X*((w*(y-mu))**2)[:,None])
114     return np.sqrt(np.diag(Ai@B@Ai)*n/(n-1))
115
116 def firth(X,y,w=None,maxit=400,tol=1e-10):
117     """Firth penalized logistic regression; weights enter score and information.
118     Weighted fits: sandwich variance (design-based). Unweighted: model-based."""
119     weighted = w is not None
120     if w is None: w=np.ones(len(y))
121     b=np.zeros(X.shape[1])
122     for _ in range(maxit):
123         mu=1/(1+np.exp(-(X@b))); V=w*mu*(1-mu); XW=X*V[:,None]
124         Iinv=la.inv(X.T@XW)
125         H=np.einsum('ij,jk,ik->i',XW,Iinv,X)
126         step=Iinv@X.T@(w*(y-mu)+H*(0.5-mu)); b=b+step
127         if np.max(np.abs(step))<tol: break
128     if weighted:
129         se=sandwich(X,y,w,b)
130     else:
131         mu=1/(1+np.exp(-(X@b)))
132         se=np.sqrt(np.diag(la.inv(X.T@(X*(mu*(1-mu))[:,None]))))
133     return b,se
134
135 def firth_tab(f,dat,weighted=True):
136     dat=dat.dropna(subset=[v for v in ['tert','inc3'] if v in f])
137     y,X=patsy.dmatrices(f,dat,return_type='dataframe')
138     b,se=firth(X.values,y.values.ravel(),dat.w.values if weighted else None)
139     return _tab(b,se,X.columns), len(dat)
140
141 def clust(p,rho,m=MBAR):
142     return 2*(1-stats.norm.cdf(stats.norm.ppf(1-p/2)/np.sqrt(1+rho*(m-1))))
143
144 def holm(praw):
145     praw=np.asarray(praw,float); order=np.argsort(praw); k=len(praw)
146     out=np.empty(k); run=0.
147     for i,idx in enumerate(order):
148         run=max(run,(k-i)*praw[idx]); out[idx]=min(1.,run)
149     return out
150
151 def show(df): print(df.round(4).to_string())
152

```

```

153 # self-check: the sandwich() helper reproduces the ML path exactly
154 _y,_X=patsy.dmatrices('vuln ~ female_d*old',A,return_type='dataframe')
155 _m=smf.glm('vuln ~ female_d*old',A,family=sm.families.Binomial(),var_weights=A.w.values).fit(cov_type='HC0'
156 )
157 _s=sandwich(_X.values,_y.values.ravel(),A.w.values,_m.params.values)
158 assert np.allclose(_s,_m.bse.values*np.sqrt(len(A)/(len(A)-1)),rtol=1e-8), 'sandwich check failed'
159
160 # ----- TABLE 1 / R1 audit, EPV
161 hdr('TABLE 1 Outcome recoding and case-exclusion audit')
162 print('fin24 distribution:', raw.fin24.value_counts().sort_index().to_dict())
163 print(f'raw N = {len(raw)}; exclude fin24=8 ({(raw.fin24==8).sum()}) and '
164       f'fin24=9 ({(raw.fin24==9).sum()}); valid outcome = {int(d.vuln.notna().sum())}')
165 print(f'restrict to ages 18-55 -> analytic n = {len(A)}')
166 print('nskip-pattern verification (fin24 x fin24a missing):')
167 print(pd.crosstab(d.fin24, d.fin24a.isna(), dropna=False).to_string())
168 W1855=d[d.ages.notna()]
169 print(f'\nitem missingness within ages 18-55: vulnerability {int(W1855.vuln.isna().sum())}, '
170       f'formal savings {int(W1855.sav.isna().sum())}, education {int(W1855.tert.isna().sum())}')
171
172 # ----- TABLE R1 Events per variable
173 for lab,fr,out,f in [('vulnerability, full',A,'vuln','vuln ~ female_d*old + C(inc_q) + tert + employed +
174                     urban'),
175                     ('vulnerability, reduced',A,'vuln','vuln ~ female_d*old + C(inc3) + tert + employed'),
176                     ('formal savings, full',S,'sav','sav ~ female_d + C(inc_q) + tert + employed + urban
177                     + old')]:
178     sub=fr.dropna(subset=[out]); k=len(patsy.dmatrices(f,sub,return_type='dataframe')[1].columns)
179     ev=int(sub[out].sum()); print(f' {lab:24} n={len(sub)} events={ev:4d} k={k:2d} EPV={ev/k:5.2f}')
180
181 # ----- TABLE 2 characteristics
182 hdr('TABLE 2 Weighted sample characteristics by gender and age group')
183 rows=[]
184 for v,l in [('tert','Tertiary education'),('digital','Digital account'),
185            ('sav','Saved at a financial institution'),('vuln','Financially vulnerable')]:
186     for g in ['18-35','36-55']:
187         for fm,fl in [(1,'Women'),(0,'Men')]:
188             sub=d[(d.ages==g)&(d.female_d==fm)]
189             r=w_prop(sub[v],sub.wgt/sub.wgt.mean())
190             rows.append([l,g,fl,f"{r['num']}/{r['n']}",f"{100*r['p']:.2f}",
191                         f"{100*r['se']:.2f}",f"{100*r['lo']:.2f},{100*r['hi']:.2f}"])
192 print(pd.DataFrame(rows,columns=['Variable','Age','Sex','cases/n','wpct','SE','95% CI']).to_string())
193 tw=w_prop(A[A.female_d==1].tert,A[A.female_d==1].w); tm=w_prop(A[A.female_d==0].tert,A[A.female_d==0].w)
194 g=w_gap(tw,tm); print(f"\nP00LED tertiary gap: {g['d']*100:+.2f} pp 95% CI [{g['lo']*100:.2f},
195                       {g['hi']*100:.2f}] p={g['p']:.4f}")
196 print(f"digitally enabled account, ages 18-55: {100*w_prop(A.digital,A.w)['p']:.1f}%")
197
198 # ----- H1
199 hdr('H1 Financial vulnerability by gender and age group')
200 cells={}
201 for g_ in ['18-35','36-55']:
202     for fm in [1,0]:
203         sub=A[(A.ages==g_)&(A.female_d==fm)]; cells[(g_,fm)]=w_prop(sub.vuln,sub.w)
204         r=cells[(g_,fm)]
205         print(f" {g_} {'W' if fm else 'M'}: {100*r['p']:.2f}% ({r['num']}/{r['n']}) SE {100*r['se']:.2f}
206               DEFF {r['deff']:.2f}")
207 gy=w_gap(cells[(g_,'18-35')],cells[(g_,'36-55')]); go=w_gap(cells[(g_,'36-55')],cells[(g_,'18-35')])
208 dd=gy['d']-go['d']; sedd=np.hypot(gy['se'],go['se'])
209 print(f" gap 18-35 {gy['d']*100:+.2f}pp p={gy['p']:.4f} | gap 36-55 {go['d']*100:+.2f}pp p={go['p']:.4f}")
210 print(f" DIFFERENCE IN GAPS {dd*100:+.2f}pp 95% CI [{(dd-1.96*sedd)*100:.2f},{(dd+1.96*sedd)*100:.2f}]
211       p={2*(1-stats.norm.cdf(abs(dd/sedd))):.4f}")
212 pw=w_prop(A[A.female_d==1].vuln,A[A.female_d==1].w); pm=w_prop(A[A.female_d==0].vuln,A[A.female_d==0].w)
213 gp=w_gap(pw,pm); print(f" P00LED gap {gp['d']*100:+.2f}pp 95% CI [{gp['lo']*100:.2f},{gp['hi']*100:.2f}]
214       p={gp['p']:.4f}")
215 for f,l in [('vuln ~ female_d*old','M1 unadjusted'),
216            ('vuln ~ female_d*old + C(inc_q) + tert + employed + urban','M2 adjusted full')]:
217     t,n=wml(f,A); print(f"\n{l} (n={n})"); show(t)
218 m1,_wml('vuln ~ female_d*old',A); P_H1=float(m1.loc['female_d:old','p'])
219
220 # ----- H2 / TABLE 3 / R5
221 hdr('H2 Formal savings by gender -- every specification (Table 3 / Table R5)')
222 sv={}
223 for g_ in ['18-35','36-55']:
224     a=w_prop(S[(S.ages==g_)&(S.female_d==1)].sav,S[(S.ages==g_)&(S.female_d==1)].w)
225     b=w_prop(S[(S.ages==g_)&(S.female_d==0)].sav,S[(S.ages==g_)&(S.female_d==0)].w)
226     sv[g_]=w_gap(a,b); print(f" {g_}: W {100*a['p']:.2f}% ({a['num']}/{a['n']}) M {100*b['p']:.2f}%
227           ({b['num']}/{b['n']}) gap {sv[g_]['d']*100:+.2f}pp p={sv[g_]['p']:.4f}")
228 dc=sv['18-35']['d']-sv['36-55']['d']; sc=np.hypot(sv['18-35']['se'],sv['36-55']['se'])

```

```

221 print(f"  between-cohort contrast of the two gaps: p={2*(1-stats.norm.cdf(abs(dc/sc)):.4f}")
222 sa=w_prop(S[S.female_d==1].sav,S[S.female_d==1].w); sb=w_prop(S[S.female_d==0].sav,S[S.female_d==0].w)
223 gp2=w_gap(sa,sb)
224 print(f"  P00LED gap {gp2['d']*100:+.2f}pp 95% CI [{gp2['lo']*100:.2f},{gp2['hi']*100:.2f}]
    p={gp2['p']:.4f} h={gp2['h']:.3f}")
225 FULLS='sav ~ female_d + C(inc_q) + tert + employed + urban + old'
226 REDS='sav ~ female_d + C(inc3) + tert + employed + old'
227 for f,l in [(FULLS,'adjusted ML full'),(REDS,'adjusted ML reduced')]:
228     t,n=wml(f,S); r=t.loc['female_d']; print(f"    {l:24} OR={r.OR:.3f} [{r.lo:.3f},{r.hi:.3f}] p={r.p:.4f}
    (n={n})")
229 for f,l in [(FULLS,'adjusted Firth full'),(REDS,'adjusted Firth reduced')]:
230     t,n=firth_tab(f,S); r=t.loc['female_d']; print(f"    {l:24} OR={r.OR:.3f} [{r.lo:.3f},{r.hi:.3f}]
    p={r.p:.4f}")
231 lo_,hi_=np.percentile(S.wgt,[1,99])
232 print(f"  weights already capped: 1st pct = min ({lo_:.4f}), 99th pct = max ({hi_:.4f}); "
    f"int(((S.wgt<lo_)|(S.wgt>hi_)).sum()) weights lie outside")
233 lo_,hi_=np.percentile(S.wgt,[5,95]); T=S.copy(); T['w']=np.clip(T.wgt,lo_,hi_); T['w']=T.w/T.w.mean()
234 t,_=wml(REDS,T); print(f"    {'trimmed 5/95 pct (ML)':24} OR={t.loc['female_d','OR']:.3f}
    p={t.loc['female_d','p']:.4f} ({int(((S.wgt<lo_)|(S.wgt>hi_)).sum())} weights trimmed)")
235 pf,_=firth_tab(REDS,S); P_H2=float(pf.loc['female_d','p'])
236 print(f"  clustering (Firth reduced): rho=.01 p={clust(P_H2,.01):.4f} rho=.03 p={clust(P_H2,.03):.4f}
    rho=.06 p={clust(P_H2,.06):.4f}")
237 for assume,l in [(1,'all excluded = SAVED'),(0,'all excluded = NOT saved')]:
238     B=d[d.ageg.notna()].copy(); B['sav']=np.where(B.fin17a.isin([3,4]),assume,B.sav)
239     B=B[B.sav.notna()].copy(); B['w']=B.wgt/B.wgt.mean()
240     t,n=wml(REDS,B); r=t.loc['female_d']
241     ua=w_prop(B[B.female_d==1].sav,B[B.female_d==1].w); ub=w_prop(B[B.female_d==0].sav,B[B.female_d==0].w)
242     print(f"  Manski, {l:26} OR={r.OR:.3f} [{r.lo:.3f},{r.hi:.3f}] p={r.p:.4f} (unadjusted contrast
    p={w_gap(ua,ub)['p']:.4f}")
243
244
245 # ----- H3 / TABLE R7
246 hdr('H3 Digital account x age among women (Table R7)')
247 WM=A[A.female_d==1].copy()
248 for a,g in [(0,'18-35'),(1,'36-55')]:
249     for dg,dl in [(0,'no digital'),(1,'digital')]:
250         sub=WM[(WM.old==a)&(WM.digital==dg)]; r=w_prop(sub.vuln,sub.w)
251         print(f"  women {g} {dl:11}: {100*r['p']:.2f}% ({r['num']}/{r['n']})")
252     t,n=wml('vuln ~ digital*old',WM); r=t.loc['digital:old']
253     print(f"    {'weighted ML (design-based, unpenalized)':46} OR={r.OR:.3f} [{r.lo:.3f},{r.hi:.3f}]
    p={r.p:.4f} (n={n})")
254     tf,_=firth_tab('vuln ~ digital*old',WM,True); rf=tf.loc['digital:old']; P_H3=float(rf.p)
255     print(f"    {'weighted Firth (design-based, penalized)':46} OR={rf.OR:.3f} [{rf.lo:.3f},{rf.hi:.3f}]
    p={rf.p:.4f}")
256     tu,_=firth_tab('vuln ~ digital*old',WM,False); ru=tu.loc['digital:old']
257     print(f"    {'UNweighted Firth (not a population estimator)':46} OR={ru.OR:.3f} [{ru.lo:.3f},{ru.hi:.3f}]
    p={ru.p:.4f}")
258     mu_=smf.glm('vuln ~ digital*old',WM,family=sm.families.Binomial()).fit()
259     print(f"    {'UNweighted ML (not a population estimator)':46} OR={np.exp(mu_.params['digital:old']):.3f}
    p={mu_.pvalues['digital:old']:.4f}")
260     print("  NOTE: the divergence is driven by the weights, not by the penalty.")
261     FW=FR[FR.female_d==1].copy()
262     print(f"  composite outcome: n={len(FW)} women, events={int(FW.fragile.sum())}")
263     tcf,_=firth_tab('fragile ~ digital*old',FW,True); rc=tcf.loc['digital:old']; P_H3C=float(rc.p)
264     print(f"  composite, weighted Firth OR={rc.OR:.3f} [{rc.lo:.3f},{rc.hi:.3f}] p={rc.p:.4f}")
265     tc,_=wml('fragile ~ digital*old',FW); rcm=tc.loc['digital:old']
266     print(f"  composite, weighted ML OR={rcm.OR:.3f} [{rcm.lo:.3f},{rcm.hi:.3f}] p={rcm.p:.4f}")
267     print(f"  Holm within the three pre-specified hypotheses: p={holm([P_H1,P_H2,P_H3])[2]:.4f}; rho=.06:
    p={clust(P_H3,.06):.4f}")
268     ps=[]
269     for i in range(len(WM)):
270         tt,_=wml('vuln ~ digital*old',WM.drop(WM.index[i])); ps.append(tt.loc['digital:old','p'])
271     print(f"  leave-one-out on unpenalized ML: n={len(ps)} min p={min(ps):.4f} max p={max(ps):.4f}, {sum(1 for
    x in ps if x>.05)} exceed .05")
272 def did(D):
273     c=[]
274     for a in (1,0):
275         x=D[D.old==a]
276         p0=np.average(x.vuln[x.digital==0],weights=x.w[x.digital==0]) if (x.digital==0).any() else np.nan
277         p1=np.average(x.vuln[x.digital==1],weights=x.w[x.digital==1]) if (x.digital==1).any() else np.nan
278         c.append(p0-p1)
279     return c[0]-c[1]
280 est=did(WM); boots=[]
281 for _ in range(2000):
282     bs=[]
283     for a in (0,1):
284         # resample within age group
    x=WM[WM.old==a]; bs.append(x.iloc[RNG.integers(0,len(x),len(x))])

```

```

285     v=did(pd.concat(bs))
286     if np.isfinite(v): boots.append(v)
287     lo_b,hi_b=np.percentile(boots,[2.5,97.5])
288     print(f"    bootstrap difference-in-differences: {100*est:.2f} pp, 95% CI [{100*lo_b:.2f}, {100*hi_b:.2f}]
    ({len(boots)} replicates)")
289
290 # ----- TABLE 4 / R2 / R3 / R4
291 hdr('TABLE 4 Reduced penalized vulnerability model (and Tables R2-R3)')
292 RED='vuln ~ female_d*old + C(inc3) + tert + employed'
293 t4,n=firth_tab(RED,A); print(f'Firth penalized, sandwich variance, n={n}'); show(t4)
294 m3,_=wml(RED,A); print('\nSame specification, weighted ML (Table R3):'); show(m3[['OR','p']])
295 print('\nIncome gradient across estimators (Table R2):')
296 for f,l in [('vuln ~ female_d*old + C(inc_q) + tert + employed + urban','full'),(RED,'reduced')]:
297     ml,_=wml(f,A); fw,_=firth_tab(f,A,True); fu,_=firth_tab(f,A,False)
298     key=[c for c in ml.index if c.startswith('C(inc)')][-1]
299     for tab,est_ in [(ml,'weighted ML'),(fw,'weighted Firth'),(fu,'unweighted Firth')]:
300         r=tab.loc[key]; print(f"    {l:8} {est_:17} {key:22} OR={r.OR:.3f} [{r.lo:.3f},{r.hi:.3f}]
    p={r.p:.4f}")
301
302 hdr('TABLE R4 Education: total effect vs income-adjusted direct effect')
303 for f,l in [('vuln ~ tert','total (unadjusted)'),
304             ('vuln ~ tert + female_d + old + employed','+demographics, no income'),
305             ('vuln ~ tert + C(inc3) + female_d + old + employed','+income (direct effect)')]:
306     ml,_=wml(f,A); fw,_=firth_tab(f,A)
307     print(f"    {l:26} ML OR={ml.loc['tert','OR']:.3f} p={ml.loc['tert','p']:.4f}    Firth OR={fw.loc['tert',
    'OR']:.3f} [{fw.loc['tert','lo']:.3f},{fw.loc['tert','hi']:.3f}] p={fw.loc['tert','p']:.4f}")
308 mt,_=wml('vuln ~ tert + female_d + old + employed',A); md,_=wml('vuln ~ tert + C(inc3) + female_d + old +
    employed',A)
309 print(f"    income mediates {(1-md.loc['tert','coef']/mt.loc['tert','coef'])*100:.1f}% of education's
    coefficient")
310
311 # ----- exploratory
312 hdr('Exploratory: urban-rural')
313 t,n=wml('vuln ~ female_d*urban + C(inc3) + tert + employed + old',A)
314 r=t.loc['female_d:urban']; print(f"    Gender x Urban OR={r.OR:.3f} [{r.lo:.3f},{r.hi:.3f}] p={r.p:.4f}
    (n={n})")
315
316 # ----- TABLE 5 / R6
317 hdr('TABLE 5 Pre-specified tests: clustering sensitivity and Holm correction')
318 H=holm([P_H2,P_H3,P_H1])
319 rows5=[('H2 savings gap, pooled adjusted (penalized)',P_H2,H[0]),
320         ('H3 digital x age (weighted Firth, primary outcome)',P_H3,H[1]),
321         ('H3 digital x age (composite outcome)',P_H3C,None),
322         ('H1 gender x age interaction',P_H1,H[2]),
323         ('Vulnerability gap, pooled (secondary)',gp['p'],None)]
324 print(f"    {'test':52}{rho=0':>8}{'.03':>8}{'.06':>8}{Holm':>8}")
325 for l,pp,hh in rows5:
326     print(f"    {l:52}{pp:8.4f}{clust(pp,.03):8.4f}{clust(pp,.06):8.4f}{('%8.4f'%hh) if hh is not None else
    ' '}")
327
328 hdr('TABLE R6 Clustering sensitivity for all key tests')
329 rows6=[('Savings gap, pooled adjusted (Firth, reduced)',P_H2),
330         ('Savings gap, pooled unadjusted',gp2['p']),
331         ('Vulnerability gap, pooled',gp['p']),
332         ('H1 gender x age interaction',P_H1),
333         ('H3 digital x age (weighted Firth)',P_H3),
334         ('H3 digital x age (composite, weighted Firth)',P_H3C)]
335 print(f"    {'test':46}{rho=0':>8}{'.01':>8}{'.03':>8}{'.06':>8}")
336 for l,pp in rows6:
337     print(f"    {l:46}{pp:8.4f}{clust(pp,.01):8.4f}{clust(pp,.03):8.4f}{clust(pp,.06):8.4f}")
338
339 # ----- TABLE R8
340 hdr('TABLE R8 Weight diagnostics (vulnerability model sample)')
341 MS=A.dropna(subset=['tert']); w=MS.wgt
342 print(f"    n={len(MS)} min={w.min():.4f} max={w.max():.4f} mean={w.mean():.4f} CV={w.std()/w.mean():.4f}
    max/min={w.max()/w.min():.2f}")
343 print(f"    Kish design effect = {len(MS)*np.sum(w**2)/w.sum():.4f}")
344 lo_,hi_=np.percentile(A.wgt,[5,95]); T=A.copy(); T['w']=np.clip(T.wgt,lo_,hi_); T['w']=T.w/T.w.mean()
345 t,_=wml('vuln ~ female_d*old',T); print(f"    H1 interaction after 5/95 trimming: p={t.loc['female_d:old',
    'p']:.4f}")
346 tt,_=firth_tab('vuln ~ digital*old',T[T.female_d==1]); print(f"    H3 interaction (weighted Firth) after
    5/95 trimming: p={tt.loc['digital:old','p']:.4f}")
347
348 A[['wgt','w','female_d','age','ageg','old','vuln','fragile','sav','digital','tert','educ',
349     'inc_q','inc3','employed','urban']].to_csv('findex_v3_analytic_n%d.csv'%len(A),index=False)
350 print(f"\nAnalytic dataset written: findex_v3_analytic_n{len(A)}.csv")

```