The Role of Tacit Knowledge in Auditor Expertise and Human Capital Development

Jasmijn C. Bol, *Tulane University*
Cassandra Estep, *Emory University*
Frank Moers, *Maastricht University*
Mark E. Peecher, *University of Illinois at Urbana-Champaign*

Compliance with Data Policy for the Journal of Accounting Research:
Code Document

June 2018

Study 1

The code for Study 1 is made in SAS. Below is the code for the analyses related to Tables 4-7, as well as the robustness tests related to Study 1. Note that, with the exception of the robustness test discussed in Footnote 17, the same code as presented below for Tables 4-7 is used for the robustness tests with the difference being that independent variables are added or replaced, as discussed in detail in the paper. The code related to the test of Footnote 17 is described separately.

The variables names used in the code below link to the variable names in the paper as follows:

Variable name in paper	**Variable name in code**
TK Subordinate	`TK_Subordinate`
TK Supervisors	`TK_Supervisor`
Firm Tenure	`firm_tenure`
Job Tenure	`job_tenure`
Age	`age`
Job Level	`job_level`
Overall Evaluation	`Overall_Eval`
Technical Performance	`Technical_Perf`
Competencies	`Competencies`
Firm Commitment	`FirmCommitment`
Outside Opportunities	`find_job`
TeK Subordinate	`TeK_Subordinate`
TeK Supervisors	`TeK_Supervisor`
Aggregate	`Aggregate`
Potential Manager	`potential_manager`
Potential Partner	`potential_partner`

START OF CODE

```
/*label of dataset*/
%macro data;
      BEMP
%mend data;


/*creation of additional (interaction) variables*/
data %data; set %data;
TKSub_TKSup = TK_Subordinate*TK_Supervisor;
TechPerf_TKSup = Technical_Perf*TK_Supervisor;
Aggregate = Technical_Perf + Competencies;
Aggreg_TKSup = Aggregate*TK_Supervisor;
Compet_TKSup = TK_Supervisor*Competencies;
TeKSubSup = TeK_Subordinate*TeK_Supervisor;
run;


/*label of different sets of independent variables*/
%macro ivs1;
       TK_Subordinate job_level job_tenure age
%mend ivs1;
%macro ivs2;
       TK_Subordinate Technical_Perf job_level job_tenure age
%mend ivs2;
%macro ivs3;
       TK_Subordinate TK_Supervisor TKSub_TKSup Technical_Perf TechPerf_TKSup
job_level job_tenure age
%mend ivs3;
%macro ivs4;
      TK_Supervisor Aggregate Competencies  Aggreg_TKSup Compet_TKSup
%mend ivs4;
%macro ivs5;
       TK_Subordinate TK_Supervisor TKSub_TKSup job_level firm_tenure age
Overall_Eval find_job
%mend ivs5;
%macro ivs6;
       TeK_Subordinate TeK_Supervisor TeKSubSup job_level firm_tenure age
Overall_Eval find_job
%mend ivs6;


/*TABLE 4: STUDY 1 TESTS OF H1A & H1B -- PROMOTABILITY*/
proc logistic data=%data;
model potential_manager (event='1') = %ivs1; /* Column (1)*/
proc logistic data=%data;
model potential_manager (event='1') = %ivs2; /* Column (2)*/
proc logistic data=%data;
model potential_manager (event='1') = %ivs3; /* Column (3)*/
proc logistic data=%data;
model potential_partner (event='1') = %ivs1; /* Column (4)*/
proc logistic data=%data;
model potential_partner (event='1') = %ivs2; /* Column (5)*/
proc logistic data=%data;
model potential_partner (event='1') = %ivs3; run; /* Column (6)*/
```

```
/*TABLE 5: STUDY 1 TESTS OF H1A & H1B -- ANNUAL PERFORMANCE EVALUATIONS*/
proc reg data=%data;
model Overall_Eval = %ivs2; /* Column (1)*/
run; quit;
proc reg data=%data;
model Overall_Eval = %ivs3; /* Column (2)*/
model Overall_Eval = %ivs4; /* Column (3)*/
run; quit;


/*TABLE 6:STUDY 1 TESTS OF H2A -- SUBORDINATES' TACIT KNOWLEDGE ACQUISITION*/
data BEMP2; set %data;
if job_tenure<=12 then delete; run;
data BEMP2; set BEMP2;
if    TK_Subordinate=. then delete; run;
data BEMP2; set BEMP2;
if TK_Supervisor>=0.5 then psmTK_Supervisor=1; if TK_Supervisor<0.5 then
psmTK_Supervisor=0;
if TK_Supervisor=. then psmTK_Supervisor=.;
run;
      /*Panel A*/
proc logistic data=BEMP2 noprint;
model psmTK_Supervisor (event='1') =  job_level job_tenure / l=probit;
output out=alldata prob=prob; run;
%MACRO MATCH (Lib,Dataset,Matched,SCase,SControl,depend,digits);
data tcases (drop=cprob)
tctrl (drop=aprob) ;
set &LIB..&dataset. ;
if &depend. = 1 and prob ne .
then do;
cprob = round(prob,&digits.);
Cmatch = 0;
Length RandNum 8;
RandNum=ranuni(1234567);
Label RandNum=
'Uniform Randomization Score';
output tctrl;
end;
else if &depend. = 0 and prob ne .
then do;
aprob =round(prob,&digits.);
output tcases;
end;
run;
proc sort data=tcases
out=&LIB..&Scase.;
by prob;
run;
proc sort data=tctrl
out=&LIB..&Scontrol.;
by prob randnum;
run;
data &lib..&matched.
(drop=Cmatch randnum aprob cprob start
oldi curctrl matched);
```

```
set &lib..&SCase. ;
curob + 1;
matchto = curob;
if curob = 1 then do;
start = 1;
oldi = 1;
end;
DO i = start to n;
set &lib..&Scontrol. point = i nobs = n;
if i gt n then goto startovr;
if _Error_ = 1 then abort;
curctrl = i;
if aprob = cprob then
do;
Cmatch = 1;
output &lib..&matched.;
matched = curctrl;
goto found;
end;
else if cprob gt aprob then
goto nextcase;
startovr: if i gt n then
goto nextcase;
END;
nextcase:
if Cmatch=0 then start = oldi;
found:
if Cmatch = 1 then do;
oldi = matched + 1;
start = matched + 1;
set &lib..&SCase. point = curob;
output &lib..&matched.;
end;
retain oldi start;
if _Error_=1 then _Error_=0;
run;
%MEND MATCH;
%MATCH(work,ALLDATA,Matchdata,Casedata,Controldata,psmTK_Supervisor,.01);
proc npar1way data = Matchdata; /*comparison based on n=28*/
class psmTK_Supervisor;
var TK_Subordinate;
run;
proc logistic data=BEMP2 noprint;
model psmTK_Supervisor (event='1') =  job_level / l=probit;
output out=alldata prob=prob; run;
%MATCH(work,ALLDATA,Matchdata,Casedata,Controldata,psmTK_Supervisor,.01);
proc npar1way data = Matchdata; /*comparison based on n=40*/
class psmTK_Supervisor;
var TK_Subordinate;
run;
      /*Panel B*/
data BEMP2; set BEMP2;
if job_level=5 then delete; run; /*creating common support for job_level*/
data BEMP2; set BEMP2;
if job_tenure<23 or job_tenure>131 then delete; /*creating common support for
job_tenure*/
run;
```

```
proc reg data=BEMP2;
model TK_Subordinate = psmTK_Supervisor job_level job_tenure; run; quit;


/*TABLE 7:STUDY 1 TESTS OF H2B -- SUBORDINATES' FIRM COMMITMENT*/
proc reg data=%data;
model  FirmCommitment =  %ivs5 ; /* Column (1)*/
run; quit;
proc reg data=%data;
model  FirmCommitment =  %ivs5 ; /* Column (2)*/
model  FirmCommitment =  %ivs6 ; /* Column (3)*/
run; quit;


/* Code for the jackknife estimator and clustered standard errors adjusted
for small sample bias: Footnote 17 */
      /*logit regressions*/
%macro jkalogit;
proc freq data=%data noprint; table manager*idmgr / list out=t1;
proc rank data=t1 out=t1; var idmgr; ranks nrman;
proc univariate data=t1 noprint; var nrman;
output out=summan max=nrsup;
Proc surveyselect data=summan out=summan2 seed=1346 method=urs sampsize=1
Rep=# noprint; /* Note that "#" is replaced by the actual number of managers
in the sample, which differs per regression */
data t1; merge t1 summan2;
data t1; set t1;
if manager="" then delete;
keep manager nrman nrsup;
data BEMP2; merge %data t1; by manager;
%do i=1 %to #; /* Note that "#" is replaced by the actual number of managers
in the sample, which differs per regression */
data jk&i; set BEMP2;
if nrman=&i then delete;
proc surveylogistic data=jk&i;
cluster manager;
model %dv (event='1') = %ivs;
ods output parameterestimates=par&i nobs=nobs&i; run;
data nobs&i; set nobs&i;
if label="Number of Observations Used" then output;
data par&i; set par&i;
replicate=&i;
data nobs&i; set nobs&i;
replicate=&i;
keep replicate n;
%end;
data par; set par1-par#; /* Note that "#" is replaced by the actual number of
managers in the sample, which differs per regression */
data nobs; set nobs1-nobs#; /* Note that "#" is replaced by the actual number
of managers in the sample, which differs per regression */
data par; merge par nobs; by replicate;
data par; set par;
BLRse=stderr*(((#-1)/(#-2))**(0.5)); /* Note that "#" is replaced by the
actual number of managers in the sample, which differs per regression */
BLRt=abs(estimate/blrse);
proc sort data=par; by variable;
proc univariate data=par noprint; by variable; var n;
```

```
output out=ss sum=totn;
data par; merge par ss; by variable;
data par; set par;
proc univariate data=par noprint; by variable; var estimate BLRt;
output out=result mean=estimate BLRt;
data result; set result;
drop x;
RUN; quit;
%mend jkalogit;
%macro dv;
      /* Here either "potential_manager" or "potential_partner" is used */
%mend dv;
%macro ivs;
    /* Here the relevant independent variables are used, related to
regressions of Table 4 */
%mend ivs;
%jkalogit;

      /*OLS regressions*/
%macro jkaols;
proc freq data=%data noprint; table manager*idmgr / list out=t1;
proc rank data=t1 out=t1; var idmgr; ranks nrman;
proc univariate data=t1 noprint; var nrman;
output out=summan max=nrsup;
Proc surveyselect data=summan out=summan2 seed=1346 method=urs sampsize=1
Rep=# noprint; /* Note that "#" is replaced by the actual number of managers
in the sample, which differs per regression */
data t1; merge t1 summan2;
data t1; set t1;
if manager="" then delete;
keep manager nrman nrsup;
data BEMP2; merge %data t1; by manager;
%do i=1 %to #; /* Note that "#" is replaced by the actual number of managers
in the sample, which differs per regression */
data jk&i; set BEMP2;
if nrman=&i then delete;
proc surveyreg data=jk&i;
cluster manager;
model %dv  = %ivs ;
ods output parameterestimates=par&i; run;
data par&i; set par&i;
replicate=&i;
%end;
data par; set par1-par#; /* Note that "#" is replaced by the actual number of
managers in the sample, which differs per regression */
data par; merge par; by replicate;
data par; set par;
BLRse=stderr*(((#-1)/(#-2))**(0.5)); /* Note that "#" is replaced by the
actual number of managers in the sample, which differs per regression */
BLRt=abs(estimate/blrse);
proc sort data=par; by parameter;
proc univariate data=par noprint; by parameter; var estimate BLRt;
output out=result mean=estimate BLRt;
RUN; quit;
%mend jkaols;
%macro dv;
      /* Here either "Overall_Eval" or "FirmCommitment" is used */
```

```
%mend dv;
%macro ivs;
    /* Here the relevant independent variables are used, related to the
regressions of Tables 5 or 7 */
%mend ivs;
%jkaols;
```

END OF CODE

Study 2

The code for Study 2 is made in Stata. Below is the code for the analyses related to Tables 10-11, as well as the robustness tests related to Study 2. Note that the same code as presented below for Tables 10-11 is used for the robustness tests with the difference being that are added or replaced, as discussed in detail in the paper.

The variables names used in the code below link to the variable names in the paper as follows:

Variable name in paper	**Variable name in code**
TK-All	tk_all
TK-BEP	tk_bep
TK-TL	tk_tl
TeK	techknow
Changes in TK-All	change_tkall
Changes in TK-BEP	chage_tkbep
Changes in TK-TL	change_tktl
Big 4	big4
Performance Evaluation	perfeval_overall
Bonus	bonus
Initial Salary	salary_init

START OF CODE

```
***Creation of additional variables
*Big 4 variable – the Firm variable is based on exported response codes from Qualtrics
gen big4=0
replace big4=1 if Firm == 3
replace big4=1 if Firm == 5
replace big4=1 if Firm == 7
replace big4=1 if Firm == 9
*Technical knowledge score – based on CPA exam scores (per Footnote 27 of the paper some
participants had not finished all four sections)
gen cpacount = 0
replace cpacount = cpacount + 1 if CPA_BEC < 100 & CPA_BEC > 0
replace cpacount = cpacount + 1 if CPA_AUD < 100 & CPA_AUD > 0
replace cpacount = cpacount + 1 if CPA_REG < 100 & CPA_REG > 0
replace cpacount = cpacount + 1 if CPA_FAR < 100 & CPA_FAR > 0
gen techknow = (CPA_AUD + CPA_REG + CPA_FAR + CPA_BEC)/4
replace techknow = (CPA_AUD + CPA_REG + CPA_FAR)/3 if CPA_BEC > 100 | CPA_BEC == 0
replace techknow =(CPA_AUD + CPA_FAR + CPA_BEC)/3 if CPA_REG > 100 | CPA_REG == 0
replace techknow = . if cpacount == 0
replace techknow = CPA_AUD if cpacount == 1
*Split Top and Bottom auditors based on mean of performance evauation
sum perfeval_overall, d
gen perfevalmean = r(mean)
gen topaud = 0
```

```
replace topaud = 1 if perfeval_overall > perfevalmean
```
***Split Top and Bottom auditors based on median of performance evaluation (median score = 75)**
```
gen topaudmed = 0
replace topaudmed = 1 if perfeval_overall > 75
```

*****Tests for Table 10: TABLE 10: STUDY 2 TEST OF H1A–SUBORDINATES' ANNUAL PERFORMANCE EVALUATIONS, PANEL A**

***Main test in table**
```
ttest tk_all, by (topaud)
ttest tk_bep, by (topaud)
ttest tk_tl, by (topaud)
```
***Tests referenced in Footnote 34**
```
ttest tk_all, by (topaudmed)
ttest tk_bep, by (topaudmed)
ttest tk_tl, by (topaudmed)
```

*****Tests for TABLE 10: STUDY 2 TEST OF H1A–SUBORDINATES' ANNUAL PERFORMANCE EVALUATIONS, PANEL B**

***Base model**
```
regress perfeval_overall techknow big4, robust
```
***Model (1)**
```
regress perfeval_overall tk_all change_tkall techknow big4, robust
```
***Model (2)**
```
regress perfeval_overall tk_bep change_tkbep techknow big4, robust
```
***Model (3)**
```
regress perfeval_overall tk_tl change_tktl techknow big4, robust
```
***Tests referenced in Footnote 35**
```
regress perfeval_overall tk_all techknow big4, robust
regress perfeval_overall tk_bep techknow big4, robust
regress perfeval_overall tk_tl techknow big4, robust
regress perfeval_overall tk_all change_tkall big4, robust
regress perfeval_overall tk_bep change_tkbep big4, robust
regress perfeval_overall tk_tl change_tktl big4, robust
regress perfeval_overall tk_all change_tkall techknow, robust
regress perfeval_overall tk_bep change_tkbep techknow, robust
regress perfeval_overall tk_tl change_tktl techknow, robust
regress perfeval_overall tk_all change_tkall, robust
regress perfeval_overall tk_bep change_tkbep, robust
regress perfeval_overall tk_tl change_tktl, robust
regress perfeval_overall tk_all techknow, robust
regress perfeval_overall tk_bep techknow, robust
regress perfeval_overall tk_tl techknow, robust
regress perfeval_overall tk_all big4, robust
regress perfeval_overall tk_bep big4, robust
regress perfeval_overall tk_tl big4, robust
```
***Results are similar with or without using the robust command**

*****Tests for TABLE 11: STUDY 2 TEST OF H1A–SUBORDINATES' BONUS COMPENSATION**

```
*Base model
regress bonus techknow perfeval_overall big4 salary_init, robust
*Model (1)
regress bonus change_tkall tk_all techknow perfeval_overall big4 salary_init, robust
*Model (2)
regress bonus change_tkbep tk_bep techknow perfeval_overall big4 salary_init, robust
*Model (3)
regress bonus change_tktl tk_tl techknow perfeval_overall big4 salary_init, robust
*Tests referenced in Footnote 37 - excluding TK level variable
regress bonus change_tkall techknow perfeval_overall big4 salary_init, robust
regress bonus change_tkbep techknow perfeval_overall big4 salary_init, robust
regress bonus change_tktl techknow perfeval_overall big4 salary_init, robust
*Tests referenced in Footnote 37 - using total compensation as DV
gen totalcomp = bonus + salary_current
regress totalcomp change_tkall techknow perfeval_overall big4 salary_init, robust
regress totalcomp change_tkbep techknow perfeval_overall big4 salary_init, robust
regress totalcomp change_tktl techknow perfeval_overall big4 salary_init, robust
*Tests referenced in Footnote 37 - using change in total compensation as DV
gen compchange = [(bonus + salary_current) - salary_init]/salary_init
regress compchange change_tkall techknow perfeval_overall big4 salary_init, robust
regress compchange change_tkbep techknow perfeval_overall big4 salary_init, robust
regress compchange change_tktl techknow perfeval_overall big4 salary_init, robust
*Tests referenced in Footnote 39 - using only Big 4 auditors
regress bonus change_tkall tk_all techknow perfeval_overall salary_init if big4 == 1, robust
regress bonus change_tkbep tk_bep techknow perfeval_overall big4 salary_init if big4 == 1, robust
regress bonus change_tktl tk_tl techknow perfeval_overall big4 salary_init if big4 == 1, robust
*Results are similar with or without using the robust command
```

END OF CODE