```
LIBNAME c '_path_';

*Income shifting macro;
%macro avgftr(n=, dsetout=);

RSUBMIT;
data tax;
set comp.funda;
where indfmt = 'INDL' and datafmt = 'STD' and consol = 'C' and popsrc = 'D';
if fyear<1987 then STR=0.46;
if fyear=1987 then STR=0.4;
if 1988<=fyear<=1992 then STR=0.34;
if 1992<fyear<=2017 then STR=0.35;
if fyear>2017 then STR=0.21;
if txdfo=. then txdfo=txdi-txdfed-txds;
if txfo=. then txfo=txc-txfed-txs;
if pifo=. then pifo=pi-pidom;
if pidom=. then pidom=pi-pifo;
if xrd=. then xrd=0;
if xad=. then xad=0;
if intan=. then intan=0;
if ch=. then ch=0;
if dt=. then dt=0;
keep gvkey cik datadate fyear at pi pifo pidom sale txfed txfo txdfed txdfo
STR
      xrd xad intan ch dt;
run;

PROC SQL;
CREATE TABLE segment
AS SELECT   gvkey, datadate, geotp, sales, ias
FROM        compsegd.WRDS_SEGMERGED
WHERE       stype='GEOSEG'
Group BY    gvkey, datadate having srcdate = min(srcdate)
order by    gvkey, datadate;
QUIT;
proc sql;
create table segment1
as select distinct gvkey, datadate, geotp, sum(sales) as total_sale
from segment
group by gvkey, datadate, geotp;
quit;
proc transpose data=segment1 out=segment2
(rename=(_1=obsolete _2=dsale _3=fsale));
   by gvkey datadate;
   id geotp;
   var total_sale;
run;
proc sql;
create table segment3
as select distinct gvkey, datadate, geotp, sum(ias) as total_at
from segment
group by gvkey, datadate, geotp;
quit;
proc transpose data=segment3 out=segment4
(rename=(_1=obsolete _2=d_at _3=f_at));
   by gvkey datadate;
```

```
   id geotp;
   var total_at;
run;
proc sql;
create table tax2
as select a.*, b.dsale, b.fsale, c.d_at, c.f_at
from tax as a left join segment2 as b
on a.gvkey = b.gvkey and a.datadate=b.datadate
left join segment4 as c
on a.gvkey = c.gvkey and a.datadate=c.datadate;
quit;
data tax2;
set tax2;
if dsale=. then dsale=sale-fsale;
if fsale=. then fsale=sale-dsale;
if d_at=. then d_at=at-f_at;
if f_at=. then f_at=at-d_at;
run;

data company;
set comp.company;
keep gvkey fic sic;
run;
proc sql;
create table tax3 as
select a.*, b.fic, b.sic
from tax2 a left join Company b
on a.gvkey = b.gvkey
order by a.gvkey, a.fyear;
quit;
PROC download data=tax3 out=x1;
run;
endrsubmit;

proc sql;
create table x2
as select distinct a.*, sum(b.pifo) as cum_pifo, sum(b.pidom) as cum_pidom,
sum(b.txfo) as cum_txfo,
                                    sum(b.txdfo) as cum_txdfo, sum(b.str) as
cum_STR, sum(b.pi) as cum_pi, sum(b.sale) as cum_sale,
                                    sum(b.dsale) as cum_dsale,sum(b.fsale) as
cum_fsale, max(b.fyear) as t,
                                    sum(b.f_at) as cum_f_at, sum(b.d_at) as
cum_d_at, sum(b.at) as cum_at, count(b.f_at) as count_f_at,
                                    count(b.d_at) as count_d_at,
                                    sum(b.xrd) as cum_xrd, sum(b.xad) as
cum_xad, sum(b.intan) as cum_intan,
                                    sum(b.ch) as cum_ch, sum(b.dt) as cum_dt,
avg(b.at) as avg_at
from x1 as a, x1 as b
where a.gvkey=b.gvkey and 0<=intck('year', b.datadate, a.datadate)<&n
group by a.gvkey, a.datadate having count(b.pifo)=&n and count(b.txfo)=&n and
count(b.txdfo)=&n and count(b.fsale)=&n;
quit;

data &dsetout (keep=gvkey cik ciknumber fic sic sic2 datadate fyear t AvgFTR
AvgFETR Lowavgftr Highavgftr
```

```
                        AvgROS AvgFROS AvgDROS AvgFROA AvgDROA AvgROA
fsale_sale AvgPIFO_PI AvgFsale AvgDsale
                        avgrd_at avgrd_sale avgads_at avgads_sale avgintan_at
avgintan_sale avgcash_at avgdebt_at avgsize);
set x2;
ciknumber=cik+1-1;
format ciknumber BEST12.;
informat ciknumber BEST32.;
sic2=input(substr(sic,1,2),12.);
if cum_pifo>0;
if cum_pidom>0;
AvgFETR=(cum_txfo+cum_txdfo)/cum_pifo;
AvgFTR=(cum_txfo+cum_txdfo)/cum_pifo-(cum_STR)/&n;
if AvgFTR<=0 then Lowavgftr=1; else Lowavgftr=0;
if AvgFTR>0 then Highavgftr=1; else Highavgftr=0;
AvgROS=cum_pi/cum_sale;
AvgFROS=cum_pifo/cum_fsale;
AvgDROS=cum_pidom/cum_dsale;
AvgPIFO_PI=cum_pifo/cum_pi;
AvgFROA=cum_pifo/cum_f_at;
      if count_f_at ne &n then AvgFROA=.;
AvgDROA=cum_pidom/cum_d_at;
      if count_d_at ne &n then AvgDROA=.;
AvgROA=cum_pi/cum_at;
if -1<=AvgFTR<=1;
fsale_sale=cum_fsale/(cum_dsale+cum_fsale);
AvgFsale=cum_fsale/&n;
AvgDsale=cum_dsale/&n;
avgrd_at=cum_xrd/cum_at;
avgrd_sale=cum_xrd/cum_sale;
avgads_at=cum_xad/cum_at;
avgads_sale=cum_xad/cum_sale;
avgintan_at=cum_intan/cum_at;
avgintan_sale=cum_intan/cum_sale;
avgcash_at=cum_ch/cum_at;
avgdebt_at=cum_dt/cum_at;
avgsize=log(1+avg_at);
run;

proc delete data=work.x1 work.x2 ; run;

%mend avgftr;

*Winsorization macro;
%macro winsor(dsetin=, dsetout=, byvar=none, vars=, type=winsor, pctl=1 99);

%if &dsetout = %then %let dsetout = &dsetin;

%let varL=;
%let varH=;
%let xn=1;

%do %until ( %scan(&vars,&xn)= );
    %let token = %scan(&vars,&xn);
    %let varL = &varL &token.L;
    %let varH = &varH &token.H;
    %let xn=%EVAL(&xn + 1);
```

```
%end;

%let xn=%eval(&xn-1);

data xtemp;
    set &dsetin;
    run;

%if &byvar = none %then %do;

    data xtemp;
        set xtemp;
        xbyvar = 1;
        run;

    %let byvar = xbyvar;

%end;

proc sort data = xtemp;
    by &byvar;
    run;

proc univariate data = xtemp noprint;
    by &byvar;
    var &vars;
    output out = xtemp_pctl PCTLPTS = &pctl PCTLPRE = &vars PCTLNAME = L H;
    run;

data &dsetout;
    merge xtemp xtemp_pctl;
    by &byvar;
    array trimvars{&xn} &vars;
    array trimvarl{&xn} &varL;
    array trimvarh{&xn} &varH;

    do xi = 1 to dim(trimvars);

        %if &type = winsor %then %do;
            if not missing(trimvars{xi}) then do;
              if (trimvars{xi} < trimvarl{xi}) then trimvars{xi} =
trimvarl{xi};
              if (trimvars{xi} > trimvarh{xi}) then trimvars{xi} =
trimvarh{xi};
            end;
        %end;

        %else %do;
            if not missing(trimvars{xi}) then do;
              if (trimvars{xi} < trimvarl{xi}) then delete;
              if (trimvars{xi} > trimvarh{xi}) then delete;
            end;
        %end;

    end;
    drop &varL &varH xbyvar xi;
    run;
```

```
%mend winsor;

*Obtain multi-period income shifting incentives variables [Klassen & Laplante
2012];
%avgftr(n=5, dsetout=c.avgftr);
proc sort data=c.avgftr nodupkey; by gvkey fyear; run;

%avgftr(n=3, dsetout=c.avgftr_3);
proc sort data=c.avgftr_3 nodupkey; by gvkey fyear; run;

*Code to generate HatFTR;
RSUBMIT;
data tax;
set comp.funda;
where indfmt = 'INDL' and datafmt = 'STD' and consol = 'C' and popsrc = 'D';
if fyear<1987 then STR=0.46;
if fyear=1987 then STR=0.4;
if 1988<=fyear<=1992 then STR=0.34;
if 1992<fyear<=2017 then STR=0.35;
if fyear>2017 then STR=0.21;
if txdfo=. then txdfo=txdi-txdfed-txds;
if txfo=. then txfo=txc-txfed-txs;
if pifo=. then pifo=pi-pidom;
if pidom=. then pidom=pi-pifo;
FTR=(txfo+txdfo)/pifo-STR;
ROS=pi/sale;
keep gvkey cik datadate fyear at pi pifo pidom sale ftr ros;
run;

PROC SQL;
CREATE TABLE segment
AS SELECT   gvkey, datadate, geotp, sales
FROM        compsegd.WRDS_SEGMERGED
WHERE       stype='GEOSEG'
Group BY    gvkey, datadate having srcdate = min(srcdate)
order by    gvkey, datadate;
QUIT;
proc sql;
create table segment1
as select distinct gvkey, datadate, geotp, sum(sales) as total_sale
from segment
group by gvkey, datadate, geotp;
quit;
proc transpose data=segment1 out=segment2
(rename=(_1=obsolete _2=dsale _3=fsale));
   by gvkey datadate;
   id geotp;
   var total_sale;
run;
proc sql;
create table tax2
as select a.*, b.dsale, b.fsale
from tax as a left join segment2 as b
on a.gvkey = b.gvkey and a.datadate=b.datadate;
quit;
data tax2;
```

```
set tax2;
if dsale=. then dsale=sale-fsale;
if fsale=. then fsale=sale-dsale;
run;
data company;
set comp.company;
keep gvkey fic sic;
run;
proc sql;
create table tax3 as
select a.*, b.fic, b.sic
from tax2 a left join Company b
on a.gvkey = b.gvkey
order by a.gvkey, a.fyear;
quit;
PROC download data=tax3 out=x1;
run;
endrsubmit;

proc sort data=x1 nodupkey; by gvkey fyear; run;

proc sql;
create table x2 as
select a.*, b.ftr as ftr_L1 'ftr_L1', c.ftr as ftr_L2 'ftr_L2', d.ftr as
ftr_L3 'ftr_L3'
from x1 a left join x1 b
on a.gvkey = b.gvkey and a.fyear=b.fyear+1
left join x1 c
on a.gvkey = c.gvkey and a.fyear=c.fyear+2
left join x1 d
on a.gvkey = d.gvkey and a.fyear=d.fyear+3
order by a.gvkey, a.fyear;
quit;

data x2;
set x2;
sic2=substr(left(put(sic,10.0)),1,2);
if 1997<=fyear<=2017;
if 6000<=sic<=6999 then delete; *Financial;
if 4900<=sic<=4999 then delete; *Utilities;
if fic='USA';
if -1<=ftr<=1;
if -1<=ftr_L1<=1;
if -1<=ftr_L2<=1;
if not missing(ftr+ftr_L1+ftr_L2+ROS);
if ROS>0;
run;

%winsor(dsetin=x2, dsetout=x3, byvar=none,
vars= ros, type=winsor, pctl=1 99);

ODS HTML CLOSE; ODS HTML;
proc means data=x3 n nmiss mean min p1 p5 p25 p50 p75 p95 p99 max std;
var ftr ftr_L1 ftr_L2 ROS;
quit;

ODS HTML CLOSE; ODS HTML;
```

```
proc glm data=x3;
class sic2 fyear;*fixed effects;
model ftr = ftr_L1 ftr_L2 ROS sic2 fyear /solution;
output out=x4 p=pred;
run;
quit;

data c.Hatftr;
set x4;
Hatftr=pred;
label Hatftr="HatFTR [t-2, t]";
keep gvkey fyear hatftr;
run;

/*************************Sample Selection
Process*****************************/
*Clean income shifting dataset;
data c.avgftr2;
set c.avgftr;
if not missing(AvgFROS+avgDROS+AvgROS+avgftr);
gvkey2=gvkey+1-1;
format gvkey2 BEST12.;
informat gvkey2 12.;
run;
proc sort data=c.avgftr2 out=temp nodupkey; by gvkey; run;*Count # unique
firms;
data c.avgftr2;*define sample period;
set c.avgftr2;
if 1997<=fyear<=2017;
run;
proc sort data=c.avgftr2 out=temp nodupkey; by gvkey; run;*Count # unique
firms;
data c.avgftr2;*remove financial institutions and utilities;
set c.avgftr2;
if 6000<=sic<=6999 then delete; *Financial;
if 4900<=sic<=4999 then delete; *Utilities;
run;
proc sort data=c.avgftr2 out=temp nodupkey; by gvkey; run;*Count # unique
firms;
data c.avgftr2;*US firms only;
set c.avgftr2;
if fic='USA';
run;
proc sort data=c.avgftr2 out=temp nodupkey; by gvkey; run;*Count # unique
firms;
data c.avgftr2;*Positive geographic sales and non-extreme profitability
measures;
set c.avgftr2;
if AvgFsale>0;
if AvgDsale>0;
if avgDROS<1;
if avgFROS<1;
if 0<avgROS<1;
run;
proc sort data=c.avgftr2 out=temp nodupkey; by gvkey; run;*Count # unique
firms;
```

```
/*************************Firm fundamentals*****************************/
*Industry Codes;
rsubmit;
data company;
set comp.company;
keep gvkey cik conm ggroup gind gsector gsubind naics sic fic;
run;
PROC download data=company out=c.company;
run;
endrsubmit;

*Controls;
rsubmit;
%let identifier = gvkey cik datadate fyear sich;
%let financial = at xrd xad xint xrent intan ch dt sale pi pifo pidom invfg
invt
                        emp cogs xlr xsga ni prcc_f csho capx ppent stkco;
data control;
set comp.funda;
where indfmt = 'INDL' and datafmt = 'STD' and consol = 'C' and popsrc = 'D';
keep &identifier &financial;
if xrd=. then xrd=0;
if xad=. then xad=0;
if xsga=. then xsga=0;
if capx=. then capx=0;
if xint=. then xint=0;
if xrent=. then xrent=0;
if intan=. then intan=0;
if ch=. then ch=0;
if dt=. then dt=0;
if invfg=. then invfg=0;
if invt=. then invt=0;
if STKCO=. then STKCO=0;
if fyear>=1985;
run;
PROC SQL;
CREATE TABLE geo
AS SELECT   gvkey, datadate, geotp, sales
FROM        compsegd.WRDS_SEGMERGED
WHERE       stype='GEOSEG'
Group BY    gvkey, datadate having srcdate = min(srcdate)
order by    gvkey, datadate;
QUIT;
proc sql;
create table geo1
as select distinct gvkey, datadate, geotp, sum(sales) as total_sale
from geo
group by gvkey, datadate, geotp;
quit;
proc transpose data=geo1 out=geo2
(rename=(_1=obsolete _2=dsale _3=fsale));
   by gvkey datadate;
   id geotp;
   var total_sale;
run;
proc sql;
create table control2 as
```

```
select a.*, b.fic, b.sic, b.addzip, c.dsale, c.fsale
from control a left join comp.company b
on a.gvkey = b.gvkey
left join geo2 as c
on a.gvkey = c.gvkey and a.datadate=c.datadate
order by a.gvkey, a.fyear;
quit;
data control2;
set control2;
if dsale=. then dsale=sale-fsale;
if fsale=. then fsale=sale-dsale;
run;
PROC download data=control2 out=c.control;
run;
endrsubmit;
signoff;

*Construct variables in five-year averages;
proc sql;
create table temp
as select distinct a.gvkey, a.fyear, a.datadate, a.sic, a.sich, a.addzip,
a.at, a.sale, a.fsale, sum(b.sale) as cum_sale, sum(b.at) as cum_at,
                                sum(b.invfg) as cum_invfg, sum(b.invt) as
cum_invt, avg(b.emp) as avg_emp, std(b.emp) as std_emp,
                                sum(b.cogs) as cum_cogs, avg(b.xlr) as avg_xlr,
avg(b.stkco) as avg_stkco, sum(b.emp) as cum_emp, sum(b.ni) as cum_ni
from c.control as a, c.control as b
where a.gvkey=b.gvkey and 0<=intck('year', b.datadate, a.datadate)<5
group by a.gvkey, a.datadate having count(b.at)=5;

create table temp1
as select distinct a.*, avg(b.at) as avg_at 'Average Assets [t-4,t]'
from temp as a, temp as b
where a.gvkey=b.gvkey and 0<=intck('year', b.datadate, a.datadate)<5
group by a.gvkey, a.datadate;

create table c.control2
as select distinct a.*, b.fsale as l4fsale
from temp1 as a left join temp1 as b
on a.gvkey=b.gvkey and a.fyear=b.fyear+4;
quit;

*Construct additional variables;
data c.control3;
set c.control2;
sic3=input(substr(sic,1,3),12.);
if sic3=283 | sic3=357 | 360<=sic3<=368 | sic3=481 | sic3=737 | sic3=873 then
high_tech=1;
      else high_tech=0;
Fgrowth=(fsale/l4fsale)**(1/5)-1;
invfg_at=cum_invfg/cum_at;
invt_at=cum_invt/cum_at;
cogs_sale=cum_cogs/cum_sale;
cogs_at=cum_cogs/cum_at;
cogs_exp=cum_cogs/(cum_sale-cum_ni);
pay=(avg_xlr)/avg_emp*1000;
label pay="Average Employee pay t-4 - t";
```

```
emp_vol=std_emp/avg_emp;
drop sic3;
run;

*Hoberg and Phillips data (available online);
proc sql;
create table c.HobergPhillips2
as select distinct a.*, avg(b.tnic3tsimm) as avg_similarity
from c.HobergPhillips as a left join c.HobergPhillips as b
on a.gvkey=b.gvkey and 0<=(a.year-b.year)<=4
group by a.gvkey, a.year;
quit;

/*************************Merging all datasets*************************/
*patent_wrds dataset is pre-downloaded from WRDS;
proc sql;
create table c.avgftr3 as
select a.*, b.sich, b.addzip, b.high_tech, b.Fgrowth 'average FGrowth (5th
root)', b.avg_at,
            b.invfg_at 'average invfg_at', b.invt_at 'average invt_at',
b.avg_emp 'average avg_emp', b.emp_vol,
            b.cogs_sale 'average cogs_sale', b.cogs_at 'average cogs_at',
b.cogs_exp 'average cogs_exp',
            b.pay 'average pay per employee',
            c.conm, c.ggroup, c.gind, c.gsubind, c.naics,
            d.sale as sale_annual '$sale_annual', d.fsale as fsale_annual
'$fsale_annual', d.fsale/d.sale as fsale_sale_annual 'fsale_sale_annual',
            e.Hatftr,
            f.avg_similarity,
            h.grant_count_per_year 'Annual patent grant count',
            h.avg_grant_count_per_year 'Average grant count from [2011,
2017], where patent data begin in 2011'
from c.avgftr2 as a left join c.control3 as b
on a.gvkey=b.gvkey and a.fyear=b.fyear
left join c.company as c
on a.gvkey=c.gvkey
left join c.control as d
on a.gvkey=d.gvkey and a.fyear=d.fyear
left join c.hatftr as e
on a.gvkey=e.gvkey and a.fyear=e.fyear
left join c.HobergPhillips2 as f
on a.gvkey2=f.gvkey and a.fyear=f.year
left join c.patent_wrds as h
on a.gvkey=h.gvkey and a.fyear=h.fyear
order by a.gvkey, a.datadate;
quit;

*Backfill patent data;
proc sql;
create table temp as
select distinct a.*, b.grant_count_per_year as grant_count2011 'Grant Count
in 2011'
from c.avgftr3 as a left join c.avgftr3 as b
on a.gvkey=b.gvkey and b.fyear=2011
order by a.gvkey, a.fyear;
quit;
data temp;
```

```
set temp;
if fyear<2011 & grant_count_per_year=. then
grant_count_per_year_bf=grant_count2011;
if fyear>=2011 then grant_count_per_year_bf=grant_count_per_year;
label grant_count_per_year_bf='Annual patent grant count (backfilled)';
drop grant_count2011;
run;
proc sql;
create table c.avgftr4 as
select distinct a.*, avg(b.grant_count_per_year) as avg_grant_count_per_year5
'Average grant count [t-4,t]',
                              avg(b.grant_count_per_year_bf) as
avg_grant_count_per_year_bf5 'Average grant count (backfilled) [t-4,t]'
from temp as a left join temp as b
on a.gvkey=b.gvkey and 0<=intck('year', b.datadate, a.datadate)<5
group by a.gvkey, a.datadate
order by a.gvkey, a.fyear;
quit;

*Merge with EX 21 tax haven data (pre-downloaded from WRDS) and foreign
employment data;
proc sql;
create table c.avgftr5 as
select a.*, d.avg_Haven_intensity as avg_Haven_intensity_wrds,
d.avg_Big7_intensity as avg_Big7_intensity_wrds,
                  d.avg_Big4_intensity as avg_Big4_intensity_wrds,
d.avg_Big4plus_intensity as avg_Big4plus_intensity_wrds,
                  d.avg_Holding_intensity as avg_Holding_intensity_wrds,
d.avg_Sink_intensity as avg_Sink_intensity_wrds,
                  d.avg_Conduit_intensity as avg_Conduit_intensity_wrds,
d.avg_OSG_intensity as avg_OSG_intensity_wrds,
                  d.avg_OSG5_intensity as avg_OSG5_intensity_wrds,
d.avg_OSG10_intensity as avg_OSG10_intensity_wrds,
                  d.avg_Desai12_intensity as avg_Desai12_intensity_wrds,

                  d.Big4_intensity as Big4_intensity_wrds,
d.Big4plus_intensity as Big4plus_intensity_wrds,
                  d.OSG_intensity as OSG_intensity_wrds,

                  e.F_emp_pct,
                  f.resid
from c.avgftr4 as a left join c.sub_wrds as d
on a.gvkey=d.gvkey and a.fyear=d.fyear
left join c.F_emp as e
on a.gvkey2=e.gvkey and a.fyear=e.fyear
left join c.F_empresid as f
on a.gvkey=f.gvkey and a.fyear=f.fyear
order by a.gvkey, a.fyear;
quit;
proc sort data=c.avgftr5 nodupkey; by gvkey fyear; run;

*Clean controls;
data c.avgftr5;
set c.avgftr5;
if not missing (avgsize);
Avg_Dot_intensity_WRDS=Avg_Haven_intensity_WRDS-Avg_Big7_intensity_WRDS;
label Avg_Dot_intensity_wrds="Average Dot Haven Intensity [t-4,t]";
```

```
Avg_nonBig4_intensity_WRDS=Avg_Haven_intensity_WRDS-Avg_Big4_intensity_WRDS;
label Avg_nonBig4_intensity_wrds="Average Non Big 4 Haven Intensity [t-4,t]";
Avg_Dotcount_WRDS=Avg_Havencount_WRDS-Avg_Big7count_WRDS;
label Avg_Dotcount_WRDS="Average Dot Haven Count [t-4,t]";
Avg_NonBig4count_WRDS=Avg_Havencount_WRDS-Avg_Big4count_WRDS;
label Avg_NonBig4count_WRDS="Average Non Big 4 Haven Count [t-4,t]";
*Focus on outbound shifting only;
if highavgftr=1 then do;
avg_haven_intensity_wrds=.;
avg_dot_intensity_wrds=.;
avg_nonBig4_intensity_wrds=.;
avg_big7_intensity_wrds=.;
avg_big4_intensity_wrds=.;
avg_big4plus_intensity_wrds=.;
avg_holding_intensity_wrds=.;
avg_sink_intensity_wrds=.;
avg_conduit_intensity_wrds=.;
avg_OSG_intensity_wrds=.;
avg_OSG5_intensity_wrds=.;
avg_OSG10_intensity_wrds=.;
avg_Desai12_intensity_wrds=.;
Big4_intensity_wrds=.;
Big4plus_intensity_wrds=.;
OSG_intensity_wrds=.;
end;
if F_emp_pct>1 then F_emp_pct=.;
if fsale_sale=. then fsale_sale=0;
run;

*Create industry classifications;
data c.avgftr6;
set c.avgftr5;
if avg_emp=. then avg_emp=0;
*SIC industry categories;
sic1=input(substr(sic,1,1),12.);
sic4=input(substr(sic,1,4),12.);
if sic1=2 | sic1=3 | sic1=5 then sicindcat=0;   *Manufacturing & Trade;
if sic1=7 | sic1=8 then sicindcat=1;                  *Service;
      if sic in
(4841,4899,7310,7311,7323,7330,7331,7361,7363,7370,7371,7372,7373,
      7374,7380,7385,7389,7812,7819,8111,8200,8700,8711,8721,8731,8734,8741,8
742) then service_shift_potential_sic4=1;
if sicindcat=. & sic1 ne . then sicindcat=2;    *Other industries;
label sicindcat = "SIC Industry Category";
*Historical SIC industry categories;
sich1=floor(sich/1000);
if sich1=2 | sich1=3 | sich1=5 then sichindcat=0;     *Manufacturing & Trade;
if sich1=7 | sich1=8 then sichindcat=1;                *Service;
      if sic in
(4841,4899,7310,7311,7323,7330,7331,7361,7363,7370,7371,7372,7373,
      7374,7380,7385,7389,7812,7819,8111,8200,8700,8711,8721,8731,8734,8741,8
742) then service_shift_potential_sic4=1;
if sichindcat=. & sich1 ne . then sichindcat=2;       *Other industries;
label sichindcat = "SIC(h) Industry Category";
*Identify firms with SIC different from SICH;
if not missing (sicindcat+sichindcat) then do;
      if sicindcat ne sichindcat then SIC_off=1;
```

```
      else SIC_off=0;
      end;
label SIC_off = "Discrepency between SIC & SICH";
*NAICS industry categories;
naics2=input(substr(naics,1,2),12.);
naics3=input(substr(naics,1,3),12.);
if naics2>=31 & naics2<=45  then naicsindcat=0;
      *Manufacturing & Trade;
if naics2=51 | (naics2>=53 & naics2<=81)  then naicsindcat=1;     *Service;
      if naics in
(511120,511210,512110,512191,515120,515210,517312,517410,517919,518210,519130
,541214,541219,541330,
      541380,54151,541511,541512,541513,541519,541612,541614,541620,54171,541
710,541711,541712,5418,541810,541860,541870,
      541910,541990,561110,5613,561311,561320,5614,561422,561450,561499,56151
0,561990,611420,611430,611691)
      then service_shift_potential_naics6=1;
if naicsindcat=. then naicsindcat=2;
      *Other industries;
*GICS industry categories;
gics4=input(ggroup,12.);
if gind in
(151010,151010,151010,151010,151010,151020,151030,151030,151050,151050,201010
,201020,201030,201040,
201040,201050,201060,201060,201060,201070,251010,251010,251020,252010,252010,
252010,252010,252010,252020,252030,
252030,252030,255010,255020,255030,255040,255040,255040,255040,255040,255040,
301010,301010,301010,301010,302010,
302010,302010,302020,302020,302030,303010,303020,452010,452010,452020,452020,
452020,452030,452030,452030,452030,
452050,452050,453010,453010) then gicsindcat=0;
      *Manufacturing & Trade;
if gind in
(202010,202010,202010,202010,202010,202010,202020,202020,253010,253010,253010
,253020,253020,351010,
351010,351020,351020,351020,351030,352010,352020,352030,451010,451020,451020,
451020,451030,451030,501010,501010,
502010,502010,502010,502020,502020,502030) then gicsindcat=1;     *Service;
      if gsubind in
(20201030,20201070,20202010,20202020,25302010,35102015,35103010,35201010,3520
2010,35203010,
      45101010,45102010,45102020,45102030,45103010,45103020,45203020,50101010
,50101020,50201010,50201020,
      50202010,50202020,50203010) then service_shift_potential_gics8=1;
if gicsindcat=. then gicsindcat=2;
*FAANG stocks;
if cik='0001326801' | cik='0001018724' | cik='0000320193' | cik='0001065280'
| cik='0001652044' then FAANG=1;
else FAANG=0;
*Dyreng et al. top 30 firms (Table 9 - in order);
if cik='0000320193' | cik='0001652044' | cik='0000034088' | cik='0001326801'
| cik='0001551152' |
      cik='0000789019' | cik='0000050863' | cik='0001341439' |
cik='0000077476' | cik='0000882095' |
      cik='0000051143' | cik='0000078003' | cik='0000200406' |
cik='0001300514' | cik='0000021344' |
```

```
      cik='0000804328' | cik='0001645590' | cik='0000858877' |
cik='0000040545' | cik='0000063908' |
    cik='0001618921' | cik='0000320187' | cik='0000310158' | cik='0001075531'
| cik='0000047217' |
      cik='0000816284' | cik='0000318154' | cik='0000021665' |
cik='0001045810' | cik='0000014272' then top30=1;
else top30=0;
zip=substr(addzip,1,5)+1-1;
drop addzip;
if zip
in(93635 ,94022 ,94024 ,94028 ,94035 ,94040 ,94041 ,94043 ,94085 ,94086 ,9408
7 ,94089 ,94301 ,94303 ,94304 ,94305 ,
94306 ,94550 ,95002 ,95008 ,95013 ,95014 ,95020 ,95023 ,95030 ,95032 ,95033 ,
95035 ,95037 ,95046 ,95050 ,95051 ,
95053 ,95054 ,95070 ,95101 ,95110 ,95111 ,95112 ,95113 ,95116 ,95117 ,95118 ,
95119 ,95120 ,95121 ,95122 ,95123 ,
95124 ,95125 ,95126 ,95127 ,95128 ,95129 ,95130 ,95131 ,95132 ,95133 ,95134 ,
95135 ,95136 ,95138 ,95139 ,95140 ,
95141 ,95148 ,95192 ,94023 ,94039 ,94042 ,94088 ,94302 ,94309 ,95009 ,95011 ,
95015 ,95021 ,95026 ,95031 ,95036 ,
95038 ,95042 ,95044 ,95052 ,95055 ,95056 ,95071 ,95103 ,95106 ,95108 ,95109 ,
95115 ,95150 ,95151 ,95152 ,95153 ,
95154 ,95155 ,95156 ,95157 ,95158 ,95159 ,95160 ,95161 ,95164 ,95170 ,95172 ,
95173 ,95190 ,95191 ,95193 ,95194 ,
95196 ,94002 ,94005 ,94010 ,94014 ,94015 ,94019 ,94020 ,94021 ,94025 ,94027 ,
94028 ,94030 ,94038 ,94044 ,94060 ,
94061 ,94062 ,94063 ,94065 ,94066 ,94070 ,94074 ,94080 ,94128 ,94303 ,94401 ,
94402 ,94403 ,94404 ,94011 ,94016 ,
94017 ,94018 ,94026 ,94037 ,94064 ,94083 ,94497 ,94102 ,94103 ,94104 ,94105 ,
94107 ,94108 ,94109 ,94110 ,94111 ,
94112 ,94114 ,94115 ,94116 ,94117 ,94118 ,94121 ,94122 ,94123 ,94124 ,94127 ,
94129 ,94130 ,94131 ,94132 ,94133 ,
94134 ,94158 ,94188 ,94119 ,94120 ,94125 ,94126 ,94140 ,94141 ,94142 ,94146 ,
94147 ,94159 ,94164 ,94172 ,94137 ,
94139 ,94143 ,94144 ,94145 ,94151 ,94153 ,94154 ,94156 ,94160 ,94161 ,94162 ,
94163 ,94171 ,94177 ,94501 ,94502 ,
94514 ,94536 ,94538 ,94539 ,94541 ,94542 ,94544 ,94545 ,94546 ,94550 ,94551 ,
94552 ,94555 ,94560 ,94566 ,94568 ,
94577 ,94578 ,94579 ,94580 ,94583 ,94586 ,94587 ,94588 ,94601 ,94602 ,94603 ,
94605 ,94606 ,94607 ,94608 ,94609 ,
94610 ,94611 ,94612 ,94613 ,94615 ,94617 ,94618 ,94619 ,94621 ,94702 ,94703 ,
94704 ,94705 ,94706 ,94707 ,94708 ,
94709 ,94710 ,94720 ,95377 ,95391 ,94537 ,94540 ,94543 ,94557 ,94604 ,94614 ,
94620 ,94623 ,94624 ,94661 ,94662 ,
94701 ,94712 ,94622 ,94649 ,94659 ,94660 ,94666 ,94505 ,94506 ,94507 ,94509 ,
94513 ,94514 ,94517 ,94518 ,94519 ,
94520 ,94521 ,94523 ,94525 ,94526 ,94528 ,94530 ,94531 ,94547 ,94548 ,94549 ,
94551 ,94553 ,94556 ,94561 ,94563 ,
94564 ,94565 ,94569 ,94572 ,94582 ,94583 ,94595 ,94596 ,94597 ,94598 ,94611 ,
94707 ,94708 ,94801 ,94803 ,94804 ,
94805 ,94806 ,94850 ,95206 ,95219 ,94511 ,94516 ,94522 ,94524 ,94527 ,94570 ,
94575 ,94802 ,94807 ,94808 ,94820 ,
94529) then SiliconValley=1; else SiliconValley=0;
run;

%winsor(dsetin=c.avgftr6, dsetout=c.avgftr7, byvar=none,
vars= AvgFROS avgDROS AvgROS AvgFsale AvgDsale
```

```
        avgrd_sale avgrd_at avgads_sale avgads_at avgintan_sale avgintan_at
avgsize avgcash_at avgdebt_at
        Fgrowth avg_grant_count_per_year5
        invfg_at invt_at avg_emp cogs_sale cogs_at cogs_exp pay
        fsale_annual fsale_sale_annual emp_vol avg_similarity, type=winsor,
pctl=1 99);

proc export data=c.avgftr7 dbms=dta
outfile="_path_" replace;
quit;

*Create a sample of outbound shifters only;
data avgftr8;
set c.avgftr6;
if lowavgftr=1;
run;

%winsor(dsetin=avgftr8, dsetout=c.avgftr8, byvar=none,
vars= AvgFROS avgDROS AvgROS AvgFsale AvgDsale
        avgrd_sale avgrd_at avgads_sale avgads_at avgintan_sale avgintan_at
avgsize avgcash_at avgdebt_at
        Fgrowth avg_grant_count_per_year5
        invfg_at invt_at avg_emp cogs_sale cogs_at cogs_exp pay
        fsale_annual fsale_sale_annual emp_vol avg_similarity, type=winsor,
pctl=1 99);

proc export data=c.avgftr8 dbms=dta
outfile="_path_" replace;
quit;
```