GRA 19703 - Master Thesis

BI Norwegian Business School, Oslo

Programme: Master of Science in Business, Major in Finance

Hand-in date: 01st July 2019

Master Thesis codes

- Role of the Board and the case of special meetings -

Laura Sešek

Martin Standstå

Supervisor:
Salvatore Miglietta

1. *Stata code*

You can see the Stata code for generating a dummy variable for dual shares, performing summary statistics, correlation matrix and regression model (The SMM Model constructed during our research).

```
import excel "/Users/laurasesek/Downloads/Laura.xlsx", sheet("Sheet1")
cellrange(A1:E136) firstrow
```

Generating Dummy variable for Dual shares:

```
gen d=1
replace d=0 if dualshares==0
label variable d "Dual shares (1=no 0=yes)"
gen dalt=1
replace dalt=0 if dualshares==1
tab d, sum(specialmeeting)
```

Summary statistics (Mean, St. deviation, Min, Max)

```
summarize
```

Summary statistics on Kurtosis, Skewness

```
summarize, detail
```

Correlation Matrix

```
pwcorr specialmeeting ipo shareholders insholding d, sig
```

Classical Linear Regression Model:

```
regress specialmeeting ipo shareholders insholding d
```

2. *Matlab code*

You can see the Matlab code for model diagnostics, with which we tested the model for normality of disturbances (Jarque-Bera test), multicollinearity (VIF), heteroscedasticity (White test) and autocorrelation (Breusch-Godfrey test).

```
clc; clearvars; close all;

%% Importing data
SP = readtable ('SMM.xlsx','Sheet','Sheet1','Range','A1:E101');

%% Fitlm

% Fitlm was used to check the results
% and stored all the outputs in unr_regr.SSE.
alpha = 0.05;
regression     =     regstats(SP.specialMeetingThreshold,[SP.ipo     SP.dualShares
SP.shareholders SP.insholding]);
u = regression.r;
input=SP(:,[1:5]);
unr_regr        =        fitlm([SP.ipo        SP.dualShares        SP.shareholders
SP.insholding],SP.specialMeetingThreshold);
URSS1 = unr_regr.SSE;
T = unr_regr.NumObservations;
k = unr_regr.NumEstimatedCoefficients;

%% Multicollinearity: Variance Inflation factor (VIF)

Mult = [SP.ipo, SP.dualShares, SP.shareholders, SP.insholding];
M1 = table(SP.ipo, SP.dualShares, SP.shareholders, SP.insholding);
M1.Properties.VariableNames ={'ipo', 'dS' ,'sh', 'ih'};
corrplot(M1)
correlations = corrcoef(Mult);


%% First Classical Linear Regression Model Assumption: Jarque-Bera test for normality of the disturbances


skew_Jarque_Bera_test = mean(u.^3) / mean(u.^2).^(3/2);
kurt_Jarque_Bera_test= mean(u.^4) / mean(u.^2).^2;
test_stat_Jarque_Bera_test    =    T    *    (    skew_Jarque_Bera_test^2    /    6    +
(kurt_Jarque_Bera_test - 3)^2 / 24);
crit_val_Jarque_Bera_test = chi2inv(1-1/2*alpha, 2);
p_Jarque_Bera_test = 1 - cdf('Chisquare', test_stat_Jarque_Bera_test, 2);
reject_Jarque_Bera_stest = test_stat_Jarque_Bera_test > crit_val_Jarque_Bera_test;


%% Second Classical Linear Regression Model Assumption: White test for Homoscedasticity

ResRaw = unr_regr.Residuals.Raw;
SP.ResRaw2 = u.^2;

SP2  =  table(SP.specialMeetingThreshold,SP.ipo,  SP.dualShares,  SP.shareholders,
SP.insholding,    SP.specialMeetingThreshold.^2,    SP.ipo.^2,    SP.dualShares.^2,
SP.shareholders.^2, SP.insholding.^2, SP.ResRaw2);


SP2.Properties.VariableNames={'sMT', 'ipo', 'dS', 'sh', 'is', 'sMTsq', 'iposq',
'dSsq', 'shsq','issq' ,'ResRaw2'};


WhiteReg=fitlm(SP2,'ResRaw2~ipo+dS+sh+is+iposq+dSsq+shsq+issq');
```

```
m = k - 1 ;
tstat_SA = WhiteReg.Rsquared.Ordinary*T;
cvalue_SA = chi2inv(1 - alpha , m);
pv_SA = 1 - cdf('Chisquare' , tstat_SA , m);
reject_SA = tstat_SA > cvalue_SA;
```

%% Third Classical Linear Regression Model Assumption: ***Breusch-Godfrey test for autocorrelation, up to 12 lags***

```
SP.u = u;
SP.u1 = lagmatrix(u,1);
SP.u2 = lagmatrix(u,2);
SP.u3 = lagmatrix(u,3);
SP.u4 = lagmatrix(u,4);
SP.u5 = lagmatrix(u,5);
SP.u6 = lagmatrix(u,6);
SP.u7 = lagmatrix(u,7);
SP.u8 = lagmatrix(u,8);
SP.u9 = lagmatrix(u,9);
SP.u10 = lagmatrix(u,10);
SP.u11 = lagmatrix(u,11);
SP.u12 = lagmatrix(u,12);


SP = fillmissing(SP, 'constant', 0, 'DataVariables', {'u', ...
'u1', 'u2', 'u3', 'u4', 'u5', 'u6', 'u7', 'u8', 'u9', 'u10', 'u11', 'u12'});


Breusch_Godfrey_regr = fitlm(SP, 'u ~  ipo + dualShares + shareholders + insholding
+ u1 + u2 + u3 + u4 + u5 + u6 + u7 + u8 + u9 + u10 + u11 + u12');

r = 12;
test_TA = Breusch_Godfrey_regr.Rsquared.Ordinary * (T - r);
crit_TA = chi2inv(1-1/2*alpha, r);
pTA = 1 - cdf('Chisquare', test_TA, r);
rejectTA = test_TA > crit_TA;


cg = zeros(17,1);
Hg = zeros(17);
Hg(6,6) = 1;
Hg(7,7) = 1;
Hg(8,8) = 1;
Hg(9,9) = 1;
Hg(10,10) = 1;
Hg(11,11) = 1;
Hg(12,12) = 1;
Hg(13,13) = 1;
Hg(14,14) = 1;
Hg(15,15) = 1;
Hg(16,16) = 1;
Hg(17,17) = 1;

[PTA, FTA] = coefTest(Breusch_Godfrey_regr, Hg, cg);
```

%% Fourth Classical Linear Regression Model Assumption: ***The average value of errors is 0***

```
Mean_Residuals=mean(unr_regr.Residuals.Raw);
Stdev_Residuals=std(unr_regr.Residuals.Raw);

% test statistic
tStat_FA=(Mean_Residuals)/(Stdev_Residuals/sqrt(T));
% Critical Value at 5%
CValueFA=tinv(1-1/2*alpha,T-k);
% p-value
PvFA=2*(1-tcdf(abs(tStat_FA),T-k));
```

```
% test if H0 should be rejected
rejectFA = abs(tStat_FA) > abs(CValueFA);

%% Fifth Classical Linear Regression Model Assumption: OLS estimators are consistent
and unbiased if regressors are not correlated with the error term

correlation_ipo_u = corr(SP.ipo,u);
correlation_dualShares_u = corr(SP.dualShares,u);
correlation_shareholders_u = corr(SP.shareholders,u);
correlation_insholding_u = corr(SP.insholding,u);
```