```
/*************************************************************
*************************************************************
*************************************************************
*************************************************************/

/*Daren Purnell, 2017SP_PREDICT_411-DL_SEC60*/

/* Connect Predict411 Data */
libname mydata "/scs/wtm926/" access=readonly;
proc datasets library=mydata;
run;
ods graphics on;

title 'Moneyball OLS Regression Project';

/* EXPLORATORY DATA ANALYSIS */

/* Create a shortcut for the Moneyball Data */
data m_ball;
   set mydata.MONEYBALL;
run;

/*Sort and print first 100 records */
proc sort data=m_ball;
          by Target_Wins;
run;
proc print data=m_ball (obs=10); run;

/*Identify Missing Values*/
proc means data=m_ball n nmiss; * max min mean stddev p25 p50 p75 p99;
          var Target_Wins Team_Batting_H Team_Batting_2B Team_Batting_3B
Team_Batting_HR Team_Batting_BB
   Team_Batting_HBP Team_Batting_SO Team_Baserun_SB Team_Baserun_CS
Team_Fielding_E
   Team_Fielding_DP Team_Pitching_BB Team_Pitching_H Team_Pitching_HR
Team_Pitching_SO;
run;

/* Analyze each variable*/
proc univariate normal plot data = m_ball;
   var Target_Wins;
   histogram Target_Wins/normal (color=red w=5);
   title 'Univariate Target_Wins';
run;
```

```
proc univariate normal plot data = m_ball;
   var Team_Batting_H;
   histogram Team_Batting_H/normal (color=red w=5);
   title 'Univariate Team_Batting_H: Base Hits by Batters +';
run;
proc univariate normal plot data = m_ball;
   var Team_Batting_2B;
   histogram Team_Batting_2B/normal (color=red w=5);
   title 'Univariate Team_Batting_2B: Doubles by Batters +';
run;
proc univariate normal plot data = m_ball;
   var Team_Batting_3B;
   histogram Team_Batting_3B/normal (color=red w=5);
   title 'Univariate Team_Batting_3B: Triples by Batters +';
run;
proc univariate normal plot data = m_ball;
   var Team_Batting_HR;
   histogram Team_Batting_HR/normal (color=red w=5);
   title 'Univariate Team_Batting_HR: Homeruns by Batters +';
run;
proc univariate normal plot data = m_ball;
   var Team_Batting_BB;
   histogram Team_Batting_BB/normal (color=red w=5);
   title 'Univariate Team_Batting_BB: Walks by Batters +';
run;
proc univariate normal plot data = m_ball;
   var Team_Batting_SO;
   histogram Team_Batting_SO/normal (color=red w=5);
   title 'Univariate Team_Batting_SO: Strikeouts by Batters -';
run;
proc univariate normal plot data = m_ball;
   var Team_Baserun_SB;
   histogram Team_Baserun_SB/normal (color=red w=5);
   title 'Univariate Team_Baserun_SB: Stolen Bases +';
run;
proc univariate normal plot data = m_ball;
   var Team_Baserun_CS;
   histogram Team_Baserun_CS/normal (color=red w=5);
   title 'Univariate Team_Baserun_CS: Caught Stealing -';
run;
proc univariate normal plot data = m_ball;
   var Team_Fielding_E;
   histogram Team_Fielding_E/normal (color=red w=5);
   title 'Univariate Team_Fielding_E: Fielding Errors -';
```

```
run;
proc univariate normal plot data = m_ball;
   var Team_Fielding_DP;
   histogram Team_Fielding_DP/normal (color=red w=5);
   title 'Univariate Team_Fielding_DP: Double Plays +';
run;
proc univariate normal plot data = m_ball;
   var Team_Pitching_BB;
   histogram Team_Pitching_BB/normal (color=red w=5);
   title 'Univariate Team_Pitching_BB: Walks Allowed -';
run;
proc univariate normal plot data = m_ball;
   var Team_Pitching_H;
   histogram Team_Pitching_H/normal (color=red w=5);
   title 'Univariate Team_Pitching_H: Hits Allowed -';
run;
proc univariate normal plot data = m_ball;
   var Team_Pitching_HR;
   histogram Team_Pitching_HR/normal (color=red w=5);
   title 'Univariate Team_Pitching_HR: Homeruns Allowed -';
run;
proc univariate normal plot data = m_ball;
   var Team_Pitching_SO;
   histogram Team_Pitching_SO/normal (color=red w=5);
   title 'Univariate Team_Pitching_SO: Strikeouts by Pitchers +';
run;

/* PROC CORR to produce Pearson Correlation Coefficients */
proc corr data = m_ball plots=matrix(histogram);
   title "Correlation Matrix of All Variables";
   var Team_Batting_H Team_Batting_2B Team_Batting_3B Team_Batting_HR
Team_Batting_BB
   Team_Batting_HBP Team_Batting_SO Team_Baserun_SB Team_Baserun_CS
Team_Fielding_E
   Team_Fielding_DP Team_Pitching_BB Team_Pitching_H Team_Pitching_HR
Team_Pitching_SO;
run;

/* DATA PREPARATION */

/* Impute missing values */
data impute_mb;
        set m_ball;
```

```
	drop team_batting_HBP; /* missing 90% of values so not using */

	Imp_Team_Batting_SO = Team_Batting_SO;
	F_Team_Batting_SO = 0; /* Missing value flag */
	if missing (Team_Batting_SO) then do;
		Imp_Team_Batting_SO = 735.61; /*Normal Distro; impute with mean */
		F_Team_Batting_SO = 1;
	end;
	drop Team_Batting_SO;

	Imp_Team_Baserun_SB = Team_Baserun_SB;
	F_Team_Baserun_SB = 0;
	if missing (Team_Baserun_SB)then do;
		Imp_Team_Baserun_SB = 65; /* Left-skewed; impute with mode */
		F_Team_Baserun_SB = 1;
	end;
	drop Team_Baserun_SB;

	Imp_Team_Baserun_CS = Team_Baserun_CS;
	F_Team_Baserun_CS = 0;
	if missing (Team_Baserun_CS) then do;
		Imp_Team_Baserun_CS = 52.80; /*Normal Distro; impute with mean */
		F_Team_Baserun_CS = 1;
	end;
	drop Team_Baserun_CS;
	Imp_Team_Fielding_DP = Team_Fielding_DP;
	F_Team_Fielding_DP = 0;
	if missing (Team_Fielding_DP) then do;
		Imp_Team_Fielding_DP = 146.39; /* Normal Distro; impute with mean */
		F_Team_Fielding_DP = 1;
	end;
	drop Team_Fielding_DP;
	Imp_Team_Pitching_SO = Team_Pitching_SO;
	F_Team_Pitching_SO = 0;
	if missing (Team_Pitching_SO) then do;
		Imp_Team_Pitching_SO = 817.73; /* Normal Distro; impute with mean */
		F_Team_Pitching_SO = 1;
	end;
	drop Team_Pitching_SO;
run;
/* For Rate Conversion based on 162 Game Season */

/* Data Rate Conversions */
data rate_mb;
```

```
	set impute_mb;
	R_TEAM_BATTING_H = (TEAM_BATTING_H/162); drop TEAM_BATTING_H;
	R_TEAM_BATTING_2B = (TEAM_BATTING_2B/162); drop TEAM_BATTING_2B;
	R_TEAM_BATTING_3B = (TEAM_BATTING_3B/162); drop TEAM_BATTING_3B;
	R_TEAM_BATTING_HR = (TEAM_BATTING_HR/162); drop TEAM_BATTING_HR;
	R_TEAM_BATTING_BB = (TEAM_BATTING_BB/162); drop TEAM_BATTING_BB;
	R_TEAM_PITCHING_H	= (TEAM_PITCHING_H/162); drop TEAM_PITCHING_H;
	R_TEAM_PITCHING_HR = (TEAM_PITCHING_HR/162); drop TEAM_PITCHING_HR;
	R_TEAM_PITCHING_BB = (TEAM_PITCHING_BB/162); drop TEAM_PITCHING_BB;
	R_TEAM_FIELDING_E = (TEAM_FIELDING_E/162); drop TEAM_FIELDING_E;
	R_Imp_Team_Batting_SO = (Imp_Team_Batting_SO/162); drop Imp_Team_Batting_SO;
	R_Imp_Team_Baserun_SB = (Imp_Team_Baserun_SB/162); drop
Imp_Team_Baserun_SB;
	R_Imp_Team_Baserun_CS	= (Imp_Team_Baserun_CS/162); drop
Imp_Team_Baserun_CS;
	R_Imp_Team_Fielding_DP = (Imp_Team_Fielding_DP/162); drop
Imp_Team_Fielding_DP;
	R_Imp_Team_Pitching_SO = (Imp_Team_Pitching_SO/162); drop
Imp_Team_Pitching_SO;
run;

/* Rate Data Transformations */
data R_trans_mb;
	set rate_mb;
	Trans_Team_Batting_3B = R_Team_Batting_3B;
		if		Trans_Team_Batting_3B >= (186/162)  then
Trans_Team_Batting_3B = (186/162);
		Trans_Team_Batting_3B = log(Trans_Team_Batting_3B + 1);
		drop R_TEAM_BATTING_3B;
	Trans_Team_Baserun_SB = log(R_Imp_Team_Baserun_SB + 1);
		drop R_Imp_Team_Baserun_SB;
	Trans_Team_Fielding_E = abs(log(R_Team_Fielding_E/(246.48/162))); /*Standardize
and log*/
		drop R_Team_Fielding_E;
	Trans_Team_Pitching_BB = log(R_Team_Pitching_BB/(553/162) + 1)/(0.678/162);
/*Double Standardize and log*/
		drop R_Team_Pitching_BB;
	Trans_Team_Pitching_H = R_Team_Pitching_H;
		if		Trans_Team_Pitching_H <= (627/162) then
Trans_Team_Pitching_H = (627/162);
		else if		Trans_Team_Pitching_H >= (2475/162)  then
Trans_Team_Pitching_H = (2475/162);
		Trans_Team_Pitching_H = log(Trans_Team_Pitching_H);
		drop R_Team_Pitching_H;
```

```
	Trans_Team_Pitching_SO = log(R_Imp_Team_Pitching_SO/(817.73/162) + 1);
		drop R_Imp_Team_Pitching_SO;
run;
/* Transformations without Rate Conversion */
data trans_mb;
	set impute_mb;
	Trans_Team_Batting_3B = Team_Batting_3B;
		if		Trans_Team_Batting_3B >= 186  then Trans_Team_Batting_3B =
186;
		Trans_Team_Batting_3B = log(Trans_Team_Batting_3B + 1);
	Trans_Team_Baserun_SB = log(Imp_Team_Baserun_SB + 1);
	Trans_Team_Fielding_E = abs(log(Team_Fielding_E/246.48)); /*Standardize and log*/
	Trans_Team_Pitching_BB = log(Team_Pitching_BB/553 + 1)/0.678; /*Double
Standardize and log*/
	Trans_Team_Pitching_H = Team_Pitching_H;
		if			Trans_Team_Pitching_H <= 627 then
Trans_Team_Pitching_H = 627;
		else if		Trans_Team_Pitching_H >= 2475  then Trans_Team_Pitching_H =
2475;
		Trans_Team_Pitching_H = log(Trans_Team_Pitching_H);
	Trans_Team_Pitching_SO = log(Imp_Team_Pitching_SO/817.73 + 1);
run;

/* Verifying Xfers resulted in more normality */
/* Team_Batting_3B */
proc univariate normal plot data = trans_mb;
	var Trans_Team_Batting_3B;
	histogram Trans_Team_Batting_3B/normal (color=red w=5);
run;
/* Team_Baserun_SB */
proc univariate normal plot data = trans_mb;
	var Trans_Team_Baserun_SB;
	histogram Trans_Team_Baserun_SB/normal (color=red w=5);
run;
/* Team_Fielding_E */
proc univariate normal plot data = trans_mb;
	var Trans_Team_Fielding_E;
	histogram Trans_Team_Fielding_E/normal (color=red w=5);
run;
/* Team_Pitching_BB */
proc univariate normal plot data = trans_mb;
	var Trans_Team_Pitching_BB;
	histogram Trans_Team_Pitching_BB/normal (color=red w=5);
run;
```

```
/* Team_Pitching_H */
proc univariate normal plot data = trans_mb;
        var Trans_Team_Pitching_H;
        histogram Trans_Team_Pitching_H/normal (color=red w=5);
run;
/* Team_Pitching_SO */
proc univariate normal plot data = trans_mb;
        var Trans_Team_Pitching_SO;
        histogram Trans_Team_Pitching_SO/normal (color=red w=5);
run;

proc print data = trans_mb (obs=10); run;

/* Build Models */
/* Base Model */
proc reg data = m_ball; /* Team_Batting_HBP Team_Batting_SO Team_Baserun_SB
Team_Baserun_CS
Team_Fielding_DP Team_Pitching_SO not included due to missing records Adj_Rsq = 0.2674 */
        title 'Base Model ALL Variables with NO MISSING VALUES';
        model Target_Wins = Team_Batting_H Team_Batting_2B Team_Batting_3B
Team_Batting_HR Team_Batting_BB
  Team_Fielding_E  Team_Pitching_BB Team_Pitching_H Team_Pitching_HR / vif;
run;
/* Impute Values Model*/
proc reg data = impute_mb;
        title 'Base Imputed Model'; /* 0.4069 */
        model Target_Wins = Team_Batting_H Team_Batting_2B Team_Batting_3B
Team_Batting_HR Team_Batting_BB
  Imp_Team_Batting_SO F_Team_Batting_SO Imp_Team_Baserun_SB F_Team_Baserun_SB
Imp_Team_Baserun_CS
  F_Team_Baserun_CS Team_Fielding_E Imp_Team_Fielding_DP F_Team_Fielding_DP
  Team_Pitching_BB Team_Pitching_H Team_Pitching_HR Imp_Team_Pitching_SO
F_Team_Pitching_SO / vif;
run;
/* Transformed and Imputed using single instances of the varibles */
proc reg data = trans_mb; /* Adj R_sq 0.34 */
        title 'Single Instance Stepwise Selection Transformed and Imputed Value Model';
        model Target_Wins = TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_HR
        TEAM_BATTING_BB TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB
        Imp_Team_Batting_SO          F_Team_Batting_SO Imp_Team_Baserun_SB
F_Team_Baserun_SB  Imp_Team_Baserun_CS
        F_Team_Baserun_CS Imp_Team_Fielding_DP F_Team_Fielding_DP
Imp_Team_Pitching_SO F_Team_Pitching_SO
```

```
	Trans_Team_Batting_3B Trans_Team_Baserun_SB  Trans_Team_Fielding_E
Trans_Team_Pitching_BB
	Trans_Team_Pitching_H Trans_Team_Pitching_SO/vif selection = stepwise;
run;

/* Transformed & Imputed: Automated Variable Selection */
proc reg data = trans_mb; /* Adj R_sq 0.4198 */
	title 'Stepwise Selection Transformed and Imputed Value Model';
	model Target_Wins = TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B
TEAM_BATTING_HR
	TEAM_BATTING_BB TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB
TEAM_FIELDING_E
	Imp_Team_Batting_SO	F_Team_Batting_SO Imp_Team_Baserun_SB
F_Team_Baserun_SB  Imp_Team_Baserun_CS
	F_Team_Baserun_CS Imp_Team_Fielding_DP F_Team_Fielding_DP
Imp_Team_Pitching_SO F_Team_Pitching_SO
	Trans_Team_Batting_3B Trans_Team_Baserun_SB  Trans_Team_Fielding_E
Trans_Team_Pitching_BB
	Trans_Team_Pitching_H Trans_Team_Pitching_SO/vif selection = stepwise;
run;
proc reg data = trans_mb; /* Adj R_sq 0.4216 */
	title 'Forward Selection Transformed and Imputed Value Model';
	model Target_Wins = TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B
TEAM_BATTING_HR
	TEAM_BATTING_BB TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB
TEAM_FIELDING_E
	Imp_Team_Batting_SO	F_Team_Batting_SO Imp_Team_Baserun_SB
F_Team_Baserun_SB  Imp_Team_Baserun_CS
	F_Team_Baserun_CS Imp_Team_Fielding_DP F_Team_Fielding_DP
Imp_Team_Pitching_SO F_Team_Pitching_SO
	Trans_Team_Batting_3B Trans_Team_Baserun_SB  Trans_Team_Fielding_E
Trans_Team_Pitching_BB
	Trans_Team_Pitching_H Trans_Team_Pitching_SO/vif selection = forward;
run;
proc reg data = trans_mb; /* Adj R_sq 0.4212 */
	title 'Backward Selection Transformed and Imputed Value Model';
	model Target_Wins = TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B
TEAM_BATTING_HR
	TEAM_BATTING_BB TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB
TEAM_FIELDING_E
	Imp_Team_Batting_SO	F_Team_Batting_SO Imp_Team_Baserun_SB
F_Team_Baserun_SB  Imp_Team_Baserun_CS
	F_Team_Baserun_CS Imp_Team_Fielding_DP F_Team_Fielding_DP
Imp_Team_Pitching_SO F_Team_Pitching_SO
```

```
	Trans_Team_Batting_3B Trans_Team_Baserun_SB  Trans_Team_Fielding_E
Trans_Team_Pitching_BB
	Trans_Team_Pitching_H Trans_Team_Pitching_SO/vif selection = backward;
run;
/* Rate Based Models */
/* Rate Models 0.3439 */
proc reg data = R_trans_mb;
	title "Rate Model";
	model Target_Wins = F_Team_Batting_SO F_Team_Baserun_SB    F_Team_Baserun_CS
F_Team_Fielding_DP
	F_Team_Pitching_SO R_TEAM_BATTING_H  R_TEAM_BATTING_2B
R_TEAM_BATTING_HR          R_TEAM_BATTING_BB
	R_TEAM_PITCHING_HR R_Imp_Team_Batting_SO R_Imp_Team_Baserun_CS
R_Imp_Team_Fielding_DP
	Trans_Team_Batting_3B Trans_Team_Baserun_SB
	Trans_Team_Fielding_E Trans_Team_Pitching_BB Trans_Team_Pitching_H
Trans_Team_Pitching_SO;
run;
/* Rate Model Using Stepwise */
proc reg data = R_trans_mb;
	title "Rate Model: Stepwise Adj R-Sq 0.3442";
	model Target_Wins = F_Team_Batting_SO F_Team_Baserun_SB    F_Team_Baserun_CS
F_Team_Fielding_DP
	F_Team_Pitching_SO R_TEAM_BATTING_H  R_TEAM_BATTING_2B
R_TEAM_BATTING_HR          R_TEAM_BATTING_BB
	R_TEAM_PITCHING_HR R_Imp_Team_Batting_SO
	R_Imp_Team_Baserun_CS R_Imp_Team_Fielding_DP Trans_Team_Batting_3B
Trans_Team_Baserun_SB
	Trans_Team_Fielding_E Trans_Team_Pitching_BB Trans_Team_Pitching_H
Trans_Team_Pitching_SO
	/vif selection = stepwise adjrsq aic bic;
run;

/* Bonus */
proc glm data = trans_mb;
	model Target_Wins = TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B
	TEAM_BATTING_BB TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_FIELDING_E
	Imp_Team_Batting_SO Imp_Team_Baserun_SB F_Team_Baserun_SB
	Imp_Team_Fielding_DP F_Team_Fielding_DP F_Team_Pitching_SO
	Trans_Team_Baserun_SB Trans_Team_Fielding_E Trans_Team_Pitching_H;
run;

proc genmod data = trans_mb;
	model Target_Wins = TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B
```

```
        TEAM_BATTING_BB TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_FIELDING_E
        Imp_Team_Batting_SO Imp_Team_Baserun_SB F_Team_Baserun_SB
        Imp_Team_Fielding_DP F_Team_Fielding_DP F_Team_Pitching_SO
        Trans_Team_Baserun_SB Trans_Team_Fielding_E Trans_Team_Pitching_H
        / link=identity dist=normal;
run;
```