```
/*data dating;
        infile "/home/u63760461/STA512/data/Speed Dating Original.csv";
run;*/

*pull in data from work - import code in data import file. create
smrace and closea variables;

data dating;
        set dating;
        if upcase(racem) eq upcase(racef) then smrace = 1;
                else smrace = 0;
        if abs(agem - agef) le 2 then closea = 1;
                else closea = 0;
run;

*check for missing val & outliers;
proc means data = dating NMISS N MISSING;
title "Table 1: Missing Values";
run;

proc means data = dating;
run;

*-----------------------------------------------;
*histograms of data distributon;

title "Distribution of Attractiveness Ranking (Split by Gender)";
proc sgplot data = dating;
        histogram attractivem / binwidth=1 transparency = 0.5
                name = "Attractive_Male" legendlabel = "Attractiveness Rating of
Partner (Male)";
        histogram attractivef / binwidth=1 transparency = 0.5
                name = "Attractive_Female" legendlabel = "Attractiveness Rating of
Partner (Female)";
        density attractivem / type = kernel lineattrs = graphdata1;
        density attractivef / type =  kernel lineattrs = graphdata2;
        xaxis label = "Rating (1-10 scale)" min = 0 max = 11;
        yaxis max = 40;
        keylegend "Attractive_Male" "Attractive_Female" / across = 1 position =
topright location = inside;
run;


title "Distribution of Sincerity Ranking (Split by Gender)";
proc sgplot data = dating;
        histogram sincerem / binwidth=1 transparency = 0.5
                name = "Sincere_Male" legendlabel = "Sincerity Rating of Partner
(Male)";
        histogram sinceref / binwidth=1 transparency = 0.5
                name = "Sincere_Female" legendlabel = "Sincerity Rating of Partner
```

```
(Female)";
        density sincerem / type = kernel lineattrs = graphdata1;
        density sinceref / type =  kernel lineattrs = graphdata2;
        xaxis label = "Rating (1-10 scale)" min = 0 max = 11;
        yaxis max = 40;
        keylegend "Sincere_Male" "Sincere_Female" / across = 1 position = topright
location = inside;
run;

title "Distribution of Intelligence Ranking (Split by Gender)";
proc sgplot data = dating;
        histogram intelligentm / binwidth=1 transparency = 0.5
                name = "intelligent_Male" legendlabel = "Intelligence Rating of
Partner (Male)";
        histogram intelligentf / binwidth=1 transparency = 0.5
                name = "intelligent_Female" legendlabel = "Intelligence Rating of
Partner (Female)";
        density intelligentm / type = kernel lineattrs = graphdata1;
        density intelligentf / type =  kernel lineattrs = graphdata2;
        xaxis label = "Rating (1-10 scale)" min = 0 max = 11;
        yaxis max = 40;
        keylegend "intelligent_Male" "intelligent_Female" / across = 1 position =
topright location = inside;
run;

title "Distribution of Fun Ranking (Split by Gender)";
proc sgplot data = dating;
        histogram Funm / binwidth=1 transparency = 0.5
                name = "Fun_Male" legendlabel = "Fun Rating of Partner (Male)";
        histogram Funf / binwidth=1 transparency = 0.5
                name = "Fun_Female" legendlabel = "Fun Rating of Partner (Female)";
        density Funm / type = kernel lineattrs = graphdata1;
        density Funf / type =  kernel lineattrs = graphdata2;
        xaxis label = "Rating (1-10 scale)" min = 0 max = 11;
        yaxis max = 40;
        keylegend "Fun_Male" "Fun_Female" / across = 1 position = topright location
= inside;
run;

title "Distribution of Ambitiousness Ranking (Split by Gender)";
proc sgplot data = dating;
        histogram ambitiousm / binwidth=1 transparency = 0.5
                name = "ambitious_Male" legendlabel = "Ambitiousness Rating of
Partner (Male)";
        histogram ambitiousf / binwidth=1 transparency = 0.5
                name = "ambitious_Female" legendlabel = "Ambitiousness Rating of
Partner (Female)";
        density ambitiousm / type = kernel lineattrs = graphdata1;
        density ambitiousf / type =  kernel lineattrs = graphdata2;
        xaxis label = "Rating (1-10 scale)" min = 0 max = 11;
```

```
        yaxis max = 40;
        keylegend "ambitious_Male" "ambitious_Female" / across = 1 position =
topright location = inside;
run;

title "Distribution of Shared Interests Ranking (Split by Gender)";
proc sgplot data = dating;
        histogram sharedinterestsm / binwidth=1 transparency = 0.5
                name = "sharedinterests_Male" legendlabel = "Shared Interests Rating
of Partner (Male)";
        histogram sharedinterestsf / binwidth=1 transparency = 0.5
                name = "sharedinterests_Female" legendlabel = "Shared Interests
Rating of Partner (Female)";
        density sharedinterestsm / type = kernel lineattrs = graphdata1;
        density sharedinterestsf / type =  kernel lineattrs = graphdata2;
        xaxis label = "Rating (1-10 scale)" min = 0 max = 11;
        yaxis max = 40;
        keylegend "sharedinterests_Male" "sharedinterests_Female" / across = 1
position = topright location = inside;
run;


*overall summary stats for freq. of same/diff race & close/far age gap;

proc freq data = dating;
        tables smrace closea;
run;


proc corr data = dating;
        var closea likem likef;
run;

proc corr data = dating;
        var smrace likem likef;
run;

*pulling variable names into temp table for droplist use;
proc contents data = dating out = contentsm(keep=name) noprint;
run;

*pulling all "M" variables into droplist;
proc sql noprint;
        select name into :droplist separated by ' '
        from contentsm
        where upcase(name) like '%M';
quit;


*dropping all "M" variables to create "F" data set;
```

```
*Create cross val ID in f data set;
data datingf;
        set dating (drop=&droplist);
        cv_id = ranuni(2025);
run;

*----------------------------------------;
*pulling all "F" variables into droplist;
proc contents data = dating out = contentsf(keep=name) noprint;
run;

proc sql noprint;
        select name into :droplist separated by ' '
        from contentsf
        where upcase(name) like '%F';
quit;

*dropping all "F" variables to create "M" data set;
*Create cross val ID in m data set;

data datingm;
        set dating (drop = &droplist);
        cv_id = ranuni(2025);
run;


*----------------------------------------------------;
*create variable transformations;

data datingf;
        set datingf;
        attr_sinc = attractivef * sinceref;
        attr_inte = attractivef * intelligentf;
        attr_fun = attractivef * funf;
        attr_ambi = attractivef * AmbitiousF;
        attr_sint = attractivef * sharedinterestsf;
        sinc_inte = sinceref * intelligentf;
        sinc_fun = sinceref * funf;
        sinc_ambi = sinceref * ambitiousf;
        sinc_sint = sinceref * sharedinterestsf;
        inte_fun = intelligentf * funf;
        inte_ambi = intelligentf * ambitiousf;
        inte_sint = intelligentf * sharedinterestsf;
        fun_ambi = funf * ambitiousf;
        fun_sint = funf * sharedinterestsf;
        ambi_sint = AmbitiousF * sharedinterestsf;
        likef_2 = likef**2;
run;

data datingm;
```

```
        set datingm;
        attr_sinc = attractivem * sincerem;
        attr_inte = attractivem * intelligentm;
        attr_fun = attractivem * funm;
        attr_ambi = attractivem * ambitiousm;
        attr_sint = attractivem * sharedinterestsm;
        sinc_inte = sincerem * intelligentm;
        sinc_fun = sincerem * funm;
        sinc_ambi = sincerem * ambitiousm;
        sinc_sint = sincerem * sharedinterestsm;
        inte_fun = intelligentm * funm;
        inte_ambi = intelligentm * ambitiousm;
        inte_sint = intelligentm * sharedinterestsm;
        fun_ambi = funm * ambitiousm;
        fun_sint = funm * sharedinterestsm;
        ambi_sint = ambitiousm * sharedinterestsm;
         *centers linear attractive term;
        attractivem_c = attractivem - 6.6490826;
         *squares centered linear attractive term;
        attractivemc_2 = attractivem_c**2;
        att_2 = attractivem**2;
        likem_2 = likem**2;
run;

*make sure centering worked to reduce colinearity;
proc corr data = dm_train;
        var attractivem att_2;
run;

proc corr data = dm_train;
        var attractivem_c attractivemc_2;
run;

*-------------------------------------------------------;
*sort data by CV ID & then cut into 2 training & test data sets (M);

proc sort data = datingm;
        by cv_id;
run;

data dm_test dm_train;
        set datingm;
        counter = _N_;
        if counter le 55 then output dm_test;
        else output dm_train;
run;

*sort data by CV ID & then cut into 2 training & test data sets(F);

proc sort data = datingf;
```

```
        by cv_id;
run;


data df_test df_train;
        set datingf;
        counter = _N_;
        if counter le 55 then output df_test;
        else output df_train;
run;


*-------------------------------------------------------;

*Creating initial male model on training data;

title "Male Model Residuals";
proc reg data = dm_train PLOTS;
        model likem = attractivem_c sincerem intelligentm funm ambitiousm
sharedinterestsm attr_sinc
        attr_inte attr_fun attr_ambi attr_sint sinc_inte sinc_fun sinc_ambi
sinc_sint
        inte_fun inte_ambi inte_sint fun_ambi fun_sint ambi_sint attractivemc_2 /
selection = stepwise sls = 0.05 sle = 0.05;
        run;


proc reg data = dm_train PLOTS;
        model likem = attractivem_c sincerem fun_sint attractivemc_2 / vif
collinoint;
        output out = male_train_resid p = p residual = ei rstudent = jackknife
student = student h = lev cookd = cook;
run;

*initial male model residual tests & normality tests;

proc freq data = male_train_resid;
        tables jackknife;
        where abs(jackknife) >2;
run;

proc freq data = male_train_resid;
        tables lev;
        where lev > 2*(5/221);
run;

proc freq data = male_train_resid;
        tables cook;
        where cook > 1;
run;
```

```
proc univariate data = male_train_resid normal plots;
        var jackknife;
run;


*creating modified male model after residual tests indicated need for transformation
due to poor normality;

proc reg data = dm_train PLOTS;
        model likem_2 = attractivem_c sincerem intelligentm funm ambitiousm
sharedinterestsm attr_sinc
        attr_inte attr_fun attr_ambi attr_sint sinc_inte sinc_fun sinc_ambi
sinc_sint
        inte_fun inte_ambi inte_sint fun_ambi fun_sint ambi_sint attractivemc_2 /
selection = stepwise sls = 0.05 sle = 0.05;
        output out = male_train_resid_3 p = p residual = ei rstudent = jackknife
student = student h = lev cookd = cook;
run;

proc reg data = dm_train;
        model likem_2 = funm attr_sinc attr_sint / vif collinoint;
        output out = male_train_resid_2 p = p residual = ei rstudent = jackknife
student = student h = lev cookd = cook;
        run;

title "Distribution of updated male model";
proc univariate data = male_train_resid_2 normal plots;
        var jackknife;
run;

proc freq data = male_train_resid_2;
        tables jackknife;
        where abs(jackknife) >2;
run;

*corrected male model residual tests & normality tests;

proc freq data = male_train_resid_2;
        tables lev;
        where lev > 2*(5/221);
run;

proc freq data = male_train_resid_2;
        tables cook;
        where cook > 1;
run;


*residual plots for refined & original male models;
```

```
proc sgplot data = male_train_resid_2;
        scatter x = p y = jackknife;
        refline 0 / axis = y;
        title "Residual Plot for Refined Male Model";
run;

proc sgplot data = male_train_resid;
        scatter x = p y = jackknife;
        refline 0 / axis = y;
        title "Residual Plot for Initial Male Model";
run;

*cross validation of male model on test data;

data dm_test;
        set dm_test;
        y_hat = 1.87367*funm + 0.39251*attr_sinc + 0.41240*attr_sint - 2.73330;
run;

proc corr data = dm_test;
        var y_hat likem_2;
run;

*-------------------------------------------;

*Creating initial female model on training data;

proc means data = df_train;
run;

title "Female Model Residuals";
proc reg data = df_train PLOTS;
        model likef = attractivef sinceref intelligentf funf ambitiousf
sharedinterestsf attr_sinc
        attr_inte attr_fun attr_ambi attr_sint sinc_inte sinc_fun sinc_ambi
sinc_sint
        inte_fun inte_ambi inte_sint fun_ambi fun_sint ambi_sint / selection =
stepwise sls = 0.05 sle = 0.05;
run;


proc reg data = df_train PLOTS;
        model likef = intelligentf attr_fun sinc_sint/ vif collinoint;
        output out = female_train_resid p = p residual = ei rstudent = jackknife
student = student h = lev cookd = cook;
run;

*initial female model residual tests & normality tests;
```

```
proc freq data = female_train_resid;
        tables jackknife;
        where abs(jackknife) >2;
run;

proc freq data = female_train_resid;
        tables lev;
        where lev > 2*(4/221);
run;

proc freq data = female_train_resid;
        tables cook;
        where cook > 1;
run;

proc univariate data = female_train_resid normal plots;
        var jackknife;
run;

*residual plots for original female models;

proc sgplot data = female_train_resid;
        scatter x = p y = jackknife;
        refline 0 / axis = y;
        title "Residual Plot for Intial Female Model";
run;

*creating modified female model after residual tests indicated need for
transformation due to poor normality;

proc reg data = df_train PLOTS;
        model likef_2 = attractivef sinceref intelligentf funf ambitiousf
sharedinterestsf attr_sinc
        attr_inte attr_fun attr_ambi attr_sint sinc_inte sinc_fun sinc_ambi
sinc_sint
        inte_fun inte_ambi inte_sint fun_ambi fun_sint ambi_sint / selection =
stepwise sls = 0.05 sle = 0.05;
run;

proc reg data = df_train;
        model likef_2 = attr_fun sinc_inte inte_sint / vif collinoint;
        output out = female_train_resid_2 p = p residual = ei rstudent = jackknife
student = student h = lev cookd = cook;
        run;


*revised female model residual tests & normality tests;

proc freq data = female_train_resid_2;
        tables jackknife;
```

```
        where abs(jackknife) >2;
run;

proc freq data = female_train_resid_2;
        tables lev;
        where lev > 2*(4/221);
run;

proc freq data = female_train_resid_2;
        tables cook;
        where cook > 1;
run;

proc univariate data = female_train_resid_2 normal plots;
        var jackknife;
run;

*residual plots for revised female models;

proc sgplot data = female_train_resid_2;
        scatter x = p y = jackknife;
        refline 0 / axis = y;
        title "Residual Plot for Revised Female Model";
run;


*cross validation of male model on test data;

data df_test;
        set df_test;
        y_hat = 0.49207*attr_fun + 0.12500*sinc_inte + 0.28038*inte_sint + 2.72659;
run;

proc corr data = df_test;
        var y_hat likef_2;
run;
```