SPEED DATING EXPERIMENT

What are the least desirable attributes in a male partner?

Speed Dating Experiment

- Columbia Business School.
- Experimental speed dating events from 2002-2004.
- 8378 dates.
- 4 minute dates with every other participant of the opposite sex.
- Participants were asked **if they would like to see their date again** (decision) and to **rate their date on 6 attributes** (0 to 10):

  1. Attractiveness
  2. Sincerity
  3. Intelligence
  4. Fun
  5. Ambition
  6. Shared Interests.

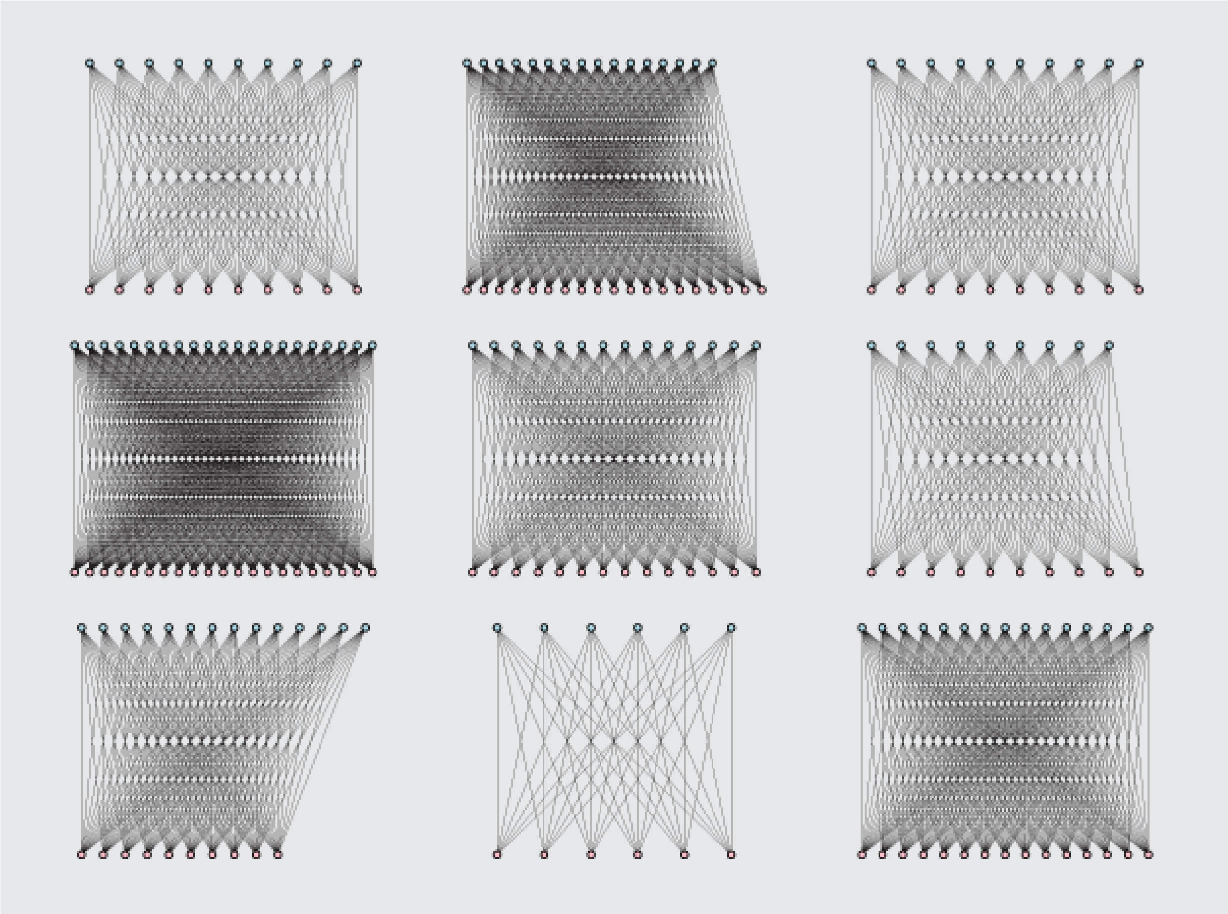
DATASET

- **Key Challenge: Dropping columns to make it more manageable**.
  I decided to reduce it from 195 to less than 15 columns.

- **Missing values and holes in the data**.
  If a column has many missing values,
  I decided to drop it so it doesn't bias the result.

- **The column names were hard to read (amb7_2).**
  I decided to rename some of them to something more readable to make analyzing the data simpler.

How the speed dating rounds worked



1) There are two groups.

2) One group is women and the other is men.

3) The point of it all is to match every woman with every man for a short period of time so that **by the end, every one has gotten a chance to quickly know each other.**

The assumption here: it is possible to learn a lot about a person in a short period of time.

PARTICIPANTS

Total: 551

	MALE	FEMALE
Participants	277 (50,3%)	274 (49.7%)
Avg. dates	15.1	15.2
Match rate	20.2%	20.9%
Partner wanted a date	39.7%	48.6%

Number of unique participants

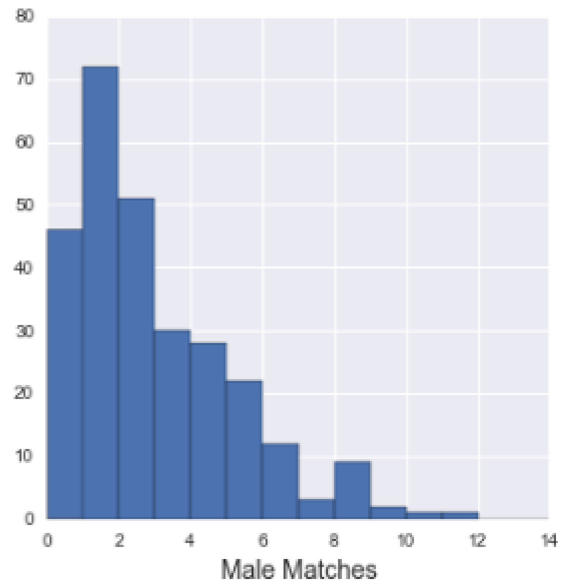
In [60]:
```
f = data_df.loc[data_df.gender == 0]
m = data_df.loc[data_df.gender == 1]
print('The total number of dates is: {}'.format(len(data_df)))
print('The total number of unique participants is: {}'.format(len(data_df['iid'].unique())))
print('The number of female participants is: {}'.format(len(f['iid'].unique())))
print('The number of male participants is: {}'.format(len(m['iid'].unique())))
```

```
The total number of dates is: 8378
The total number of unique participants is: 551
The number of female participants is: 274
The number of male participants is: 277
```

Number of match per gender

In [65]:
```
data_m = data_df[data_df['gender']==1].groupby('iid').sum()
match_m = data_m['match']
g = plt.figure(figsize=(5,5))
g = plt.hist(match_m, range(15))
g = plt.xlabel('Male Matches', fontsize=14)

# Number of dates males
dates_male = data_df[data_df.gender == 1].groupby('iid').apply(len)

# The of matches males
matches_m = data_df[data_df.match == 1]
matches_male = matches_m[matches_m.gender == 1].groupby('iid').apply(len)

#Male match percentage
mmp = (matches_male / dates_male).mean() * 100.0
mmp
print('The avg. dates per male is: %s' %(dates_male.mean()))
print('The match percentage for males is : %s' % (mmp))

# Date? decision of partner == 1
partner_yes_M = data_df[data_df.dec_o == 1]
partner_syes_M = partner_yes_M[partner_yes_M.gender == 1].groupby('iid').apply(len)
pyp_M = ((partner_syes_M/dates_male).mean())*100
print('Female partner said yes %s percent of the times' % (pyp_M))
```

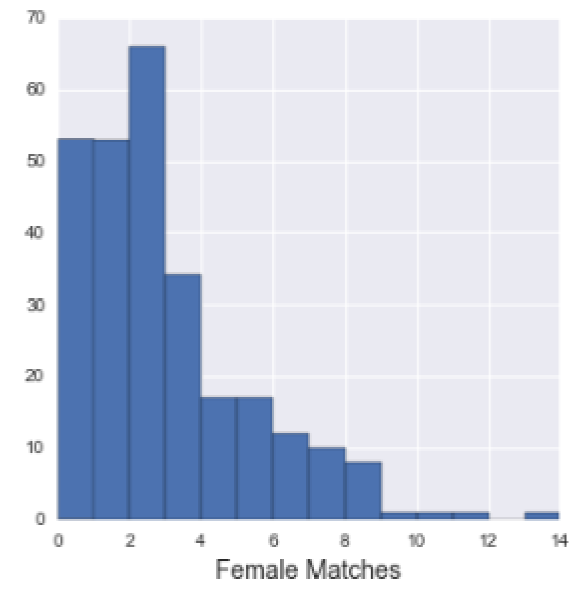
```
The avg. dates per male is: 15.1407942238
The match percentage for males is : 20.2888324872
Female partner said yes 39.7366771615 percent of the times
```



In [66]:

```python
data_f = data_df[data_df['gender']==0].groupby('iid').sum()
match_f = data_f['match']
g = plt.figure(figsize=(5,5))
g = plt.hist(match_f, range(15))
g = plt.xlabel('Female Matches', fontsize=14)

# Number of dates Females
dates_female = data_df[data_df.gender == 0].groupby('iid').apply(len)
# The of matches Females
matches = data_df[data_df.match == 1]
matches_female = matches[matches.gender == 0].groupby('iid').apply(len)
#Female match percentage
fmp = (matches_female / dates_female).mean() * 100.0
fmp
print('The avg. dates per female is: %s' %(dates_female.mean()))
print('The match percentage for females is : %s' % (fmp))

# Date? decision of partner == 1
partner_yes_F = data_df[data_df.dec_o == 1]
partner_syes_F = partner_yes_F[partner_yes_F.gender == 0].groupby('iid').apply(len)
pyp_F = ((partner_syes_F/dates_female).mean())*100
print('Male partner said yes %s percent of the times' % (pyp_F))
```

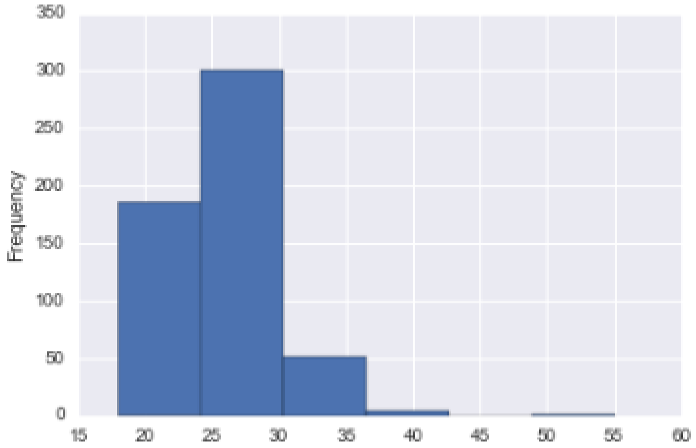
```
The avg. dates per female is: 15.2700729927
The match percentage for females is : 20.9103753144
Male partner said yes 48.6359814379 percent of the times
```



Participant's age distribution

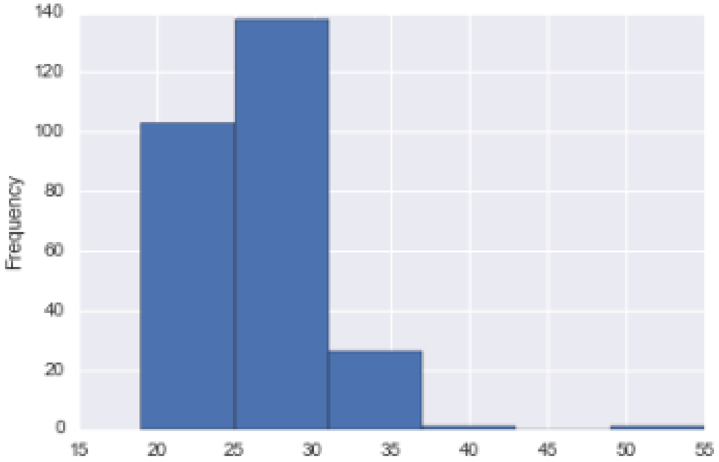
```
data_unique = data_df.groupby('iid').mean()
data_unique.age.plot(kind='hist', bins=6)
print("participant's age distribution")
```

```
participant's age distribution
```



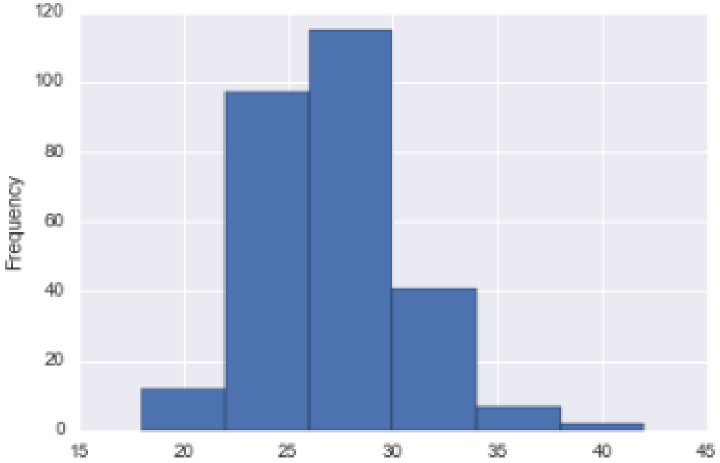

```
data_unique_f = data_df[data_df['gender']==0].groupby('iid').mean()
data_unique_f.age.plot(kind='hist', bins=6)
print("Females age distribution")
```

```
Females age distribution
```



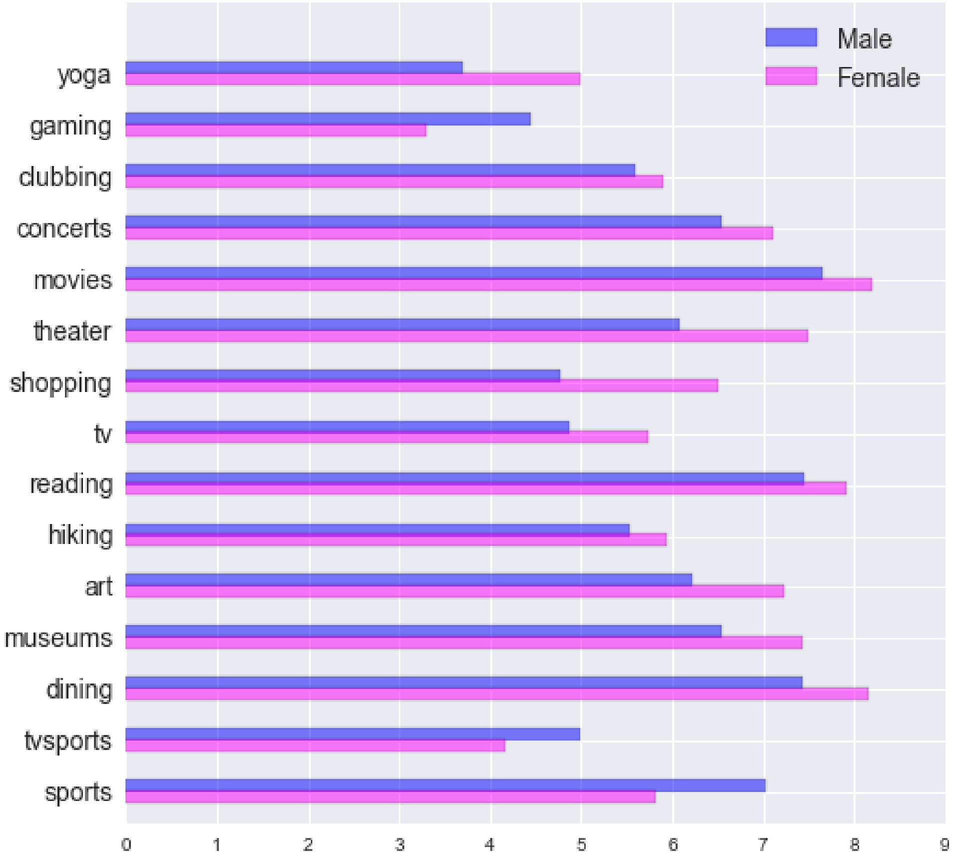

```
data_unique_m = data_df[data_df['gender']==1].groupby('iid').mean()
data_unique_m.age.plot(kind='hist', bins=6)
print("Males age distribution")
```

```
Males age distribution
```



Participant's activities (Interest)

```
activities_interested=['sports','tvsports', 'dining', 'museums', 'art', 'hiking', 'reading', 'tv', 'shopping',
                       'theater','movies', 'concerts', 'clubbing', 'gaming', 'yoga']
temp = data_df.groupby(['gender']).mean()[activities_interested].values

g = plt.figure(figsize=(8,8))
g = plt.barh(np.arange(0,2*temp.shape[1],2)+0.2,temp[1,:], height=0.5,color=[0,0,1],alpha=0.5,label='Male')
g = plt.barh(np.arange(0,2*temp.shape[1],2)-0.2,temp[0,:], height=0.5,color=[1,0,1],alpha=0.5,label='Female')
g = plt.yticks(np.arange(0,2*temp.shape[1],2)+0.2,activities_interested,fontsize=14)
g = plt.ylim(-1,2*temp.shape[1]+1)
g = plt.legend(loc=0,fontsize=14)
```



Getting Dummies

In [67]:
```
dummy_gender = pd.get_dummies(data_df['gender'], prefix='gender')
print dummy_gender.head()
```

```
   gender_0  gender_1
0       1.0       0.0
1       1.0       0.0
2       1.0       0.0
3       1.0       0.0
4       1.0       0.0
```

Columns to keep

```
cols_to_keep = ['dec_o', 'age' ,'attr_o', 'sinc_o', 'intel_o', 'fun_o', 'amb_o', 'shar_o']
data_b = data_df[cols_to_keep].join(dummy_gender.ix[:, 'gender':])
data_b.describe()
```

	dec_o	age	attr_o	sinc_o	intel_o	fun_o	amb_o	shar_o	gender_0	gender_1
count	8378.000000	8283.000000	8166.000000	8091.000000	8072.000000	8018.000000	7656.000000	7302.000000	8378.000000	8378.000000
mean	0.419551	26.358928	6.190411	7.175256	7.369301	6.400599	6.778409	5.474870	0.499403	0.500597
std	0.493515	3.566763	1.950305	1.740575	1.550501	1.954078	1.794080	2.156163	0.500029	0.500029
min	0.000000	18.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	24.000000	5.000000	6.000000	6.000000	5.000000	6.000000	4.000000	0.000000	0.000000
50%	0.000000	26.000000	6.000000	7.000000	7.000000	7.000000	7.000000	6.000000	0.000000	1.000000
75%	1.000000	28.000000	8.000000	8.000000	8.000000	8.000000	8.000000	7.000000	1.000000	1.000000
max	1.000000	55.000000	10.500000	10.000000	10.000000	11.000000	10.000000	10.000000	1.000000	1.000000

```python
# intercept
data_b['intercept'] = 1.0
data_b.head()
```

	dec_o	age	attr_o	sinc_o	intel_o	fun_o	amb_o	shar_o	gender_0	gender_1	intercept
0	0	21.0	6.0	8.0	8.0	8.0	8.0	6.0	1.0	0.0	1.0
1	0	21.0	7.0	8.0	10.0	7.0	7.0	5.0	1.0	0.0	1.0
2	1	21.0	10.0	10.0	10.0	10.0	10.0	10.0	1.0	0.0	1.0
3	1	21.0	7.0	8.0	9.0	8.0	9.0	8.0	1.0	0.0	1.0
4	1	21.0	8.0	7.0	9.0	6.0	9.0	7.0	1.0	0.0	1.0

Droping data points with missing data

```python
data_b.rename(columns={'dec_o': 'decision', 'attr_o': 'attractive', 'sinc_o' : 'sincere', 'intel_o' : 'intelligent' , 'fun_o' : 'fun',
                       'amb_o' : 'ambitious', 'shar_o': 'shared interests'}, inplace=True)
```

```python
data_c = data_b.dropna()
data_c.isnull().sum()
```

```
decision            0
age                 0
attractive          0
sincere             0
intelligent         0
fun                 0
ambitious           0
shared interests    0
gender_0            0
gender_1            0
intercept           0
dtype: int64
```

```python
len(data_c)
```

```
6959
```

```python
data_c.isnull().count()
```

```
decision             6959
age                  6959
attractive           6959
sincere              6959
intelligent          6959
fun                  6959
ambitious            6959
shared interests     6959
gender_0             6959
gender_1             6959
intercept            6959
dtype: int64
```

General Dataset descriptive measure

```
data_c.describe()
```

	decision	age	attractive	sincere	intelligent	fun	ambitious	shared interests	gender_0	gender_1	inte
count	6959.000000	6959.000000	6959.000000	6959.000000	6959.000000	6959.000000	6959.000000	6959.000000	6959.000000	6959.000000	695
mean	0.429947	26.318437	6.183561	7.162954	7.361690	6.395315	6.759089	5.460052	0.506538	0.493462	1.0
std	0.495104	3.564386	1.949638	1.745162	1.559914	1.959143	1.797901	2.149901	0.499993	0.499993	0.0
min	0.000000	18.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.0
25%	0.000000	24.000000	5.000000	6.000000	6.000000	5.000000	6.000000	4.000000	0.000000	0.000000	1.0
50%	0.000000	26.000000	6.000000	7.000000	7.000000	7.000000	7.000000	6.000000	1.000000	0.000000	1.0
75%	1.000000	28.000000	8.000000	8.000000	8.000000	8.000000	8.000000	7.000000	1.000000	1.000000	1.0
max	1.000000	55.000000	10.000000	10.000000	10.000000	11.000000	10.000000	10.000000	1.000000	1.000000	1.0

```
data_c = data_c.replace(11, 10)
data_c.describe()
```

	decision	age	attractive	sincere	intelligent	fun	ambitious	shared interests	gender_0	gender_1	inte
count	6959.000000	6959.000000	6959.000000	6959.000000	6959.000000	6959.000000	6959.000000	6959.000000	6959.000000	6959.000000	695
mean	0.429947	26.318437	6.183561	7.162954	7.361690	6.395172	6.759089	5.460052	0.506538	0.493462	1.0
std	0.495104	3.564386	1.949638	1.745162	1.559914	1.958842	1.797901	2.149901	0.499993	0.499993	0.0
min	0.000000	18.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.0
25%	0.000000	24.000000	5.000000	6.000000	6.000000	5.000000	6.000000	4.000000	0.000000	0.000000	1.0
50%	0.000000	26.000000	6.000000	7.000000	7.000000	7.000000	7.000000	6.000000	1.000000	0.000000	1.0
75%	1.000000	28.000000	8.000000	8.000000	8.000000	8.000000	8.000000	7.000000	1.000000	1.000000	1.0
max	1.000000	55.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	1.000000	1.000000	1.0

CORRELATION

DATASET

```
data_c.corr()
```

	decision	age	attractive	sincere	intelligent	fun	ambitious	shared interests	gender_0	gender_1	intercept
decision	1.000000	-0.046645	0.487717	0.207160	0.214155	0.411843	0.184109	0.400070	0.117529	-0.117529	NaN
age	-0.046645	1.000000	-0.047709	0.004041	0.033008	-0.035223	0.019968	0.005233	-0.071570	0.071570	NaN
attractive	0.487717	-0.047709	1.000000	0.406055	0.388974	0.590472	0.359268	0.490608	0.129130	-0.129130	NaN
sincere	0.207160	0.004041	0.406055	1.000000	0.667933	0.507764	0.464358	0.398944	0.041191	-0.041191	NaN
intelligent	0.214155	0.033008	0.388974	0.667933	1.000000	0.500992	0.629279	0.401784	-0.057852	0.057852	NaN
fun	0.411843	-0.035223	0.590472	0.507764	0.500992	1.000000	0.493640	0.617335	0.058479	-0.058479	NaN
ambitious	0.184109	0.019968	0.359268	0.464358	0.629279	0.493640	1.000000	0.434890	-0.098770	0.098770	NaN
shared interests	0.400070	0.005233	0.490608	0.398944	0.401784	0.617335	0.434890	1.000000	0.029991	-0.029991	NaN
gender_0	0.117529	-0.071570	0.129130	0.041191	-0.057852	0.058479	-0.098770	0.029991	1.000000	-1.000000	NaN
gender_1	-0.117529	0.071570	-0.129130	-0.041191	0.057852	-0.058479	0.098770	-0.029991	-1.000000	1.000000	NaN
intercept	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

MULTICOLLINEARITY ??

Dataset (Female only)

```
subF = data_c[(data_c['gender_0']== 1)]
del subF['gender_1']
subF.head()
```

	decision	age	attractive	sincere	intelligent	fun	ambitious	shared interests	gender_0	intercept
0	0	21.0	6.0	8.0	8.0	8.0	8.0	6.0	1.0	1.0
1	0	21.0	7.0	8.0	10.0	7.0	7.0	5.0	1.0	1.0
2	1	21.0	10.0	10.0	10.0	10.0	10.0	10.0	1.0	1.0
3	1	21.0	7.0	8.0	9.0	8.0	9.0	8.0	1.0	1.0
4	1	21.0	8.0	7.0	9.0	6.0	9.0	7.0	1.0	1.0

Analysis

```
data_fem.corr()
```

	decision	age	attractive	sincere	intelligent	fun	ambitious	shared interests	intercept
decision	1.000000	-0.041354	0.519460	0.185635	0.209391	0.402063	0.211431	0.386069	NaN
age	-0.041354	1.000000	-0.057430	0.025408	0.041816	-0.046985	0.004968	0.011518	NaN
attractive	0.519460	-0.057430	1.000000	0.395661	0.413555	0.567985	0.416321	0.462070	NaN
sincere	0.185635	0.025408	0.395661	1.000000	0.666376	0.506795	0.447917	0.384916	NaN
intelligent	0.209391	0.041816	0.413555	0.666376	1.000000	0.531545	0.603450	0.408500	NaN
fun	0.402063	-0.046985	0.567985	0.506795	0.531545	1.000000	0.533609	0.589523	NaN
ambitious	0.211431	0.004968	0.416321	0.447917	0.603450	0.533609	1.000000	0.474901	NaN
shared interests	0.386069	0.011518	0.462070	0.384916	0.408500	0.589523	0.474901	1.000000	NaN
intercept	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```python
# covariates
train_cols = data_fem.columns[1:]
print train_cols
```

```
Index([u'age', u'attractive', u'sincere', u'intelligent', u'fun', u'ambitious',
       u'shared interests', u'intercept'],
      dtype='object')
```

```python
# Fit the model
```

```python
logit_fem = sm.Logit(data_fem['decision'], data_fem[train_cols])
result_F = logit_fem.fit()
```

```
Optimization terminated successfully.
         Current function value: 0.501235
         Iterations 6
```

```python
print result_F.summary()
```

```
                           Logit Regression Results
==============================================================================
Dep. Variable:               decision   No. Observations:                 3525
Model:                          Logit   Df Residuals:                     3517
Method:                           MLE   Df Model:                            7
Date:                Mon, 08 Aug 2016   Pseudo R-squ.:                  0.2765
Time:                        11:18:26   Log-Likelihood:                -1766.9
converged:                       True   LL-Null:                       -2442.2
                                        LLR p-value:                1.764e-287
====================================================================================
                       coef    std err          z      P>|z|      [95.0% Conf. Int.]
------------------------------------------------------------------------------------
age                 -0.0047      0.011     -0.412      0.681        -0.027     0.018
attractive           0.6912      0.033     20.675      0.000         0.626     0.757
sincere             -0.1585      0.037     -4.274      0.000        -0.231    -0.086
intelligent         -0.0356      0.043     -0.826      0.409        -0.120     0.049
fun                  0.2593      0.035      7.502      0.000         0.192     0.327
ambitious           -0.1574      0.034     -4.604      0.000        -0.224    -0.090
shared interests     0.2671      0.026     10.141      0.000         0.215     0.319
intercept           -5.1709      0.398    -12.981      0.000        -5.952    -4.390
====================================================================================
```

MULTICOLLINEARITY!!!

Dataset (Male only)

```
subM = data_c[(data_c['gender_1']== 1)]
del subM['gender_0']
subM.tail()
```

	decision	age	attractive	sincere	intelligent	fun	ambitious	shared interests	gender_1	intercept
8373	1	25.0	10.0	5.0	3.0	2.0	6.0	5.0	1.0	1.0
8374	0	25.0	6.0	3.0	7.0	3.0	7.0	2.0	1.0	1.0
8375	0	25.0	2.0	1.0	2.0	2.0	2.0	1.0	1.0	1.0
8376	1	25.0	5.0	7.0	5.0	5.0	3.0	6.0	1.0	1.0
8377	1	25.0	8.0	8.0	7.0	7.0	7.0	7.0	1.0	1.0

```
data_male.corr()
```

	decision	age	attractive	sincere	intelligent	fun	ambitious	shared interests	intercept
decision	1.000000	-0.035481	0.442750	0.222257	0.236811	0.416736	0.185408	0.413774	NaN
age	-0.035481	1.000000	-0.020365	-0.010423	0.016054	-0.015896	0.021258	0.003138	NaN
attractive	0.442750	-0.020365	1.000000	0.412003	0.388979	0.606590	0.342481	0.517172	NaN
sincere	0.222257	-0.010423	0.412003	1.000000	0.677393	0.506644	0.490845	0.410211	NaN
intelligent	0.236811	0.016054	0.388979	0.677393	1.000000	0.484569	0.649149	0.400695	NaN
fun	0.416736	-0.015896	0.606590	0.506644	0.484569	1.000000	0.476577	0.641272	NaN
ambitious	0.185408	0.021258	0.342481	0.490845	0.649149	0.476577	1.000000	0.408830	NaN
shared interests	0.413774	0.003138	0.517172	0.410211	0.400695	0.641272	0.408830	1.000000	NaN
intercept	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
# covariates
train_cols = data_male.columns[1:]
print train_cols
```

```
Index([u'age', u'attractive', u'sincere', u'intelligent', u'fun', u'ambitious',
       u'shared interests', u'intercept'],
      dtype='object')
```

```
# Fit the model
```

```
logit_male = sm.Logit(data_male['decision'], data_male[train_cols])
result_M = logit_male.fit()
```

```
Optimization terminated successfully.
         Current function value: 0.501609
         Iterations 6
```

```
print result_M.summary()
```

```
                           Logit Regression Results
==============================================================================
Dep. Variable:               decision   No. Observations:                 3434
Model:                          Logit   Df Residuals:                     3426
Method:                           MLE   Df Model:                            7
Date:                Mon, 08 Aug 2016   Pseudo R-squ.:                  0.2394
Time:                        17:15:37   Log-Likelihood:                -1722.5
converged:                       True   LL-Null:                       -2264.7
                                        LLR p-value:                7.367e-230
====================================================================================
                       coef    std err          z      P>|z|      [95.0% Conf. Int.]
------------------------------------------------------------------------------------
age                 -0.0223      0.013     -1.784      0.074        -0.047     0.002
attractive           0.4045      0.029     13.748      0.000         0.347     0.462
sincere             -0.0943      0.035     -2.686      0.007        -0.163    -0.025
intelligent          0.1300      0.045      2.894      0.004         0.042     0.218
fun                  0.2697      0.034      8.026      0.000         0.204     0.336
ambitious           -0.1654      0.034     -4.795      0.000        -0.233    -0.098
shared interests     0.2794      0.027     10.415      0.000         0.227     0.332
intercept           -4.9712      0.425    -11.699      0.000        -5.804    -4.138
====================================================================================
```

MULTICOLLINEARITY!!!

NEXT STEP

- **Determinate new** “Categories of features”
- **Create a new** variables with the mean of the variables in that group
- or just keep one of the variables.