In [13]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [14]:

```
dating_data = pd.read_csv('speed_dating.csv')
```

In [15]:

```
dating_data.head()
```

Out[15]:

	has_null	wave	gender	age	age_o	d_age	d_d_age	race	race_o	s
0	0	1	female	21.0	27.0	6	[4-6]	asian/pacific islander/asian-american	european/caucasian-american	
1	0	1	female	21.0	22.0	1	[0-1]	asian/pacific islander/asian-american	european/caucasian-american	
2	1	1	female	21.0	22.0	1	[0-1]	asian/pacific islander/asian-american	asian/pacific islander/asian-american	
3	0	1	female	21.0	23.0	2	[2-3]	asian/pacific islander/asian-american	european/caucasian-american	
4	0	1	female	21.0	24.0	3	[2-3]	asian/pacific islander/asian-american	latino/hispanic american	

5 rows × 123 columns

In [16]:

```
dating_data.tail()
```

Out[16]:

	has_null	wave	gender	age	age_o	d_age	d_d_age	race	race_o
8373	1	21	male	25.0	26.0	1	[0-1]	european/caucasian-american	latino/hispanic american
8374	1	21	male	25.0	24.0	1	[0-1]	european/caucasian-american	other
8375	1	21	male	25.0	29.0	4	[4-6]	european/caucasian-american	latino/hispanic american
8376	1	21	male	25.0	22.0	3	[2-3]	european/caucasian-american	asian/pacific islander/asian-american
8377	1	21	male	25.0	22.0	3	[2-3]	european/caucasian-american	asian/pacific islander/asian-american

5 rows × 123 columns

In [17]:

```
dating_data.shape
```

Out[17]:

```
(8378, 123)
```

In [18]:

```
dating_data.columns
```

Out[18]:

```
Index(['has_null', 'wave', 'gender', 'age', 'age_o', 'd_age', 'd_d_age',
       'race', 'race_o', 'samerace',
       ...
       'd_expected_num_interested_in_me', 'd_expected_num_matches', 'lik
e',
       'guess_prob_liked', 'd_like', 'd_guess_prob_liked', 'met', 'decisio
n',
       'decision_o', 'match'],
      dtype='object', length=123)
```

In [19]:

```
dating_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8378 entries, 0 to 8377
Columns: 123 entries, has_null to match
dtypes: float64(57), int64(7), object(59)
memory usage: 7.9+ MB
```

In [20]:

```
dating_data.describe()
```

Out[20]:

	has_null	wave	age	age_o	d_age	samerace	importar
count	8378.00000	8378.000000	8283.000000	8274.000000	8378.000000	8378.000000	
mean	0.87491	11.350919	26.358928	26.364999	4.185605	0.395799	
std	0.33084	5.995903	3.566763	3.563648	4.596171	0.489051	
min	0.00000	1.000000	18.000000	18.000000	0.000000	0.000000	
25%	1.00000	7.000000	24.000000	24.000000	1.000000	0.000000	
50%	1.00000	11.000000	26.000000	26.000000	3.000000	0.000000	
75%	1.00000	15.000000	28.000000	28.000000	5.000000	1.000000	
max	1.00000	21.000000	55.000000	55.000000	37.000000	1.000000	

8 rows × 64 columns

In [22]:

```
with open('speed_dating.txt') as f:
    contents = f.read()
    print(contents)
```

```
* gender: Gender of self
* age: Age of self
* age_o: Age of partner
* d_age: Difference in age
* race: Race of self
* race_o: Race of partner
* samerace: Whether the two persons have the same race or not.
* importance_same_race: How important is it that partner is of same race?
* importance_same_religion: How important is it that partner has same reli
gion?
* field: Field of study
* pref_o_attractive: How important does partner rate attractiveness
* pref_o_sinsere: How important does partner rate sincerity
* pref_o_intelligence: How important does partner rate intelligence
* pref_o_funny: How important does partner rate being funny
* pref_o_ambitious: How important does partner rate ambition
* pref_o_shared_interests: How important does partner rate having shared i
nterests
* attractive_o: Rating by partner (about me) at night of event on attracti
veness
* sincere_o: Rating by partner (about me) at night of event on sincerity
* intelligence_o: Rating by partner (about me) at night of event on intell
igence
* funny_o: Rating by partner (about me) at night of event on being funny
* ambitous_o: Rating by partner (about me) at night of event on being ambi
tious
* shared_interests_o: Rating by partner (about me) at night of event on sh
ared interest
* attractive_important: What do you look for in a partner - attractiveness
* sincere_important: What do you look for in a partner - sincerity
* intellicence_important: What do you look for in a partner - intelligence
* funny_important: What do you look for in a partner - being funny
* ambtition_important: What do you look for in a partner - ambition
* shared_interests_important: What do you look for in a partner - shared i
nterests
* attractive: Rate yourself - attractiveness
* sincere: Rate yourself - sincerity
* intelligence: Rate yourself - intelligence
* funny: Rate yourself - being funny
* ambition: Rate yourself - ambition
* attractive_partner: Rate your partner - attractiveness
* sincere_partner: Rate your partner - sincerity
* intelligence_partner: Rate your partner - intelligence
* funny_partner: Rate your partner - being funny
* ambition_partner: Rate your partner - ambition
* shared_interests_partner: Rate your partner - shared interests
* sports: Your own interests [1-10]
* tvsports
* exercise
* dining
* museums
* art
* hiking
* gaming
```

```
* clubbing
* reading
* tv
* theater
* movies
* concerts
* music
* shopping
* yoga
* interests_correlate: Correlation between participantâ€™s and partnerâ€™s
ratings of interests.
* expected_happy_with_sd_people: How happy do you expect to be with the pe
ople you meet during the speed-dating event?
* expected_num_interested_in_me: Out of the 20 people you will meet, how m
any do you expect will be interested in dating you?
* expected_num_matches: How many matches do you expect to get?
* like: Did you like your partner?
* guess_prob_liked: How likely do you think it is that your partner likes
you?
* met: Have you met your partner before?
* decision: Decision at night of event.
* decision_o: Decision of partner at night of event.
* match: Match (yes/no)
```

In [21]:

```
dating_data.isnull().sum()
```

Out[21]:

```
has_null             0
wave                 0
gender               0
age                 95
age_o              104
                  ...
d_guess_prob_liked   0
met                375
decision             0
decision_o           0
match                0
Length: 123, dtype: int64
```

In [23]:

```
dating_data.nunique()
```

Out[23]:

```
has_null                 2
wave                    21
gender                   2
age                     24
age_o                   24
                        ..
d_guess_prob_liked       3
met                      7
decision                 2
decision_o               2
match                    2
Length: 123, dtype: int64
```

In [24]:

```
dating_categorical = ['gender', 'race', 'race_o', 'field']
dating_numerical = ['has_null', 'wave', 'age', 'age_o', 'd_age', 'samerace', 'importance
 'importance_same_religion', 'pref_o_attractive', 'pref_o_sincere', 'pref_o_intelligence
 'pref_o_ambitious', 'pref_o_shared_interests', 'attractive_o', 'sinsere_o', 'intelliger
 'ambitous_o', 'shared_interests_o', 'attractive_important', 'sincere_important', 'intel
 'funny_important', 'ambtition_important', 'shared_interests_important', 'attractive', '
 'funny', 'ambition', 'attractive_partner', 'sincere_partner', 'intelligence_partner', '
 'shared_interests_partner', 'sports', 'tvsports', 'exercise', 'dining', 'museums', 'art
 'reading', 'tv', 'theater', 'movies', 'concerts', 'music', 'shopping', 'yoga', 'interes
 'expected_happy_with_sd_people', 'expected_num_interested_in_me', 'expected_num_matches
```

In [26]:

```
dating_data[dating_categorical].nunique()
```

Out[26]:

```
gender      2
race        5
race_o      5
field     219
dtype: int64
```

In [28]:

```
dating_data[dating_categorical].isnull().sum()
```

Out[28]:

```
gender     0
race      63
race_o    73
field     63
dtype: int64
```

In [29]:

```
dating_data[dating_numerical].nunique()
```

Out[29]:

```
has_null                         2
wave                            21
age                             24
age_o                           24
d_age                           35
                                ..
expected_num_interested_in_me   18
expected_num_matches            17
like                            18
guess_prob_liked                19
met                              7
Length: 61, dtype: int64
```

In [30]:

```
dating_data[dating_numerical].isnull().sum()
```

Out[30]:

```
has_null                           0
wave                               0
age                               95
age_o                            104
d_age                              0
                                ...
expected_num_interested_in_me   6578
expected_num_matches            1173
like                             240
guess_prob_liked                 309
met                              375
Length: 61, dtype: int64
```

In [31]:

```
dating_data['field'].unique()
```

Out[31]:

```
array(['law', 'economics', 'masters in public administration',
       'masters of social work&education', 'finance', 'business',
       'political science', 'money', 'operations research',
       'tc [health ed]', 'psychology', 'social work',
       'speech language pathology', 'speech languahe pathology',
       'educational psychology', 'applied maths/econs', 'mathematics',
       'statistics', 'organizational psychology',
       'mechanical engineering', 'finanace', 'finance&economics',
       'undergrad - gs', 'mathematical finance', 'medicine', 'mba', nan,
       'german literature', 'business & international affairs',
       'mfa creative writing', 'engineering', 'electrical engineering',
       'classics', 'operations research [seas]', 'chemistry',
       'journalism', 'elementary/childhood education [ma]',
       'microbiology', 'masters of social work', 'communications',
       'marketing', 'international educational development',
       'education administration', 'business [mba]', 'computer science',
       'climate-earth and environ. science', 'financial math',
       'business- mba', 'religion', 'film', 'sociology',
       'economics; english', 'economics; sociology', 'polish', 'english',
       'psychology and english', 'biomedical engineering',
       'economics and political science', 'art history/medicine',
       'philosophy', 'marine geophysics', 'theory', 'nutrition/genetics',
       'neuroscience', 'comparative literature',
       'international relations', 'history of religion',
       'international affairs - economic development',
       'modern chinese literature', 'business; marketing',
       'physics [astrophysics]', 'physics',
       'business/ finance/ real estate', 'biochemistry', 'art education',
       'american studies [masters]', 'biology', 'cell biology', 'math',
       'international affairs/finance', 'international affairs',
       'international affairs/international finance', 'health policy',
       'english and comp lit', 'international finance and business',
       'sociomedical sciences- school of public health', 'epidemiology',
       'international business', 'medical informatics',
       'international finance; economic policy', 'law and social work',
       'international development', 'business/law', 'clinical psychology',
       'religion; gsas', 'international affairs and public health',
       'history',
       'business and international affairs [mba/mia dual degree]', 'qmss',
       'climate change', 'public administration', 'ma biotechnology',
       'international affairs/business', 'ecology',
       'master in public administration', 'computational biochemsistry',
       'neurobiology', 'mathematics; phd', 'history [gsas - phd]',
       'biomedicine', 'master of international affairs',
       'sociology and education', 'elementary education',
       'american studies', 'arts administration', 'conservation biology',
       'japanese literature', 'biotechnology',
       'earth and environmental science', 'philosophy [ph.d.]',
       'philosophy and physics', 'nutrition', 'ma science education',
       'genetics', 'law and english literature [j.d./ph.d.]', 'french',
       'nutritiron', 'gs postbacc premed', 'art history',
       'molecular biology', 'genetics & development', 'electrical engg.',
       'business school', 'international politics',
       'mba / master of international affairs [sipa]',
```

```
       'medicine and biochemistry', 'social studies education',
       'ma teaching social studies', 'education policy',
       'education- literacy specialist', 'anthropology/education',
       'bilingual education', 'speech pathology', 'education',
       'math education', 'tesol', 'cognitive studies in education',
       'finance/economics', 'museum anthropology',
       'environmental engineering', 'business administration',
       'curriculum and teaching/giftedness', 'anthropology',
       'instructional tech & media', 'school psychology',
       'instructional media and technology', 'sipa / mia',
       'english education', 'ma in quantitative methods',
       'early childhood education', 'architecture', 'urban planning',
       'ed.d. in higher education policy at tc',
       'international security policy - sipa',
       'applied physiology & nutrition', 'music education',
       'counseling psychology', 'communications in education',
       'intellectual property law', 'mba finance',
       'intrernational affairs', 'business consulting', 'business; media',
       'mfa -film', 'higher ed. - m.a.', 'neuroscience and education',
       'creative writing', 'creative writing - nonfiction',
       'writing: literary nonfiction', 'creative writing [nonfiction]',
       'nonfiction writing', 'theatre management & producing',
       'financial engineering', 'fundraising management',
       'business [finance & marketing]',
       'elementary education - preservice',
       'education leadership - public school administration',
       'mfa writing', 'international affairs - economic policy',
       'sipa - energy', 'public policy', 'law/business', 'mfa  poetry',
       'soa -- writing', 'biomedical informatics', 'working',
       'consulting', 'human rights: middle east', 'human rights',
       'sipa-international affairs', 'teaching of english', 'gsas',
       'african-american studies/history', 'neurosciences/stem cells',
       'theater', 'biology phd', 'biochemistry/genetics', 'stats',
       'math of finance', 'mfa acting program',
       'biochemistry & molecular biophysics', 'acting',
       'social work/sipa', 'public health', 'industrial engineering',
       'industrial engineering/operations research',
       'masters of industrial engineering',
       'mba - private equity / real estate', 'general management/finance',
       'climate dynamics'], dtype=object)
```

In [32]:

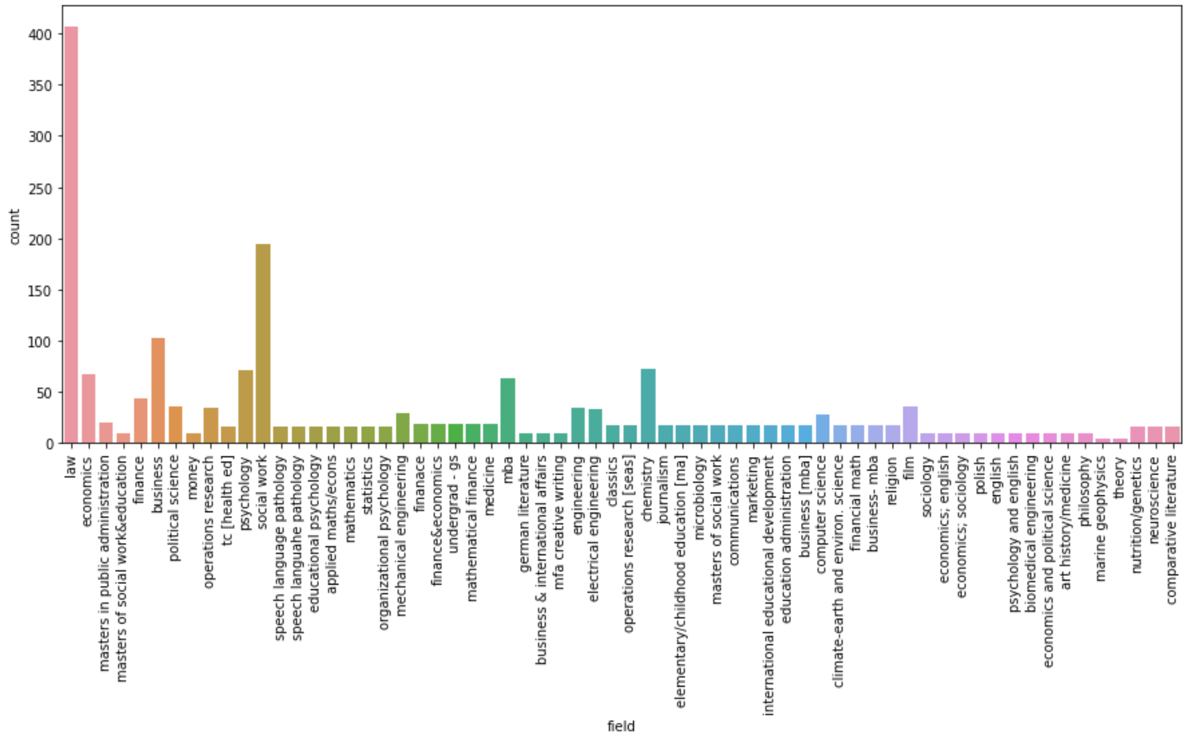
```
dating_data['field'].value_counts()
```

Out[32]:

```
business                          631
law                               604
mba                               468
social work                       414
international affairs             287
                                 ... 
mfa  poetry                         6
fundraising management              6
business [finance & marketing]      6
marine geophysics                   5
theory                              5
Name: field, Length: 219, dtype: int64
```

In [36]:

```
plt.figure(figsize=(15,6))
sns.countplot('field', data = dating_data.head(2000))
plt.xticks(rotation = 90)
plt.show()
```



In [37]:

```
import string
import re
```

In [38]:

```
dating_data['race'] = dating_data['race'].str.lower()
dating_data['race'] = dating_data['race'].str.replace("'", "", regex=False)
dating_data['race'] = dating_data['race'].str.replace(" ", "_", regex=False)
dating_data['race_o'] = dating_data['race_o'].str.lower()
dating_data['race_o'] = dating_data['race_o'].str.replace("'", "", regex=False)
dating_data['race_o'] = dating_data['race_o'].str.replace(" ", "_", regex=False)
```

In [41]:

```
dating_data.race = dating_data.race.fillna('Not Available')
dating_data.race_o = dating_data.race_o.fillna('Not Available')
dating_data.field = dating_data.field.fillna('Not Available')
```

In [42]:

```
dating_data[dating_categorical].isnull().sum()
```

Out[42]:

```
gender    0
race      0
race_o    0
field     0
dtype: int64
```

In [43]:

```
dating_data.drop(columns=['expected_num_interested_in_me'],inplace=True)
```

In [44]:

```
dating_numerical.remove('expected_num_interested_in_me')
```

In [45]:

```
for i in dating_numerical:
    dating_data[i] = dating_data[i].fillna(dating_data[i].mean())
```

In [46]:

```
dating_data[dating_numerical].isnull().sum()
```

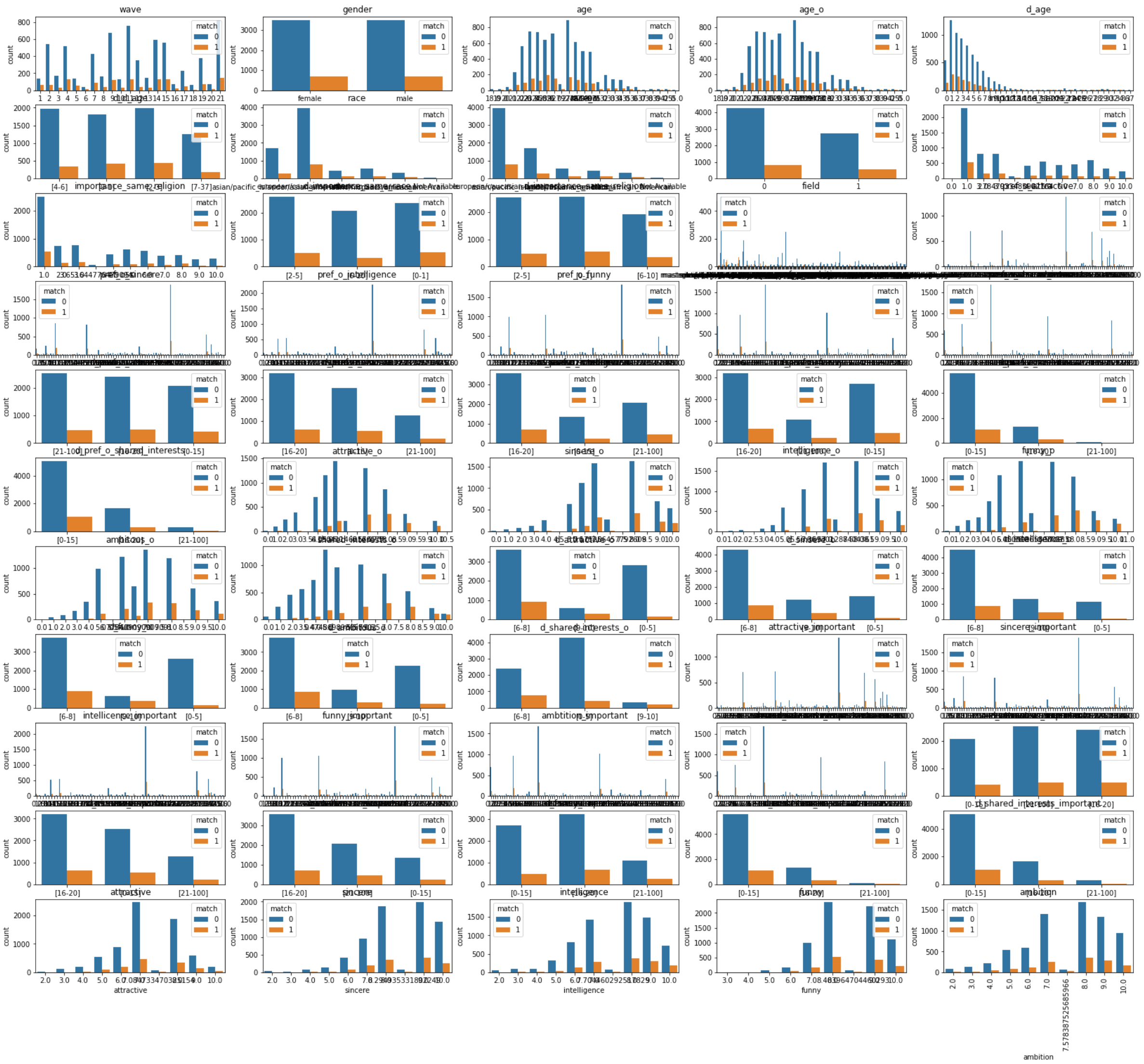
Out[46]:

```
has_null                         0
wave                             0
age                              0
age_o                            0
d_age                            0
samerace                         0
importance_same_race             0
importance_same_religion         0
pref_o_attractive                0
pref_o_sincere                   0
pref_o_intelligence              0
pref_o_funny                     0
pref_o_ambitious                 0
pref_o_shared_interests          0
attractive_o                     0
sinsere_o                        0
intelligence_o                   0
funny_o                          0
ambitous_o                       0
shared_interests_o               0
attractive_important             0
sincere_important                0
intellicence_important           0
funny_important                  0
ambtition_important              0
shared_interests_important       0
attractive                       0
sincere                          0
intelligence                     0
funny                            0
ambition                         0
attractive_partner               0
sincere_partner                  0
intelligence_partner             0
funny_partner                    0
ambition_partner                 0
shared_interests_partner         0
sports                           0
tvsports                         0
exercise                         0
dining                           0
museums                          0
art                              0
hiking                           0
gaming                           0
clubbing                         0
reading                          0
tv                               0
theater                          0
movies                           0
concerts                         0
music                            0
shopping                         0
yoga                             0
```

```
interests_correlate          0
expected_happy_with_sd_people    0
expected_num_matches         0
like                         0
guess_prob_liked             0
met                          0
dtype: int64
```

In [47]:

```python
fig, axes = plt.subplots(11,5,figsize=(28,25))
s=0
for i in range(0,11):
    for j in range(0,5):
        s+=1
        if s==123:
            break
        sns.countplot(ax = axes[i,j],x=dating_data.columns[s],
                      data=dating_data,
                      hue='match')
        plt.xticks(rotation = 90)
        axes[i,j].set_title(dating_data.columns[s])
```



In [48]:

```
dating_data.match.value_counts()
```

Out[48]:

```
0    6998
1    1380
Name: match, dtype: int64
```

In [49]:

```
match = dating_data[dating_data['match']==1]
not_match = dating_data[dating_data['match']==0]
```

In [50]:

```
match.groupby('gender')['match'].count()
```

Out[50]:

```
gender
female    690
male      690
Name: match, dtype: int64
```

In [51]:

```
not_match.groupby('gender')['match'].count()
```

Out[51]:

```
gender
female    3494
male      3504
Name: match, dtype: int64
```

In [52]:

```
dating_data.corr()
```

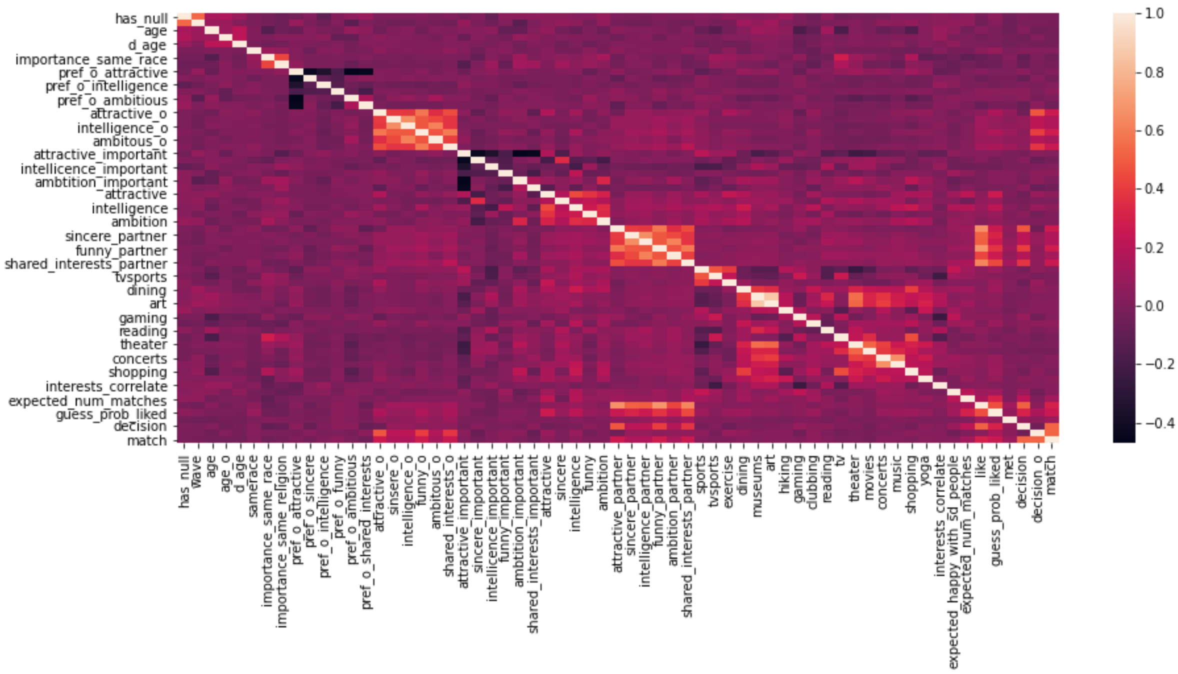
Out[52]:

	has_null	wave	age	age_o	d_age	samerace	importance_
has_null	1.000000	0.529313	0.144285	0.165107	0.094874	-0.016382	
wave	0.529313	1.000000	0.094523	0.092863	0.022024	-0.014967	
age	0.144285	0.094523	1.000000	0.099012	0.202476	0.007107	
age_o	0.165107	0.092863	0.099012	1.000000	0.208846	0.005737	
d_age	0.094874	0.022024	0.202476	0.208846	1.000000	-0.006238	
...	...	...	...	...	...	...	...
guess_prob_liked	0.041519	0.021093	-0.012547	-0.009376	-0.019391	0.082328	
met	-0.035000	-0.054883	-0.059553	-0.028931	-0.036715	-0.002383	
decision	-0.002146	-0.011598	0.015801	-0.049065	-0.026940	0.023036	
decision_o	-0.009000	-0.010831	-0.047566	0.015043	-0.028545	0.023626	
match	-0.013011	-0.017404	-0.034832	-0.035632	-0.038239	0.013028	

63 rows × 63 columns

In [54]:

```
plt.figure(figsize=(15,6))
sns.heatmap(dating_data.corr())
plt.show()
```



In [61]:

```
x = dating_data[dating_numerical]
y = dating_data['match']
```

In [71]:

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y,
                                                    test_size=0.15,
                                                    random_state=42)
```

In [72]:

```
from sklearn.tree import DecisionTreeClassifier
classifier= DecisionTreeClassifier(criterion='entropy', random_state=0)
classifier.fit(x_train, y_train)
```

Out[72]:

```
DecisionTreeClassifier(criterion='entropy', random_state=0)
```

In [73]:

```
y_pred= classifier.predict(x_test)
```

In [65]:

```
from sklearn.metrics import confusion_matrix
cm= confusion_matrix(y_test, y_pred)
```

In [78]:

```
print('Confusion matrix : \n',cm)
```

```
Confusion matrix :
 [[1207  182]
 [ 173  114]]
```

In [76]:

```
from sklearn import metrics
from sklearn.metrics import accuracy_score
```

In [77]:

```
ion report for classifier %s:\n%s\n" % (classifier,
                                         metrics.classification_report(y_test, y_pred)))
```

```
 Classification report for classifier DecisionTreeClassifier(criterion='en
tropy', random_state=0):
              precision    recall  f1-score   support

           0       0.86      0.87      0.86      1033
           1       0.36      0.34      0.35       224

    accuracy                           0.78      1257
   macro avg       0.61      0.61      0.61      1257
weighted avg       0.77      0.78      0.77      1257
```