

Speed Dating Experiment

An Inncretech Publication



In this experiment, we try to predict the outcome of a speed dating scenario using variables such as shared interests, perception, gender and a few more. Check out the steps we took and find the results below.

```
In [1]: %%HTML
<h1>Speed Dating Problem</h1>
```

Speed Dating Problem

```
In [1]: import os
import time
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from __future__ import division
```

```
In [135]: df = pd.read_csv("data/Speed Dating Data.csv")

#Creating dummy columns -> "Shared Interests", question type 2 & 5, times T1,
T2, T3

cols = ['shar3_1', 'shar3_2', 'shar3_3', 'shar5_1', 'shar5_2', 'shar5_3']
df = pd.concat([df, pd.DataFrame(columns=cols)])
```

```
In [3]: #Filling NA values in the 'id' column
#Every subject (denoted by their 'iid') has a unique 'id' value, which denotes
their number within their wave.

df['id'] = df[['iid', 'id']].groupby(['iid'])['id'].transform(lambda grp: grp.
fillna(method='ffill'))
```

```
In [227]: columns = ['iid', 'age', 'gender', 'field', 'field_cd',
'undergra', 'mn_sat', 'tuition', 'race', 'from',
'zipcode', 'income', 'career', 'career_c']

df_bio1 = df[columns].drop_duplicates()
```

```
In [303]: columns = ['iid', 'imprace', 'imprelig', 'goal', 'date',
'go_out', 'sports', 'tvsports', 'exercise', 'dining',
'museums', 'art', 'hiking', 'gaming', 'clubbing',
'reading', 'tv', 'theater', 'movies', 'concerts',
'music', 'shopping', 'yoga', 'exphappy', 'expnum',
'you_call', 'them_cal', 'date_3', 'numdat_3', 'num_in_3',
'satis_2', 'length', 'numdat_2']

df_bio2 = df[columns].drop_duplicates()
```

```
In [229]: # 1 - What do you look for in the opposite sex?
# 2 - What do you think the opposite sex looks for in a date?
# 3 - How do you think you measure up?
# 4 - What do you think MOST of your fellow men/women look for in the opposite
sex?
# 5 - How do you think others perceive you?

columns = ['iid',
           'attr1_1', 'sinc1_1', 'intel1_1', 'fun1_1', 'amb1_1', 'shar1_1',
           'attr2_1', 'sinc2_1', 'intel2_1', 'fun2_1', 'amb2_1', 'shar2_1',
           'attr3_1', 'sinc3_1', 'intel3_1', 'fun3_1', 'amb3_1',
           'attr4_1', 'sinc4_1', 'intel4_1', 'fun4_1', 'amb4_1', 'shar4_1',
           'attr5_1', 'sinc5_1', 'intel5_1', 'fun5_1', 'amb5_1']

df_exp1 = df[columns].drop_duplicates()
```

```
In [313]: columns = ['iid', 'order', 'pid', 'partner', 'match', 'dec_o', 'dec', 'int_cor
r',
                    'samerace', 'age_o', 'race_o',
                    'attr_o', 'sinc_o', 'intel_o', 'fun_o',
                    'amb_o', 'shar_o', 'like_o', 'prob_o', 'met_o',
                    'like', 'prob', 'attr', 'sinc',
                    'intel', 'fun', 'amb', 'shar']

df_dates = df[columns].drop_duplicates().sort_values(by=['iid', 'order', 'pid',
'partner'])
```

```
In [334]: cols = ['amb', 'attr', 'fun', 'intel', 'shar', 'sinc']
df2.loc[(df2.iid==1) & (df2.type==1)][cols].mean()
```

```
Out[334]: amb      13.703333
attr      16.480000
fun       17.406667
intel     17.963333
shar      15.556667
sinc      18.890000
dtype: float64
```

```
In [333]: df_dates.loc[df_dates.iid==1][['iid', 'pid', 'dec', 'dec_o', 'match', 'like']]
df_dates.loc[(df_dates.iid==1)&(df_dates.dec==1)][['pid','like','attr','sinc',
'intel','fun','amb','shar']]
```

Out[333]:

	pid	like	attr	sinc	intel	fun	amb	shar
6	17.0	6.0	7.0	6.0	7.0	4.0	6.0	7.0
1	12.0	7.0	7.0	8.0	7.0	8.0	5.0	6.0
0	11.0	7.0	6.0	9.0	7.0	7.0	6.0	5.0
3	14.0	7.0	7.0	6.0	8.0	7.0	6.0	8.0
4	15.0	6.0	5.0	6.0	7.0	7.0	6.0	6.0
8	19.0	7.0	7.0	6.0	8.0	9.0	8.0	8.0
9	20.0	6.0	5.0	6.0	6.0	8.0	10.0	8.0
2	13.0	7.0	5.0	8.0	9.0	8.0	5.0	7.0

In [240]:

```
times = range(1, 4)
types = range(1, 6)
rows = list()

for iid in df['iid'].drop_duplicates():
    record = df.loc[df.iid==iid].head(1)
    for _time in times:
        t1 = str(_time)
        for _type in types:
            t2 = str(_type)
            rows.append({
                'iid':iid,
                'gender':record['gender'].values[0],
                'race': record['race'].values[0],
                'wave': record['wave'].values[0],
                'time':_time,
                'type':_type,
                'attr':record['attr'+t2+"_"+t1].values[0],
                'sinc':record['sinc'+t2+"_"+t1].values[0],
                'intel':record['intel'+t2+"_"+t1].values[0],
                'fun':record['fun'+t2+"_"+t1].values[0],
                'amb':record['amb'+t2+"_"+t1].values[0],
                'shar':record['shar'+t2+"_"+t1].values[0]})

df2 = pd.DataFrame(rows)
```

```
In [312]: # 1 - What do you look for in the opposite sex?
# 2 - What do you think the opposite sex looks for in a date?
# 3 - How do you think you measure up?
# 4 - What do you think MOST of your fellow men/women look for in the opposite
sex?
# 5 - How do you think others perceive you?
```

```
cols = ['amb', 'attr', 'fun', 'intel', 'shar', 'sinc']
df2.groupby(['type', 'gender'])[cols].median()
```

Out[312]:

		amb	attr	fun	intel	shar	sinc
type	gender						
1	0.0	12.77	16.98	17.31	20.00	15.000	18.00
	1.0	10.00	25.00	18.00	20.00	10.000	16.28
2	0.0	10.00	30.00	20.00	10.00	14.645	10.00
	1.0	15.00	20.00	17.78	15.47	10.000	15.00
3	0.0	8.00	7.00	8.00	8.00	NaN	9.00
	1.0	8.00	7.00	8.00	8.00	NaN	8.00
4	0.0	10.00	20.00	15.00	10.00	10.000	10.00
	1.0	7.00	25.00	15.00	10.00	10.000	10.00
5	0.0	8.00	7.00	7.00	8.00	NaN	8.00
	1.0	7.00	7.00	7.00	8.00	NaN	8.00

Thank You

If you'd like to learn more, or get
in touch with our experts:

Email us: info@inncretech.com

Skype us: @voipgeek