In [1]:

```
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import scipy.stats as sts
import matplotlib as mpl
import seaborn as sns
from sklearn.decomposition import PCA # Principal Component Analysis module
%matplotlib inline
```

In [2]:

```
#For QQ Plot
import scipy.stats as sts

#Correlation p-values
from scipy.stats.stats import pearsonr

#Regression output
from sklearn.linear_model import LinearRegression
import statsmodels.formula.api as smf
```

In [3]:

```
path = "C:/Users/Administrator/Documents/Master/MSIS-5223-70250 - Programming
 for Data Sci - 8282017 - 159 PM/Data for Tutorials and ICE/Data"
os.chdir(path)
df = pd.read_table('reduction_data_new.txt', sep= '\t')
```

In [4]:

```
df
```

Out[4]:

	time	peruse01	peruse02	peruse03	peruse04	peruse05	peruse06	pereou01	pere
0	14	7	7	7	7	7	7	7	7
1	14	6	5	5	6	6	6	6	6
2	10	5	5	5	6	5	6	3	5
3	13	5	5	5	5	5	6	6	5
4	4	7	6	6	6	7	6	6	6
5	5	6	6	7	6	7	6	6	6
6	6	6	6	6	7	6	7	7	7
7	7	6	6	6	5	5	5	7	7
8	5	5	4	5	5	5	4	5	4
9	12	6	6	6	5	5	6	5	5
10	9	6	5	6	6	6	4	5	6
11	10	6	6	6	6	6	6	7	7
12	17	5	6	6	5	5	6	6	6
13	14	5	6	6	6	6	6	6	6
14	15	5	5	5	5	6	6	6	6
15	18	6	5	5	6	6	6	6	6
16	10	2	2	2	2	2	2	7	6
17	17	1	3	2	3	3	4	2	2
18	10	5	5	5	5	4	2	6	6
19	10	6	3	5	6	6	6	6	6
20	16	6	6	6	6	7	6	7	7
21	13	6	6	5	6	5	6	5	6
22	17	5	5	5	3	4	5	5	5
23	9	6	6	6	6	6	6	5	6
24	25	6	6	6	7	6	6	6	5
25	9	5	6	5	5	6	6	6	6
26	13	7	7	7	7	7	7	7	7
27	6	7	7	6	6	7	7	7	7
28	5	6	6	7	6	7	7	6	5
29	7	6	6	6	6	6	6	6	6
...	...	...	...	...	...	...	...	...	...
138	19	7	7	7	6	6	7	6	6

	time	peruse01	peruse02	peruse03	peruse04	peruse05	peruse06	pereou01	pere
139	8	5	5	4	4	4	5	7	5
140	7	5	5	5	5	5	5	6	6
141	16	6	6	6	6	6	6	6	6
142	15	6	6	6	6	6	6	6	6
143	10	7	7	7	7	7	7	7	7
144	9	6	6	6	6	6	6	7	7
145	16	4	4	4	4	4	5	6	6
146	6	6	6	6	6	6	6	6	6
147	14	4	5	5	5	5	5	7	7
148	10	4	5	5	5	6	6	7	7
149	10	6	4	4	5	5	5	7	6
150	10	6	6	6	6	6	7	6	6
151	10	6	6	6	5	4	5	7	7
152	11	7	6	6	7	7	7	7	7
153	27	6	6	6	6	6	6	5	5
154	13	6	6	6	6	6	4	6	5
155	49	5	4	4	4	5	4	5	4
156	12	6	5	5	6	5	5	6	6
157	17	7	6	6	7	7	6	6	6
158	21	6	6	7	6	5	6	5	6
159	56	7	6	6	5	6	5	3	6
160	13	6	6	5	6	6	6	7	6
161	12	7	6	7	6	7	7	7	6
162	16	5	5	5	5	5	5	6	5
163	6	5	5	5	5	4	4	5	5
164	4	5	5	5	5	4	5	5	5
165	14	6	7	6	7	7	7	7	7
166	23	5	5	6	6	6	6	5	5
167	24	6	4	5	4	5	5	7	6

168 rows × 38 columns

In [6]:

```
#Drop Null data
df.dropna(axis = 1, how = 'any')
```

Out[6]:

	time	peruse01	peruse02	peruse03	peruse04	peruse05	peruse06	pereou01	pere
0	14	7	7	7	7	7	7	7	7
1	14	6	5	5	6	6	6	6	6
2	10	5	5	5	6	5	6	3	5
3	13	5	5	5	5	5	6	6	5
4	4	7	6	6	6	7	6	6	6
5	5	6	6	7	6	7	6	6	6
6	6	6	6	6	7	6	7	7	7
7	7	6	6	6	5	5	5	7	7
8	5	5	4	5	5	5	4	5	4
9	12	6	6	6	5	5	6	5	5
10	9	6	5	6	6	6	4	5	6
11	10	6	6	6	6	6	6	7	7
12	17	5	6	6	5	5	6	6	6
13	14	5	6	6	6	6	6	6	6
14	15	5	5	5	5	6	6	6	6
15	18	6	5	5	6	6	6	6	6
16	10	2	2	2	2	2	2	7	6
17	17	1	3	2	3	3	4	2	2
18	10	5	5	5	5	4	2	6	6
19	10	6	3	5	6	6	6	6	6
20	16	6	6	6	6	7	6	7	7
21	13	6	6	5	6	5	6	5	6
22	17	5	5	5	3	4	5	5	5
23	9	6	6	6	6	6	6	5	6
24	25	6	6	6	7	6	6	6	5
25	9	5	6	5	5	6	6	6	6
26	13	7	7	7	7	7	7	7	7
27	6	7	7	6	6	7	7	7	7
28	5	6	6	7	6	7	7	6	5
29	7	6	6	6	6	6	6	6	6
...	...	...	...	...	...	...	...	...	...
138	19	7	7	7	6	6	7	6	6

	time	peruse01	peruse02	peruse03	peruse04	peruse05	peruse06	pereou01	pere
139	8	5	5	4	4	4	5	7	5
140	7	5	5	5	5	5	5	6	6
141	16	6	6	6	6	6	6	6	6
142	15	6	6	6	6	6	6	6	6
143	10	7	7	7	7	7	7	7	7
144	9	6	6	6	6	6	6	7	7
145	16	4	4	4	4	4	5	6	6
146	6	6	6	6	6	6	6	6	6
147	14	4	5	5	5	5	5	7	7
148	10	4	5	5	5	6	6	7	7
149	10	6	4	4	5	5	5	7	6
150	10	6	6	6	6	6	7	6	6
151	10	6	6	6	5	4	5	7	7
152	11	7	6	6	7	7	7	7	7
153	27	6	6	6	6	6	6	5	5
154	13	6	6	6	6	6	4	6	5
155	49	5	4	4	4	5	4	5	4
156	12	6	5	5	6	5	5	6	6
157	17	7	6	6	7	7	6	6	6
158	21	6	6	7	6	5	6	5	6
159	56	7	6	6	5	6	5	3	6
160	13	6	6	5	6	6	6	7	6
161	12	7	6	7	6	7	7	7	6
162	16	5	5	5	5	5	5	6	5
163	6	5	5	5	5	4	4	5	5
164	4	5	5	5	5	4	5	5	5
165	14	6	7	6	7	7	7	7	7
166	23	5	5	6	6	6	6	5	5
167	24	6	4	5	4	5	5	7	6

168 rows × 24 columns

In [10]:

```
# Explore data
df.dtypes
df.describe()
```

Out[10]:

	time	peruse01	peruse02	peruse03	peruse04	peruse05	peruse
count	168.000000	168.000000	168.000000	168.000000	168.000000	168.000000	168.000
mean	12.678571	5.720238	5.511905	5.619048	5.464286	5.505952	5.68452
std	7.666180	1.020306	0.966431	0.971507	1.037575	1.152955	1.05046
min	2.000000	1.000000	2.000000	2.000000	2.000000	2.000000	2.00000
25%	8.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.00000
50%	11.000000	6.000000	6.000000	6.000000	6.000000	6.000000	6.00000
75%	14.000000	6.000000	6.000000	6.000000	6.000000	6.000000	6.00000
max	56.000000	7.000000	7.000000	7.000000	7.000000	7.000000	7.00000

8 rows × 38 columns

In [13]:

```
df.columns
```

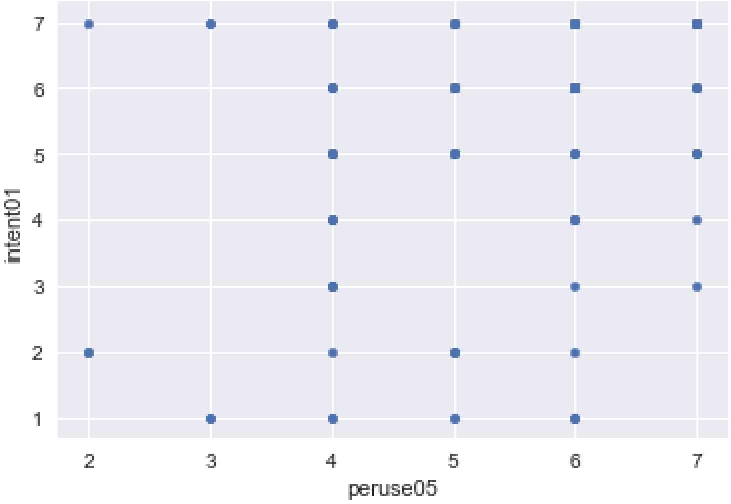
Out[13]:

```
Index(['time', 'peruse01', 'peruse02', 'peruse03', 'peruse04', 'peruse05',
       'peruse06', 'pereou01', 'pereou02', 'pereou03', 'pereou04', 'pereou0
5',
       'pereou06', 'intent01', 'intent02', 'intent03', 'operatingsys',
       'gender', 'educ_level', 'race_white', 'race_black', 'race_hisp',
       'race_asian', 'race_native', 'race_pacif', 'race_other', 'age',
       'citizenship', 'state', 'military', 'militbranch', 'familystruct',
       'children', 'income', 'employ', 'color', 'eatout', 'religion'],
      dtype='object')
```

Scatter target variable to each independence variables

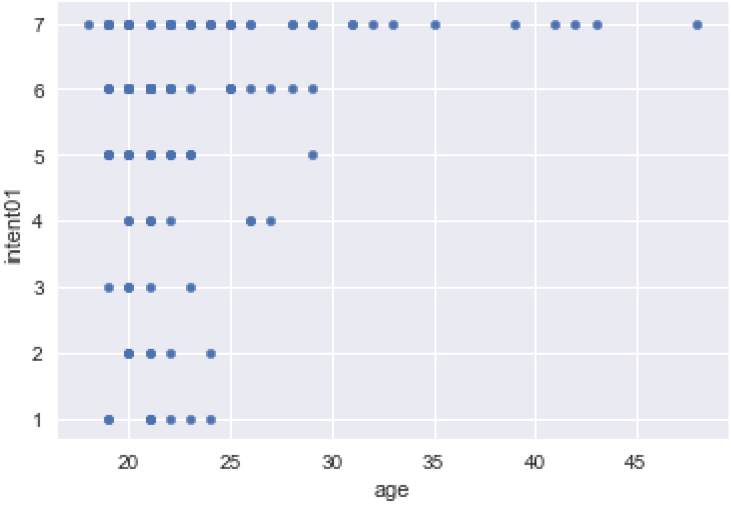
```
In [14]: df.plot.scatter(x='peruse05', y='intent01')
```

```
Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x1cf233ee6a0>
```



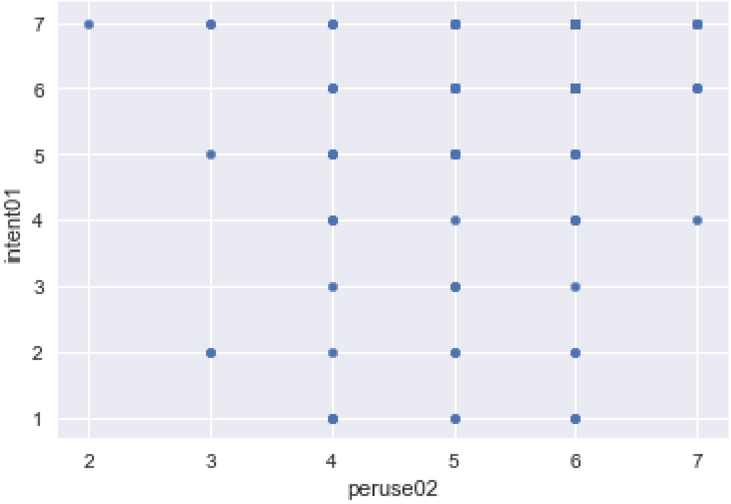
```
In [15]: df.plot.scatter(x='age', y='intent01')
```

```
Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x1cf25760d30>
```



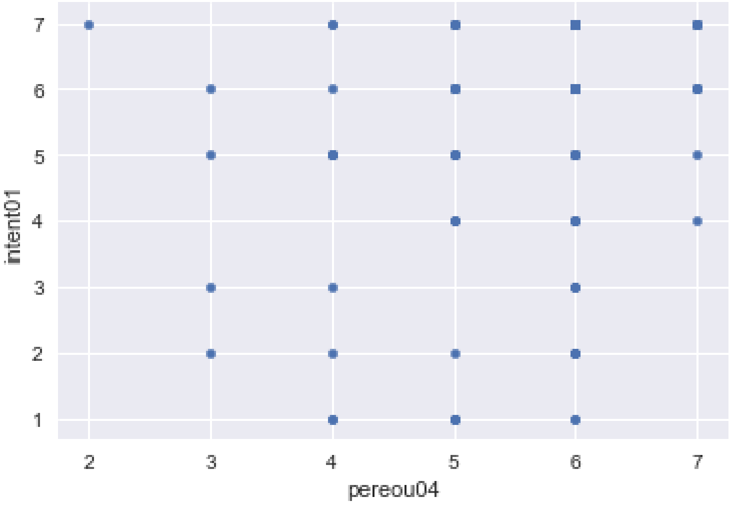
In [16]:
```
df.plot.scatter(x='peruse02', y='intent01')
```

Out[16]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1cf25806978>
```



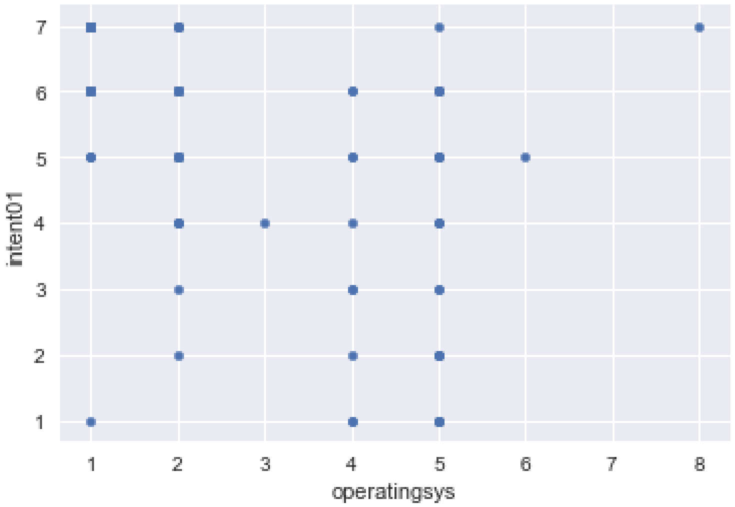
In [17]:
```
df.plot.scatter(x='pereou04', y='intent01')
```

Out[17]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1cf25733a90>
```



In [18]:
```
df.plot.scatter(x='operatingsys', y='intent01')
```

Out[18]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1cf2572a748>
```



In [31]:
```
dfnew = df[['intent01','peruse05','peruse02','pereou04','operatingsys','age']]
```

In [32]:
```
#Collinearity
dfnew.corr()
```

Out[32]:

	intent01	peruse05	peruse02	pereou04	operatingsys	age
intent01	1.000000	0.340873	0.323890	0.308517	-0.597892	0.231243
peruse05	0.340873	1.000000	0.642133	0.365810	-0.256058	0.011199
peruse02	0.323890	0.642133	1.000000	0.386221	-0.267895	-0.023765
pereou04	0.308517	0.365810	0.386221	1.000000	-0.171806	0.169316
operatingsys	-0.597892	-0.256058	-0.267895	-0.171806	1.000000	-0.101698
age	0.231243	0.011199	-0.023765	0.169316	-0.101698	1.000000

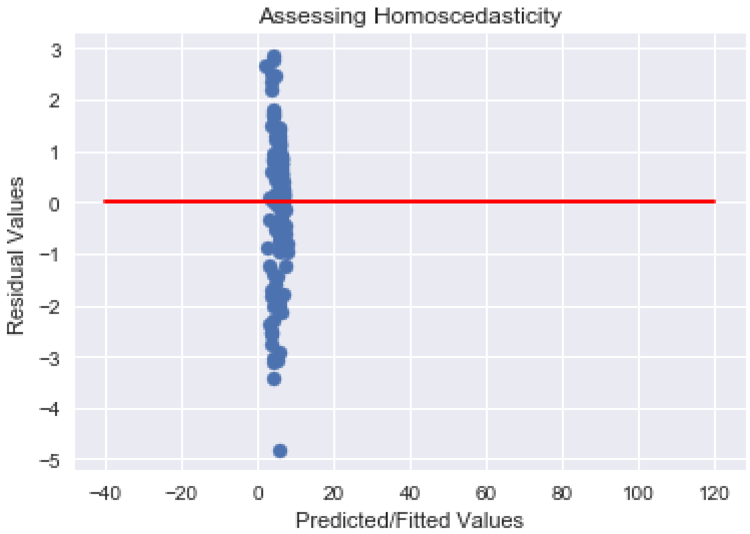
In [21]:
```
dfnew1 =
df[['peruse05','peruse02','pereou04','operatingsys','age','intent01']]
```

In [33]:

```
#Homoscedasticity

linreg2 = smf.ols('intent01 ~ peruse05 + peruse02 + pereou04 + operatingsys +
 age', df).fit()

#Assess homoscedasticity
plt.scatter(linreg2.fittedvalues, linreg2.resid)
plt.xlabel('Predicted/Fitted Values')
plt.ylabel('Residual Values')
plt.title('Assessing Homoscedasticity')
plt.plot([-40, 120],[0, 0], 'red', lw=2)   #Add horizontal line
plt.show()
```



the figure is evenly distributed across the x-asis, and there is one residual values is outlier

In [23]:
```
linreg2.summary()
```

Out[23]:

OLS Regression Results

Dep. Variable:	intent01	**R-squared:**	0.442
Model:	OLS	**Adj. R-squared:**	0.425
Method:	Least Squares	**F-statistic:**	25.24
Date:	Wed, 27 Sep 2017	**Prob (F-statistic):**	1.11e-18
Time:	22:10:28	**Log-Likelihood:**	-272.95
No. Observations:	165	**AIC:**	557.9
Df Residuals:	159	**BIC:**	576.5
Df Model:	5		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	2.5299	0.935	2.706	0.008	0.683	4.376
peruse05	0.1801	0.116	1.551	0.123	-0.049	0.409
peruse02	0.1255	0.140	0.894	0.373	-0.152	0.403
pereou04	0.2193	0.119	1.846	0.067	-0.015	0.454
operatingsys	-0.5315	0.066	-8.067	0.000	-0.662	-0.401
age	0.0583	0.022	2.629	0.009	0.014	0.102

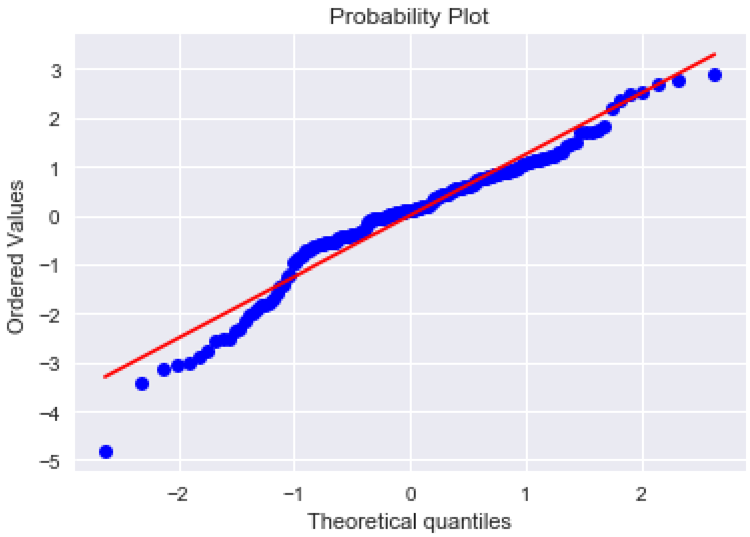
Omnibus:	19.577	**Durbin-Watson:**	2.042
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	25.259
Skew:	-0.732	**Prob(JB):**	3.27e-06
Kurtosis:	4.238	**Cond. No.**	235.

# With the data points in the model, the assumption of normally distributted residuals fails for the intent01 data

In [27]:

```
sts.probplot(linreg2.resid, dist="norm", plot=plt)
```

```
Out[27]:  ((array([-2.63619469, -2.31985126, -2.13890476, -2.00899334, -1.90613804,
                   -1.82021181, -1.74594095, -1.68021408, -1.62103502, -1.56704323,
                   -1.51726796, -1.47099118, -1.4276662 , -1.38686669, -1.34825346,
                   -1.31155201, -1.2765369 , -1.24302058, -1.21084528, -1.17987694,
                   -1.15000061, -1.12111696, -1.09313954, -1.06599258, -1.0396093 ,
                   -1.01393051, -0.98890348, -0.96448101, -0.94062067, -0.91728419,
                   -0.89443686, -0.87204715, -0.85008629, -0.82852795, -0.80734798,
                   -0.78652413, -0.76603589, -0.74586428, -0.7259917 , -0.70640177,
                   -0.68707927, -0.66800996, -0.64918054, -0.63057852, -0.61219219,
                   -0.59401054, -0.57602317, -0.55822027, -0.54059257, -0.52313129,
                   -0.50582807, -0.48867499, -0.47166452, -0.45478944, -0.4380429 ,
                   -0.42141832, -0.40490941, -0.38851013, -0.37221468, -0.35601748,
                   -0.33991315, -0.32389651, -0.30796253, -0.29210636, -0.2763233 ,
                   -0.26060878, -0.24495836, -0.22936771, -0.21383261, -0.19834896,
                   -0.18291271, -0.16751993, -0.15216674, -0.13684934, -0.12156398,
                   -0.10630698, -0.09107468, -0.07586348, -0.06066982, -0.04549015,
                   -0.03032096, -0.01515874,  0.        ,  0.01515874,  0.03032096,
                    0.04549015,  0.06066982,  0.07586348,  0.09107468,  0.10630698,
                    0.12156398,  0.13684934,  0.15216674,  0.16751993,  0.18291271,
                    0.19834896,  0.21383261,  0.22936771,  0.24495836,  0.26060878,
                    0.2763233 ,  0.29210636,  0.30796253,  0.32389651,  0.33991315,
                    0.35601748,  0.37221468,  0.38851013,  0.40490941,  0.42141832,
                    0.4380429 ,  0.45478944,  0.47166452,  0.48867499,  0.50582807,
                    0.52313129,  0.54059257,  0.55822027,  0.57602317,  0.59401054,
                    0.61219219,  0.63057852,  0.64918054,  0.66800996,  0.68707927,
                    0.70640177,  0.7259917 ,  0.74586428,  0.76603589,  0.78652413,
                    0.80734798,  0.82852795,  0.85008629,  0.87204715,  0.89443686,
                    0.91728419,  0.94062067,  0.96448101,  0.98890348,  1.01393051,
                    1.0396093 ,  1.06599258,  1.09313954,  1.12111696,  1.15000061,
                    1.17987694,  1.21084528,  1.24302058,  1.2765369 ,  1.31155201,
                    1.34825346,  1.38686669,  1.4276662 ,  1.47099118,  1.51726796,
                    1.56704323,  1.62103502,  1.68021408,  1.74594095,  1.82021181,
                    1.90613804,  2.00899334,  2.13890476,  2.31985126,  2.63619469]),
            array([-4.81103025, -3.43211475, -3.12885528, -3.06144544, -3.02083724,
                   -2.90348179, -2.77511709, -2.54711174, -2.53154524, -2.51777735,
                   -2.36040563, -2.30663839, -2.13135595, -2.00706935, -2.0010347 ,
                   -1.89824923, -1.83997254, -1.81808375, -1.77089529, -1.68891696,
                   -1.5931769 , -1.43688067, -1.40788173, -1.23710874, -1.22357649,
                   -0.96175847, -0.94496425, -0.88488152, -0.81987555, -0.78169586,
                   -0.73006268, -0.72115393, -0.69323049, -0.65098694, -0.61835422,
                   -0.60124559, -0.59530416, -0.59083268, -0.55343403, -0.55155645,
                   -0.5374009 , -0.53679205, -0.532556  , -0.47803448, -0.47351823,
                   -0.4405711 , -0.42977053, -0.42977053, -0.41264765, -0.40707964,
                   -0.39941729, -0.37149385, -0.37149385, -0.37149385, -0.36485034,
                   -0.33221762, -0.3211294 , -0.24970792, -0.15215501, -0.12947837,
                   -0.08723483, -0.08723483, -0.07675546, -0.07307927, -0.07081402,
                   -0.06595488, -0.05931138, -0.03747923, -0.03560165, -0.02003515,
                    0.0160092 ,  0.03158378,  0.0345511 ,  0.03683059,  0.04535168,
                    0.04535168,  0.05239709,  0.08196247,  0.08618428,  0.10175077,
                    0.10698334,  0.10886092,  0.11293901,  0.11778391,  0.12670691,
                    0.14627381,  0.15338395,  0.15956082,  0.16002746,  0.17082803,
                    0.17275783,  0.1760606 ,  0.20641383,  0.23433728,  0.26469051,
                    0.32108961,  0.34009006,  0.34378049,  0.40393476,  0.4125223 ,
                    0.43952055,  0.44999992,  0.45181277,  0.45367611,  0.47675466,
                    0.50672023,  0.52855238,  0.53872177,  0.55122902,  0.57022947,
                    0.57022947,  0.57061714,  0.57915247,  0.58500812,  0.59104278,
                    0.61770558,  0.63373872,  0.65885939,  0.73498206,  0.74136909,
                    0.74505952,  0.74505952,  0.7758713 ,  0.81286804,  0.8181006 ,
                    0.8189027 ,  0.82884453,  0.83125965,  0.83704441,  0.86684544,
                    0.87207801,  0.87576844,  0.88006772,  0.92168817,  0.93465397,
                    0.96439835,  0.96439835,  1.03824153,  1.04690805,  1.08987471,
                    1.10175077,  1.11410773,  1.13210401,  1.14625956,  1.16230695,
                    1.17238441,  1.22307006,  1.23245969,  1.28134674,  1.31686122,
                    1.42730904,  1.44272524,  1.49880799,  1.69631468,  1.69631468,
                    1.71531513,  1.75459136,  1.81286804,  2.20460714,  2.34666883,
                    2.48778623,  2.51780845,  2.67176045,  2.78422019,  2.87637729])),
           (1.2513292455994429, 1.1907088206622858e-14, 0.9759639312289522))
```



```
In [34]: pearsonr(df.intent01,df.operatingsys)

Out[34]: (-0.5978921715528297, 1.1613124760576334e-17)
```

Equation: intent01 = 2.5299 + 0.1801 *peruse05 + 0.1255* peruse02 + 0.2193 *pereou04 - 0.5315* operatingsys + 0.0583* age

intent01 strong dependance on beta 0 and operationsys