# Background

This project is of an applied nature and uses data that are available in the data file Capstone-HousePrices. The source of these data is Anglin and Gencay, “Semiparametric Estimation of a Hedonic Price Function”(Journal of Applied Econometrics 11, 1996, pages 633-648). We consider the modeling and prediction of house prices. Data are available for 546 observations of the following variables:

• sell: Sale price of the house

• lot: Lot size of the property in square feet

• bdms: Number of bedrooms

• fb: Number of full bathrooms

• sty: Number of stories excluding basement

• drv: Dummy that is 1 if the house has a driveway and 0 otherwise

• rec: Dummy that is 1 if the house has a recreational room and 0 otherwise

• ffin: Dummy that is 1 if the house has a full finished basement and 0 otherwise

• ghw: Dummy that is 1 if the house uses gas for hot water heating and 0 otherwise

• ca: Dummy that is 1 if there is central air conditioning and 0 otherwise

• gar: Number of covered garage places

• reg: Dummy that is 1 if the house is located in a preferred neighborhood of the city and 0 otherwise

• obs: Observation number, needed in part (h)

In [2]:

```
## load in packages

%matplotlib inline
import sys
sys.path.append('/Users/CJ/Documents/bitbucket/xforex_v1/xforex_v3')
import pandas as pd
import matplotlib.pyplot as plt
from datetime import datetime
from xforex.BackTesting.econometrics_tools import Econometrics_Tool
import numpy as np
import pprint as pp
import pandas
import statsmodels.api as sm
```

# a

Consider a linear model where the sale price of a house is the dependent variable and the explanatory variables are the other variables given above. Perform a test for linearity. What do you conclude based on the test result?

**ans:**

1. check OLS regression results Jarque-Bera statics is 247.620 and alpha for JB test 1.70e-54. Therefore the normality of residuals are rejected at 5% level.
2. check the plot of residuals vs fitted values

the plot (row1, column2) shows obvious Heteroscedasticity.

1. check the y plot

The data seems break at around observations 300

1. check the historgram of residuals

The residuals seems have mean 0 and positive skewness

The test above shows that the fitting may violate the OLS hypothsis. We may need further transformations.

In [3]:

```
dat= pd.read_csv('/Users/CJ/Documents/bitbucket/xforex_v1/xforex_v3/training/eco
nometrics-cousera/week7-project/housing-prices.txt',
                  sep = '\t')
dat.index = dat.obs
del dat['obs']
X = sm.add_constant(dat.drop('sell', axis=1))
y = dat['sell']

def ols(y, X):
    ols_model1 = sm.OLS(y, X)
    ols_re1 = ols_model1.fit()
    print ols_re1.summary()

    plt.figure(1, figsize=(14, 8))

    plt.subplot(221)
    plt.plot(y)
    plt.ylabel('sell')
    plt.subplot(222)
    plt.xlabel('fitted value')
    plt.ylabel('residuals')
    plt.plot(ols_re1.fittedvalues, ols_re1.resid, '.')

    plt.subplot(223)
    plt.hist(ols_re1.resid, bins=30)
    return ols_re1
```

# b

Now consider a linear model where the log of the sale price of the house is the dependent variable and the explanatory variables are as before. Perform again the test for linearity. What do you conclude now?

**ans:**

1. check OLS regression results Jarque-Bera statics is 8.443 and alpha for JB test 0.0147. JB statics shows better results than in (a). But the JB test is got rejected at 5% percent level, which indicates the non-normlity in residuals.
2. check the plot of residuals vs fitted values

the plot (row1, column2) shows no obvious Heteroscedasticity.

1. check the y plot

The data seems break at around observations 300

1. check the historgram of residuals

The residuals seems have mean 0 and negative skewness

The test above shows that the fitting may slightly violate the OLS hypothsis.

In [4]:

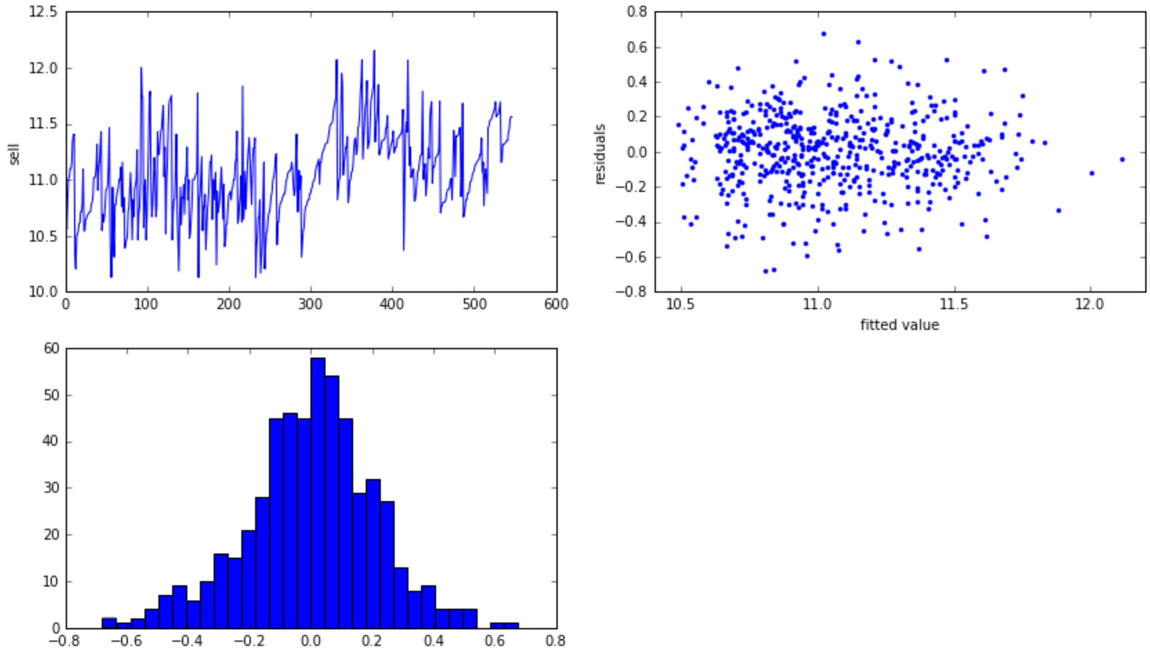
```
import math
y_log = dat['sell'].apply(math.log)

ols(y_log, X)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   sell   R-squared:
    0.677
Model:                            OLS   Adj. R-squared:
    0.670
Method:                 Least Squares   F-statistic:
    101.6
Date:                Tue, 04 Oct 2016   Prob (F-statistic):
3.67e-123
Time:                        12:12:36   Log-Likelihood:
   73.873
No. Observations:                 546   AIC:
   -123.7
Df Residuals:                     534   BIC:
   -72.11
Df Model:                          11

Covariance Type:            nonrobust

==============================================================================
                 coef    std err          t      P>|t|      [95.0% C
onf. Int.]
------------------------------------------------------------------------------
const         10.0256      0.047    212.210      0.000         9.933
   10.118
lot         5.057e-05   4.85e-06     10.418      0.000       4.1e-05
  6.01e-05
bdms           0.0340      0.015      2.345      0.019         0.006
    0.063
fb             0.1678      0.021      8.126      0.000         0.127
    0.208
sty            0.0923      0.013      7.197      0.000         0.067
    0.117
drv            0.1307      0.028      4.610      0.000         0.075
    0.186
rec            0.0735      0.026      2.792      0.005         0.022
    0.125
ffin           0.0994      0.022      4.517      0.000         0.056
    0.143
ghw            0.1784      0.045      4.000      0.000         0.091
    0.266
ca             0.1780      0.022      8.262      0.000         0.136
    0.220
gar            0.0508      0.012      4.358      0.000         0.028
    0.074
reg            0.1271      0.023      5.496      0.000         0.082
    0.173
==============================================================================
Omnibus:                        7.621   Durbin-Watson:
    1.510
Prob(Omnibus):                  0.022   Jarque-Bera (JB):
    8.443
Skew:                          -0.199   Prob(JB):
   0.0147
Kurtosis:                       3.461   Cond. No.
```

```
  3.07e+04
======================================================================
==========

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors
 is correctly specified.
[2] The condition number is large, 3.07e+04. This might indicate tha
t there are
strong multicollinearity or other numerical problems.
```

Out[4]:

```
<statsmodels.regression.linear_model.RegressionResultsWrapper at 0x1
11ead910>
```




## c

Continue with the linear model from question (b). Estimate a model that includes both the lot size variable and its logarithm, as well as all other explanatory variables without transformation. What is your conclusion, should we include lot size itself or its logarithm?

**ans**:

From the regression results below, the pvalue of lot is 0.359 and the pvalue of log-lot is 0.000. Therefore, we should include the logarithem of lot size instead of lot.

In [5]:

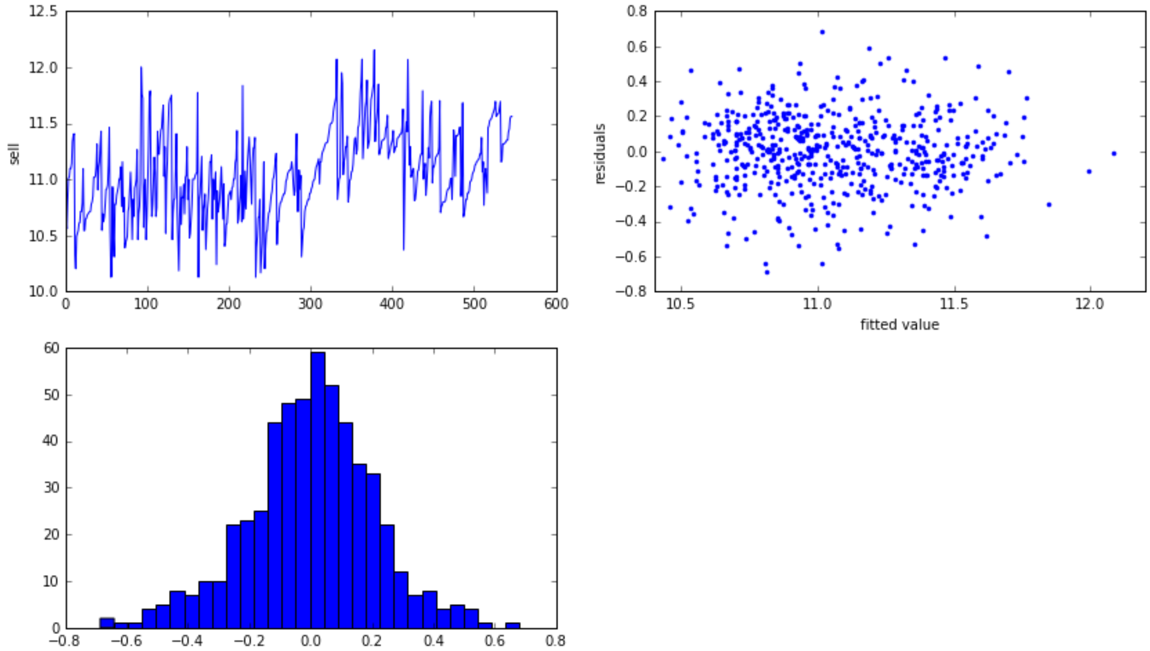
```
dat['log-lot'] = dat['lot'].apply(math.log)
X = sm.add_constant(dat.drop('sell', axis=1))

y = dat['sell'].apply(math.log)
ols(y, X)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   sell   R-squared:                       0.687
Model:                            OLS   Adj. R-squared:                  0.680
Method:                 Least Squares   F-statistic:                     97.51
Date:                Tue, 04 Oct 2016   Prob (F-statistic):         6.43e-126
Time:                        12:12:42   Log-Likelihood:                 82.843
No. Observations:                 546   AIC:                            -139.7
Df Residuals:                     533   BIC:                            -83.75
Df Model:                          12
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
const          7.1505      0.683     10.469      0.000         5.809     8.492
lot         -1.49e-05   1.62e-05     -0.918      0.359     -4.68e-05   1.7e-05
bdms           0.0349      0.014      2.442      0.015         0.007     0.063
fb             0.1659      0.020      8.161      0.000         0.126     0.206
sty            0.0912      0.013      7.224      0.000         0.066     0.116
drv            0.1068      0.028      3.752      0.000         0.051     0.163
rec            0.0547      0.026      2.078      0.038         0.003     0.106
ffin           0.1052      0.022      4.848      0.000         0.063     0.148
ghw            0.1791      0.044      4.079      0.000         0.093     0.265
ca             0.1643      0.021      7.657      0.000         0.122     0.206
gar            0.0483      0.011      4.203      0.000         0.026     0.071
reg            0.1344      0.023      5.884      0.000         0.090     0.179
log-lot        0.3827      0.091      4.219      0.000         0.205     0.561
==============================================================================
Omnibus:                        7.927   Durbin-Watson:                   1.525
Prob(Omnibus):                  0.019   Jarque-Bera (JB):                9.364
Skew:                          -0.180   Prob(JB):
```

```
    0.00926
Kurtosis:                        3.531   Cond. No.
   4.27e+05
======================================================================
==========

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors
 is correctly specified.
[2] The condition number is large, 4.27e+05. This might indicate tha
t there are
strong multicollinearity or other numerical problems.
```

Out[5]:

```
<statsmodels.regression.linear_model.RegressionResultsWrapper at 0x1
099dffd0>
```




# d

Consider now a model where the log of the sale price of the house is the dependent variable and the explanatory variables are the log transformation of lot size, with all other explanatory variables as before. We now consider interaction effects of the log lot size with the other variables. Construct these interaction variables. How many are individually significant?

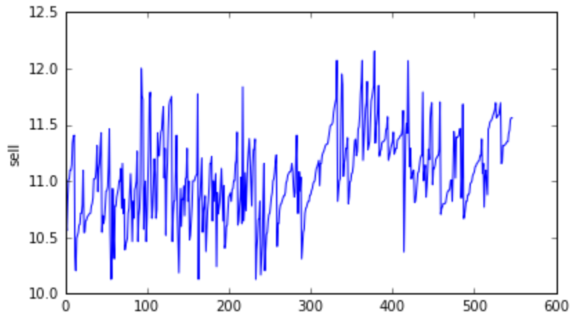
**ans**: from the regression results below, we can none of the interaction terms are signicant at 5% level

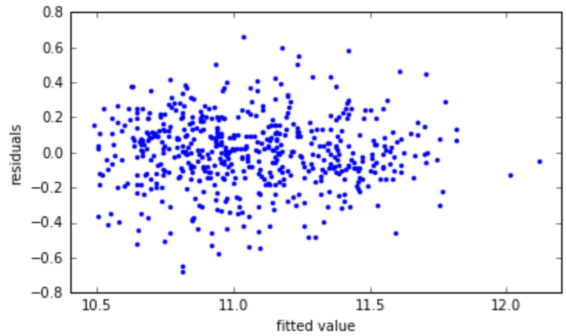
In [47]:

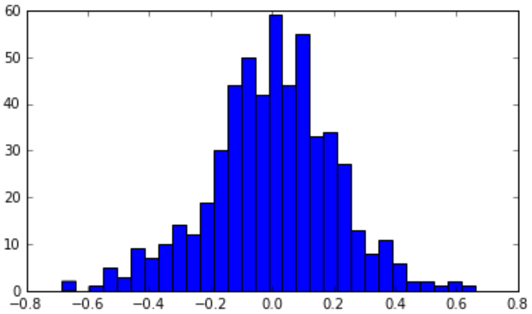
```
import itertools
rm_col = ['log-lot', 'lot','sell']
iter_col1 = [x for x in list(dat.columns) if x not in rm_col]
iter_col2 = ['log-lot']

def get_iteraction_terms(df, iter_col1, iter_col2):
    inter_terms =[]
    for item in list(itertools.product(iter_col1, iter_col2)):
        name = item[0]+' MULTIPLY '+item[1]
        df[name] = df[item[0]] * df[item[1]]
        inter_terms.append(name)
    return [df,inter_terms]

dat_cp = dat.copy()
dat_cp = get_iteraction_terms(dat_cp, iter_col1, iter_col2)[0]

print dat_cp.columns

X = sm.add_constant(dat_cp.drop(['sell', 'lot'], axis=1))
y = dat['sell'].apply(math.log)
model_with_interaction = ols(y, X)
```

```
Index([u'sell', u'lot', u'bdms', u'fb', u'sty', u'drv', u'rec', u'ff
in',
       u'ghw', u'ca', u'gar', u'reg', u'log-lot', u'bdms MULTIPLY lo
g-lot',
       u'fb MULTIPLY log-lot', u'sty MULTIPLY log-lot',
       u'drv MULTIPLY log-lot', u'rec MULTIPLY log-lot',
       u'ffin MULTIPLY log-lot', u'ghw MULTIPLY log-lot',
       u'ca MULTIPLY log-lot', u'gar MULTIPLY log-lot',
       u'reg MULTIPLY log-lot'],
      dtype='object')
                            OLS Regression Results

=====================================================================
==========
Dep. Variable:                   sell   R-squared:
     0.695
Model:                            OLS   Adj. R-squared:
     0.683
Method:                 Least Squares   F-statistic:
     56.89
Date:                Wed, 05 Oct 2016   Prob (F-statistic):
 2.26e-120
Time:                        10:44:36   Log-Likelihood:
    89.971
No. Observations:                 546   AIC:
    -135.9
Df Residuals:                     524   BIC:
    -41.28
Df Model:                          21

Covariance Type:            nonrobust

=====================================================================
=====================
                            coef    std err          t      P>|t|
   [95.0% Conf. Int.]
---------------------------------------------------------------------
---------------------
const                     8.9665      1.071      8.375      0.000
      6.863    11.070
bdms                      0.0191      0.327      0.058      0.953
     -0.623     0.661
fb                       -0.3682      0.429     -0.858      0.391
     -1.211     0.475
sty                       0.4889      0.310      1.579      0.115
     -0.120     1.097
drv                      -1.4634      0.717     -2.040      0.042
     -2.872    -0.054
rec                       1.6740      0.656      2.552      0.011
      0.385     2.963
ffin                     -0.0318      0.446     -0.071      0.943
     -0.907     0.843
ghw                      -0.5059      0.903     -0.560      0.575
     -2.279     1.268
ca                       -0.3403      0.496     -0.686      0.493
     -1.315     0.634
gar                       0.4019      0.259      1.554      0.121
     -0.106     0.910
reg                       0.1185      0.480      0.247      0.805
     -0.824     1.061
log-lot                   0.1527      0.128      1.190      0.235
```

```
      -0.099      0.405
bdms MULTIPLY log-lot       0.0021       0.039       0.054       0.957
      -0.074      0.078
fb MULTIPLY log-lot         0.0620       0.050       1.237       0.217
      -0.036      0.161
sty MULTIPLY log-lot       -0.0464       0.036      -1.290       0.198
      -0.117      0.024
drv MULTIPLY log-lot        0.1915       0.087       2.193       0.029
       0.020      0.363
rec MULTIPLY log-lot       -0.1885       0.076      -2.468       0.014
      -0.338     -0.038
ffin MULTIPLY log-lot       0.0159       0.053       0.301       0.763
      -0.088      0.120
ghw MULTIPLY log-lot        0.0811       0.107       0.759       0.448
      -0.129      0.291
ca MULTIPLY log-lot         0.0595       0.058       1.026       0.305
      -0.054      0.174
gar MULTIPLY log-lot       -0.0414       0.030      -1.372       0.171
      -0.101      0.018
reg MULTIPLY log-lot        0.0015       0.056       0.027       0.978
      -0.108      0.112
====================================================================
==========
Omnibus:                         7.141   Durbin-Watson:
     1.524
Prob(Omnibus):                   0.028   Jarque-Bera (JB):
     8.203
Skew:                           -0.173   Prob(JB):
    0.0165
Kurtosis:                        3.491   Cond. No.
  4.77e+03
====================================================================
==========

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors
 is correctly specified.
[2] The condition number is large, 4.77e+03. This might indicate tha
t there are
strong multicollinearity or other numerical problems.
```



# e

Perform an F-test for the joint significance of the interaction effects from question (d)

**ans:** In F-test, the f statics is 1.47119504552 and the critical value for df1= 10 and df2 = 524 is 1.84876723495. since 1.47< 1.84

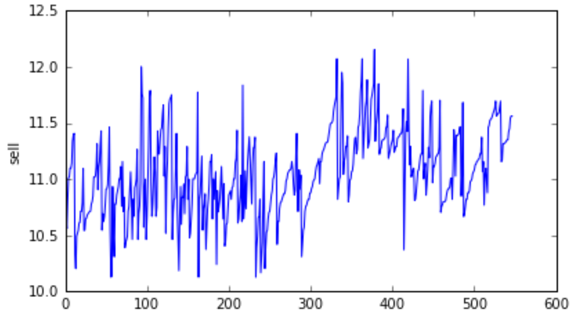
so the cannot reject null hypothesis. The interaction terms are not jointly significant.

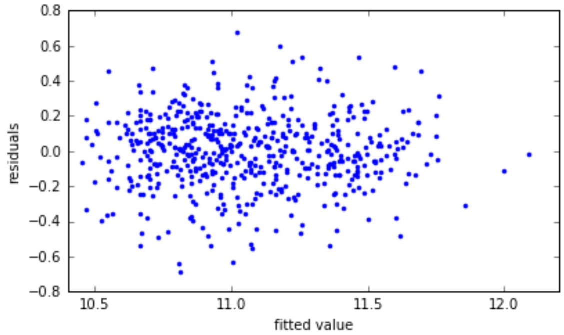
In [46]:

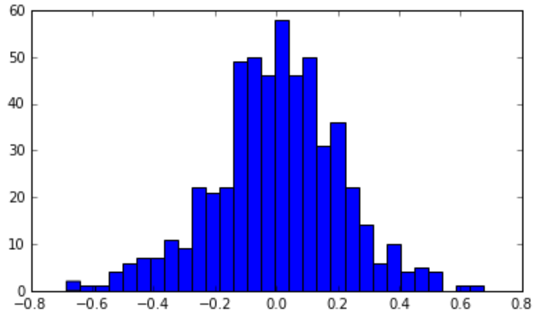
```
from scipy.stats import f

X = sm.add_constant(dat.drop(['sell', 'lot'], axis=1))
y = dat['sell'].apply(math.log)

model_no_iteraction = ols(y, X)

r_unrestricted= sum((model_with_interaction.fittedvalues - y)**2)
r_restricted= sum((model_no_iteraction.fittedvalues - y)**2)

g =10
n= 546
k = model_with_interaction.df_model
f_stat = ((r_restricted - r_unrestricted)/g)/(r_unrestricted/(n-k-1))
f_critical = f.ppf(1-0.05, g, n-k-1)
print f_stat, f_critical,(n-k-1)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   sell   R-squared:                       0.687
Model:                            OLS   Adj. R-squared:                  0.680
Method:                 Least Squares   F-statistic:                     106.3
Date:                Wed, 05 Oct 2016   Prob (F-statistic):          9.24e-127
Time:                        10:24:49   Log-Likelihood:                 82.412
No. Observations:                 546   AIC:                            -140.8
Df Residuals:                     534   BIC:                            -89.19
Df Model:                          11

Covariance Type:            nonrobust

==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
const          7.7451      0.216     35.801      0.000         7.320     8.170
bdms           0.0344      0.014      2.410      0.016         0.006     0.062
fb             0.1658      0.020      8.154      0.000         0.126     0.206
sty            0.0917      0.013      7.268      0.000         0.067     0.116
drv            0.1102      0.028      3.904      0.000         0.055     0.166
rec            0.0580      0.026      2.225      0.026         0.007     0.109
ffin           0.1045      0.022      4.817      0.000         0.062     0.147
ghw            0.1790      0.044      4.079      0.000         0.093     0.265
ca             0.1664      0.021      7.799      0.000         0.125     0.208
gar            0.0480      0.011      4.178      0.000         0.025     0.070
reg            0.1319      0.023      5.816      0.000         0.087     0.176
log-lot        0.3031      0.027     11.356      0.000         0.251     0.356
==============================================================================
Omnibus:                        7.856   Durbin-Watson:                   1.525
Prob(Omnibus):                  0.020   Jarque-Bera (JB):                9.155
Skew:                          -0.184   Prob(JB):                       0.0103
Kurtosis:                       3.517   Cond. No.
```

```
      228.
====================================================================
==========

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors
 is correctly specified.
1.47119504552 1.84876723495 524.0
```





## f

Now perform model specification on the interaction variables using the general-to-specific approach. (Only eliminate the interaction effects.)

**ans:** check below for the OLS results. Only rec MULTIPLY log-lot is remained among interaction terms after general to specific elimination

In [64]:

```
import itertools
rm_col = ['log-lot', 'lot','sell']
iter_col1 = [x for x in list(dat.columns) if x not in rm_col]
iter_col2 = ['log-lot']

dat_cp = dat.copy()
dat_cp, it_terms = get_iteraction_terms(dat_cp, iter_col1, iter_col2)

X = sm.add_constant(dat_cp.drop(['sell', 'lot'], axis=1))
y = dat['sell'].apply(math.log)

def g2s_OLS(y, X, eliminate_var, level=0.05):
    X_new = X.copy()
    while(True):
        model = sm.OLS(y, X_new)
        re = model.fit()
        max_p = re.pvalues[eliminate_var].max()
        if max_p > level:
            max_row = re.pvalues[eliminate_var].argmax()
            X_new = X_new.drop(max_row, axis = 1)
            t =0
            for name in X_new.columns:
                if name in eliminate_var:
                    t = t + 1
            if t == 0:
                return re
        else:
            return re
g2s_OLS(y, X.copy(), it_terms).summary()
```

```
Index([u'sell', u'lot', u'bdms', u'fb', u'sty', u'drv', u'rec', u'ff
in',
       u'ghw', u'ca', u'gar', u'reg', u'log-lot', u'bdms MULTIPLY lo
g-lot',
       u'fb MULTIPLY log-lot', u'sty MULTIPLY log-lot',
       u'drv MULTIPLY log-lot', u'rec MULTIPLY log-lot',
       u'ffin MULTIPLY log-lot', u'ghw MULTIPLY log-lot',
       u'ca MULTIPLY log-lot', u'gar MULTIPLY log-lot',
       u'reg MULTIPLY log-lot'],
      dtype='object')
```

Out[64]:

OLS Regression Results

| | | | |
|---|---|---|---|
| **Dep. Variable:** | sell | **R-squared:** | 0.689 |
| **Model:** | OLS | **Adj. R-squared:** | 0.682 |
| **Method:** | Least Squares | **F-statistic:** | 98.59 |
| **Date:** | Wed, 05 Oct 2016 | **Prob (F-statistic):** | 8.71e-127 |
| **Time:** | 11:02:34 | **Log-Likelihood:** | 84.909 |
| **No. Observations:** | 546 | **AIC:** | -143.8 |
| **Df Residuals:** | 533 | **BIC:** | -87.88 |
| **Df Model:** | 12 | | |
| **Covariance Type:** | nonrobust | | |

| | coef | std err | t | P>\|t\| | [95.0% Conf. Int.] |
|---|---|---|---|---|---|
| **const** | 7.5907 | 0.227 | 33.505 | 0.000 | 7.146 8.036 |
| **bdms** | 0.0384 | 0.014 | 2.680 | 0.008 | 0.010 0.067 |
| **fb** | 0.1632 | 0.020 | 8.043 | 0.000 | 0.123 0.203 |
| **sty** | 0.0908 | 0.013 | 7.220 | 0.000 | 0.066 0.115 |
| **drv** | 0.1131 | 0.028 | 4.018 | 0.000 | 0.058 0.168 |
| **rec** | 1.4431 | 0.626 | 2.304 | 0.022 | 0.212 2.674 |
| **ffin** | 0.1045 | 0.022 | 4.835 | 0.000 | 0.062 0.147 |
| **ghw** | 0.1843 | 0.044 | 4.208 | 0.000 | 0.098 0.270 |
| **ca** | 0.1659 | 0.021 | 7.804 | 0.000 | 0.124 0.208 |
| **gar** | 0.0481 | 0.011 | 4.206 | 0.000 | 0.026 0.071 |
| **reg** | 0.1337 | 0.023 | 5.917 | 0.000 | 0.089 0.178 |
| **log-lot** | 0.3202 | 0.028 | 11.562 | 0.000 | 0.266 0.375 |
| **rec MULTIPLY log-lot** | -0.1611 | 0.073 | -2.213 | 0.027 | -0.304 -0.018 |

| | | | |
|---|---|---|---|
| **Omnibus:** | 8.625 | **Durbin-Watson:** | 1.522 |
| **Prob(Omnibus):** | 0.013 | **Jarque-Bera (JB):** | 10.348 |
| **Skew:** | -0.190 | **Prob(JB):** | 0.00566 |
| **Kurtosis:** | 3.558 | **Cond. No.** | 676. |

# g

One may argue that some of the explanatory variables are endogenous and that there may be omitted variables. For example, the ‘condition’ of the house in terms of how it is maintained is not a variable (and difficult to measure) but will affect the house price. It will also affect, or be reflected in, some of the other variables, such as whether the house has an air conditioning (which is mostly in newer houses). If the condition of the house is missing, will the effect of air conditioning on the (log of the) sale price be over- or underestimated? (For this question no computer calculations are required.)

**ans:**

The air conditioning one the sale price will be overestimated. the effect of the air condition contains both itself together with the effect of condition of the hourse. And often these two are positive correlated. So if price is high due to better condition, with the condition variable missing, the effect will be reflected in the air conditioning parameter.

# h

Finally we analyze the predictive ability of the model. Consider again the model where the log of the sale price of the house is the dependent variable and the explanatory variables are the log transformation of lot size, with all other explanatory variables in their original form (and no interaction effects). Estimate the parameters of the model using the first 400 observations. Make predictions on the log of the price and calculate the MAE for the other 146 observations. How good is the predictive power of the model (relative to the variability in the log of the price)?

**ans:**

Check the below out-of-sample evaluation metrics:

- rmse if sale as y: 15056.1064008
- rmse log sale as y: 0.18223155588
- mae if sale as y: 11273.7232733
- mae log sale as y: 0.137353613959

Therefore, using log sale as y has less error.

In [81]:

```
# log model
y_log = dat['sell'].apply(math.log)
X = sm.add_constant(dat.drop(['sell', 'log-lot'], axis=1))
model = sm.OLS(y_log[:400], X[:400])
re = model.fit()
y_pred_log = re.predict(X[400:])

# not log model
y = dat['sell']
model = sm.OLS(y[:400], X[:400])
re = model.fit()
y_pred = re.predict(X[400:])

def rmse(y, y_fit):
    return math.sqrt(sum((y-y_fit)**2)/len(y))

def mae(y, y_fit):
    return sum(abs(y-y_fit))/len(y)

print 'rmse if sale as y: ',rmse(y[400:], y_pred)
print 'rmse log sale as y:', rmse(y_log[400:], y_pred_log)

print 'mae if sale as y: ',mae(y[400:], y_pred)
print 'mae log sale as y:', mae(y_log[400:], y_pred_log)
```

```
rmse if sale as y:  15056.1064008
rmse log sale as y: 0.18223155588
mae if sale as y:  11273.7232733
mae log sale as y: 0.137353613959
```

In [76]:

```
len(y)
```

Out[76]:

```
546
```

In [ ]:

```
```