In [24]:
```
import pandas as pd
from pandas import read_csv
df = pd.read_csv("sunrise_comp_data.csv")

df = df.ix[:, :]
```

In [25]:
```
df['pre_bid_price_inst1'] = df['bid_price_inst1'].shift(1)
df['pre_bid_size_inst1'] = df['bid_size_inst1'].shift(1)
df['pre_ask_price_inst1'] = df['ask_price_inst1'].shift(1)
df['pre_ask_size_inst1'] = df['ask_size_inst1'].shift(1)
df['pre_mid_price_inst1'] = df['mid_price_inst1'].shift(1)

df['pre_bid_price_inst2'] = df['bid_price_inst2'].shift(1)
df['pre_bid_size_inst2'] = df['bid_size_inst2'].shift(1)
df['pre_ask_price_inst2'] = df['ask_price_inst2'].shift(1)
df['pre_ask_size_inst2'] = df['ask_size_inst2'].shift(1)
df['pre_mid_price_inst2'] = df['mid_price_inst2'].shift(1)

df['pre_bid_price_inst3'] = df['bid_price_inst3'].shift(1)
df['pre_bid_size_inst3'] = df['bid_size_inst3'].shift(1)
df['pre_ask_price_inst3'] = df['ask_price_inst3'].shift(1)
df['pre_ask_size_inst3'] = df['ask_size_inst3'].shift(1)
df['pre_mid_price_inst3'] = df['mid_price_inst3'].shift(1)
```

In [26]:
```
df = df.drop(0)
```

In [29]:
```
df
```

Out[29]:

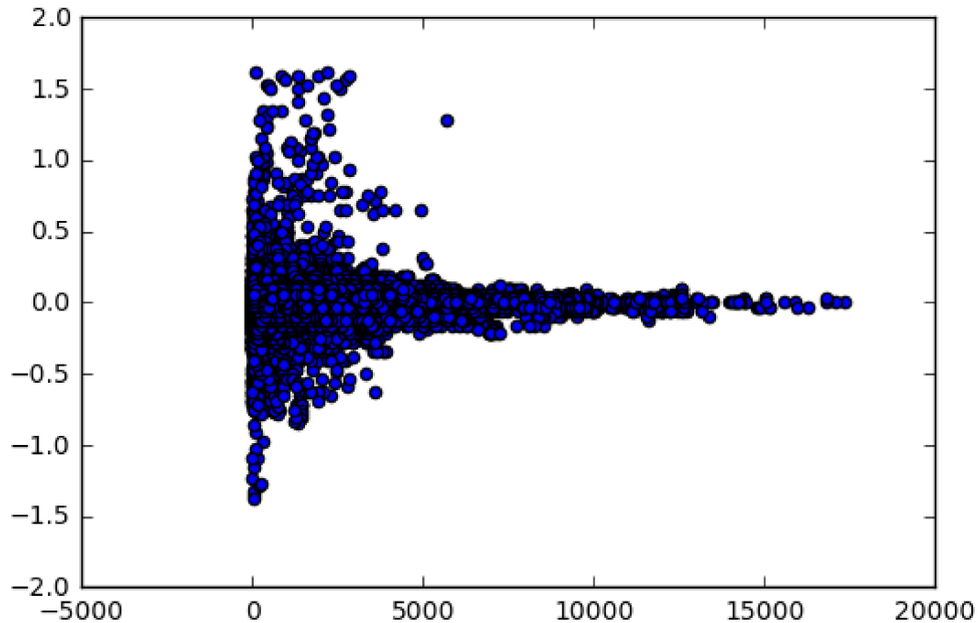
	date	time	bid_price_inst1	bid_size_inst1	ask_price_inst1	ask_size_inst1	mid_price_inst1	net_traded_inst1	bid_price_inst2	bid_size_inst2	...	pre_b
1	0	25201	239.78125	558	239.81250	156	239.796874	0	364.21875	150	...	2036.
2	0	25202	239.78125	580	239.81250	116	239.796874	0	364.21875	192	...	2028.
3	0	25203	239.81250	206	239.84375	432	239.828124	90	364.21875	312	...	2028.
4	0	25204	239.81250	230	239.84375	312	239.828124	0	364.21875	321	...	2016.
5	0	25205	239.81250	230	239.84375	278	239.828124	0	364.21875	321	...	2012.
6	0	25206	239.81250	212	239.84375	440	239.828124	0	364.21875	321	...	28.0
7	0	25207	239.81250	126	239.84375	470	239.828124	0	364.21875	315	...	224.0
8	0	25208	239.81250	280	239.84375	448	239.828124	0	364.21875	327	...	248.0
9	0	25209	239.84375	202	239.87500	520	239.859374	232	364.31250	147	...	300.0
10	0	25210	239.84375	252	239.87500	496	239.859374	0	364.31250	135	...	288.0
11	0	25211	239.84375	178	239.87500	646	239.859374	-22	364.31250	33	...	288.0
12	0	25212	239.84375	356	239.87500	560	239.859374	0	364.31250	90	...	300.0
13	0	25213	239.84375	312	239.87500	566	239.859374	0	364.31250	102	...	284.0
14	0	25214	239.84375	316	239.87500	568	239.859374	0	364.31250	117	...	280.0
15	0	25215	239.84375	320	239.87500	572	239.859374	0	364.31250	105	...	260.0
16	0	25216	239.84375	314	239.87500	572	239.859374	0	364.31250	114	...	260.0
17	0	25217	239.84375	282	239.87500	586	239.859374	-6	364.21875	441	...	972.0
18	0	25218	239.84375	268	239.87500	612	239.859374	-2	364.21875	405	...	1032.
19	0	25219	239.84375	154	239.87500	662	239.859374	0	364.21875	387	...	1032.
20	0	25220	239.84375	154	239.87500	676	239.859374	0	364.21875	381	...	1040.
21	0	25221	239.84375	138	239.87500	676	239.859374	0	364.21875	381	...	980.0
22	0	25222	239.84375	138	239.87500	676	239.859374	0	364.21875	378	...	968.0
23	0	25223	239.81250	336	239.84375	186	239.828124	-158	364.21875	351	...	968.0
24	0	25224	239.81250	396	239.84375	72	239.828124	0	364.21875	360	...	1160.(

	date	time	bid_price_inst1	bid_size_inst1	ask_price_inst1	ask_size_inst1	mid_price_inst1	net_traded_inst1	bid_price_inst2	bid_size_inst2	...	pre_b
25	0	25225	239.84375	118	239.87500	594	239.859374	-100	364.21875	393	...	1160.(
26	0	25226	239.84375	168	239.87500	580	239.859374	0	364.21875	375	...	1184.(
27	0	25227	239.84375	168	239.87500	590	239.859374	0	364.21875	366	...	776.0
28	0	25228	239.81250	506	239.84375	42	239.828124	18	364.21875	363	...	772.0
29	0	25229	239.84375	180	239.87500	610	239.859374	66	364.21875	378	...	784.0
30	0	25230	239.84375	228	239.87500	600	239.859374	0	364.21875	396	...	788.0
...	...	...	...	...	...	...	...	...	...	...	...	...
1781912	228	53911	237.68750	1002	237.71875	986	237.703124	0	360.09375	414	...	1592(
1781913	228	53912	237.68750	1008	237.71875	1044	237.703124	0	360.09375	414	...	1572(
1781914	228	53913	237.68750	596	237.71875	1558	237.703124	-92	360.09375	234	...	14964
1781915	228	53914	237.65625	776	237.68750	738	237.671874	-600	360.00000	696	...	1474(
1781916	228	53915	237.65625	482	237.68750	1122	237.671874	0	360.00000	780	...	14684
1781917	228	53916	237.65625	480	237.68750	1134	237.671874	0	360.00000	918	...	1434(
1781918	228	53917	237.65625	78	237.68750	1564	237.671874	-190	360.00000	711	...	14024
1781919	228	53918	237.65625	22	237.68750	1322	237.671874	-90	360.00000	831	...	14124
1781920	228	53919	237.65625	512	237.68750	1120	237.671874	-16	360.00000	858	...	1388(
1781921	228	53920	237.65625	1020	237.68750	614	237.671874	0	360.09375	159	...	13904
1781922	228	53921	237.65625	838	237.68750	654	237.671874	0	360.09375	198	...	1356(
1781923	228	53922	237.65625	1256	237.68750	338	237.671874	0	360.09375	417	...	13532
1781924	228	53923	237.65625	806	237.68750	578	237.671874	88	360.09375	189	...	1350(
1781925	228	53924	237.65625	852	237.68750	570	237.671874	0	360.09375	189	...	13204
1781926	228	53925	237.65625	850	237.68750	670	237.671874	-2	360.09375	189	...	1308(
1781927	228	53926	237.65625	840	237.68750	1294	237.671874	0	360.09375	675	...	13064
1781928	228	53927	237.65625	586	237.68750	1454	237.671874	-44	360.09375	672	...	1259(

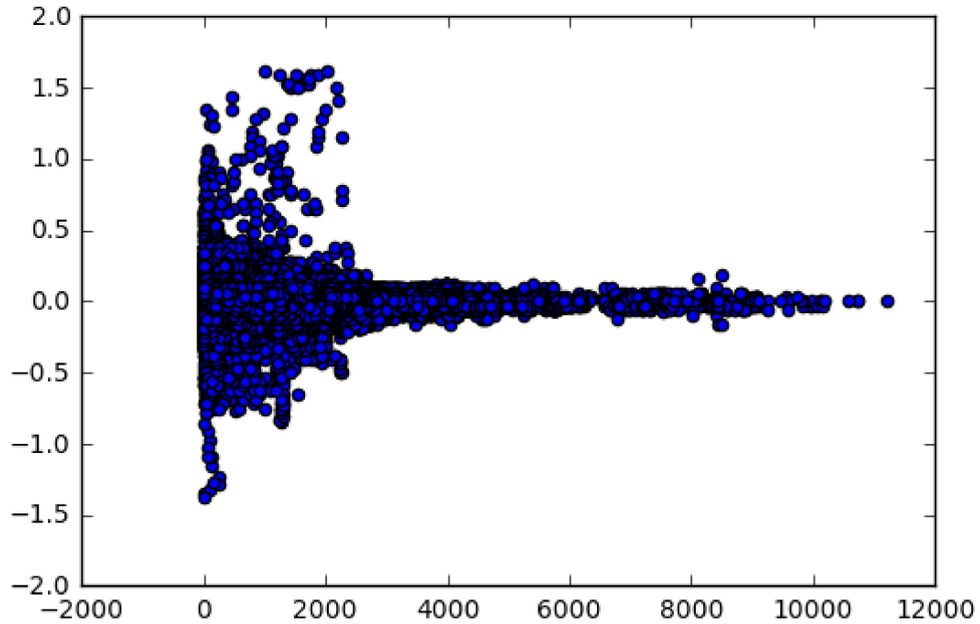
	date	time	bid_price_inst1	bid_size_inst1	ask_price_inst1	ask_size_inst1	mid_price_inst1	net_traded_inst1	bid_price_inst2	bid_size_inst2	...	pre_b
1781929	228	53928	237.65625	56	237.68750	1326	237.671874	-170	360.09375	357	...	1248(
1781930	228	53929	237.62500	2296	237.65625	234	237.640624	-16	360.09375	225	...	1265(
1781931	228	53930	237.62500	1598	237.65625	282	237.640624	-12	360.00000	1197	...	1228(
1781932	228	53931	237.62500	1432	237.65625	624	237.640624	0	360.00000	1209	...	1433(
1781933	228	53932	237.62500	1536	237.65625	512	237.640624	0	360.00000	1227	...	1528(
1781934	228	53933	237.62500	858	237.65625	1012	237.640624	6	360.00000	1074	...	1536(
1781935	228	53934	237.62500	874	237.65625	808	237.640624	0	360.00000	1122	...	1442(
1781936	228	53935	237.62500	1974	237.65625	292	237.640624	0	360.09375	12	...	1451(
1781937	228	53936	237.62500	1932	237.65625	320	237.640624	2	360.09375	12	...	1424(
1781938	228	53937	237.62500	1932	237.65625	324	237.640624	2	360.09375	12	...	1436(
1781939	228	53938	237.62500	1912	237.65625	324	237.640624	0	360.09375	12	...	1436(
1781940	228	53939	237.62500	1912	237.65625	324	237.640624	0	360.09375	12	...	1422(
1781941	228	53940	237.62500	1956	237.65625	304	237.640624	0	360.09375	12	...	1368(

1781941 rows × 42 columns

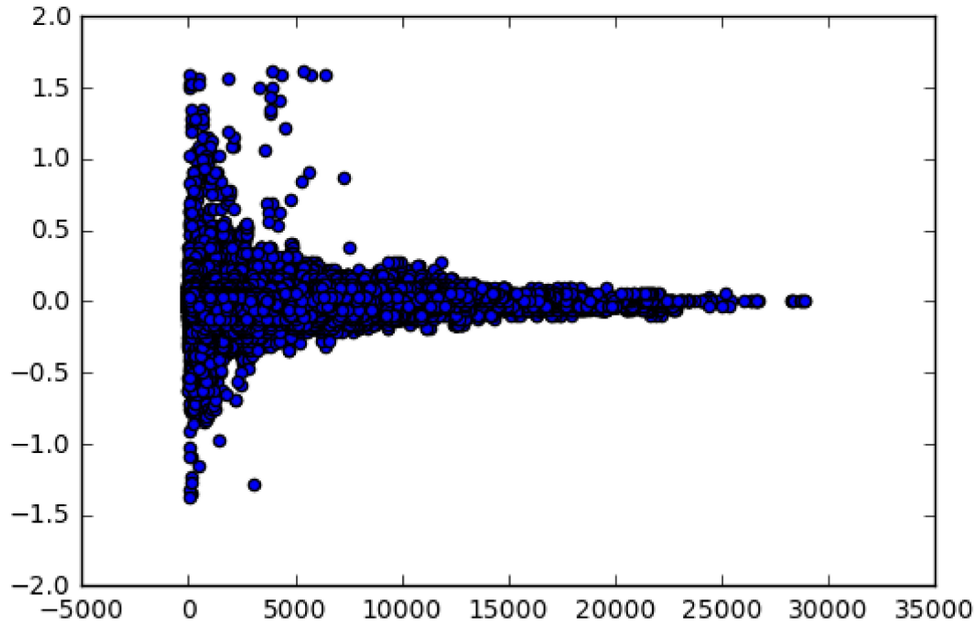
In [31]:
```
# inst _size v.s. y_value
import matplotlib.pyplot as plt
plt.scatter(df.pre_bid_size_inst1, df.y_value)
plt.show()
```



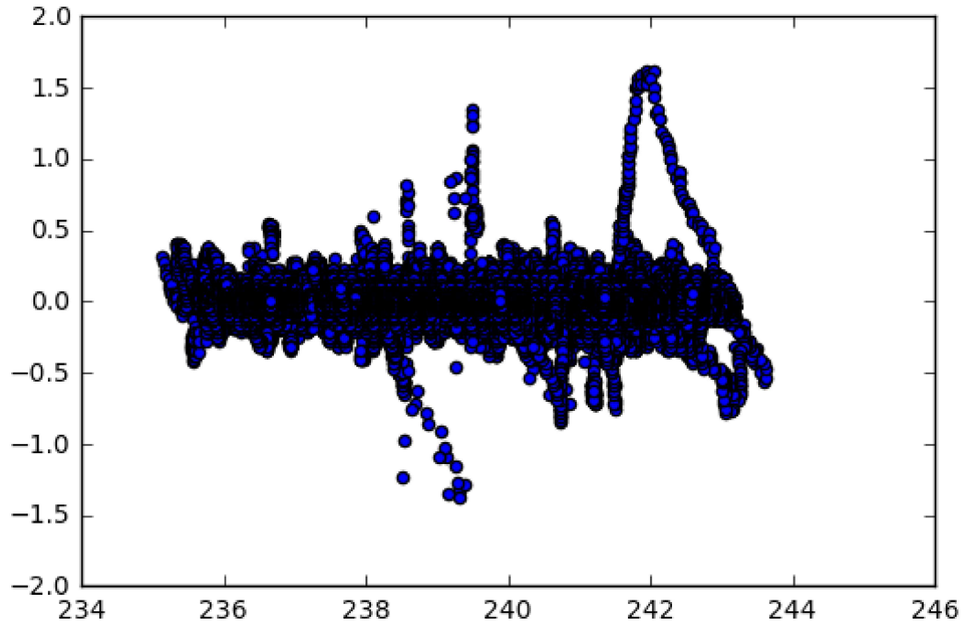
In [32]:
```
# size_inst2 vs y_value
plt.scatter(df.pre_bid_size_inst2, df.y_value)
plt.show()
```



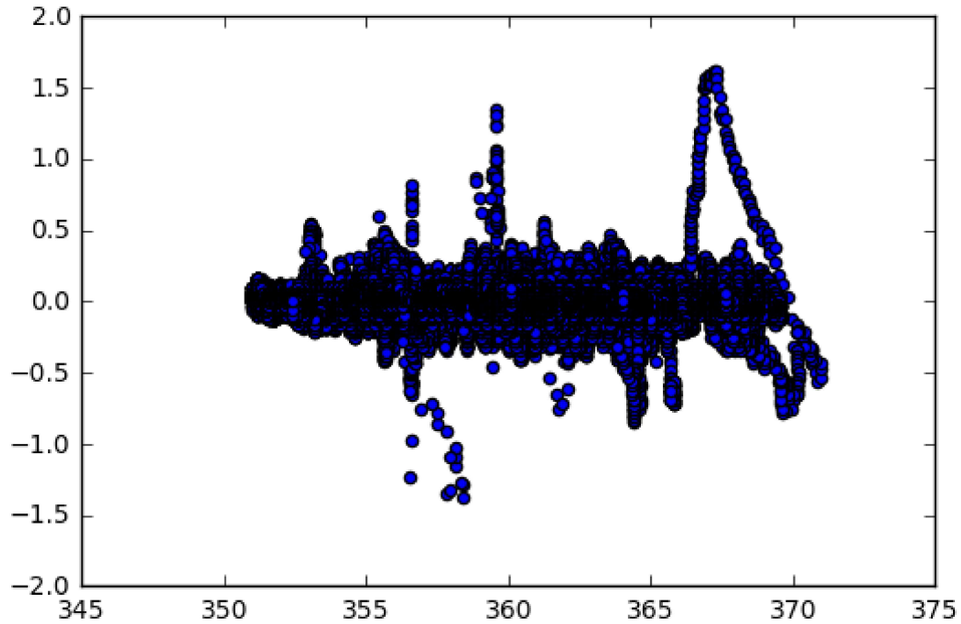
In [33]:
```
# size3 vs y_value
plt.scatter(df.pre_bid_size_inst3, df.y_value)
plt.show()
```



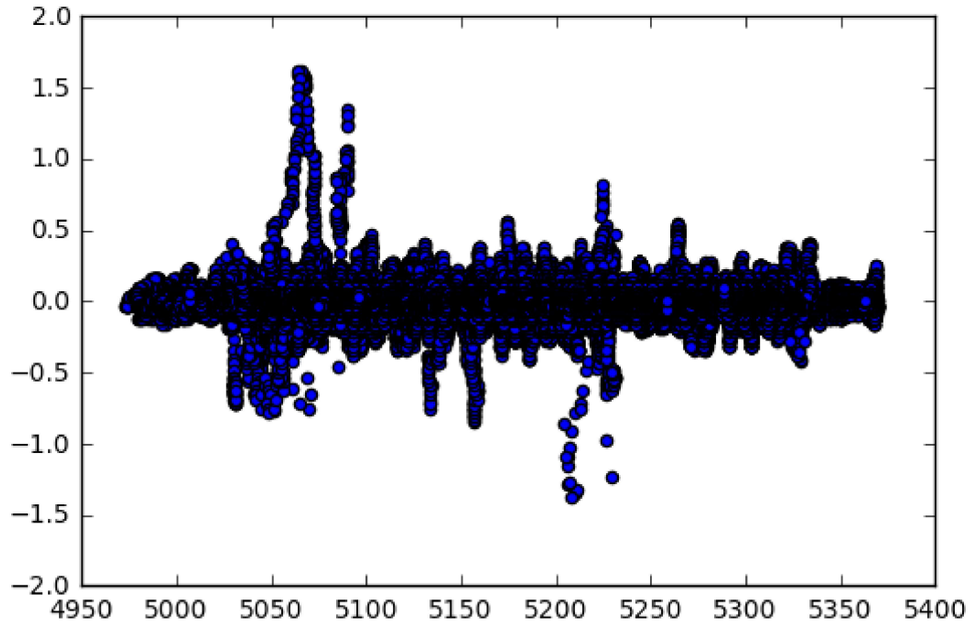
In [34]:
```
# price_inst1 vs y_value
plt.scatter(df.pre_bid_price_inst1, df.y_value)
plt.show()
```



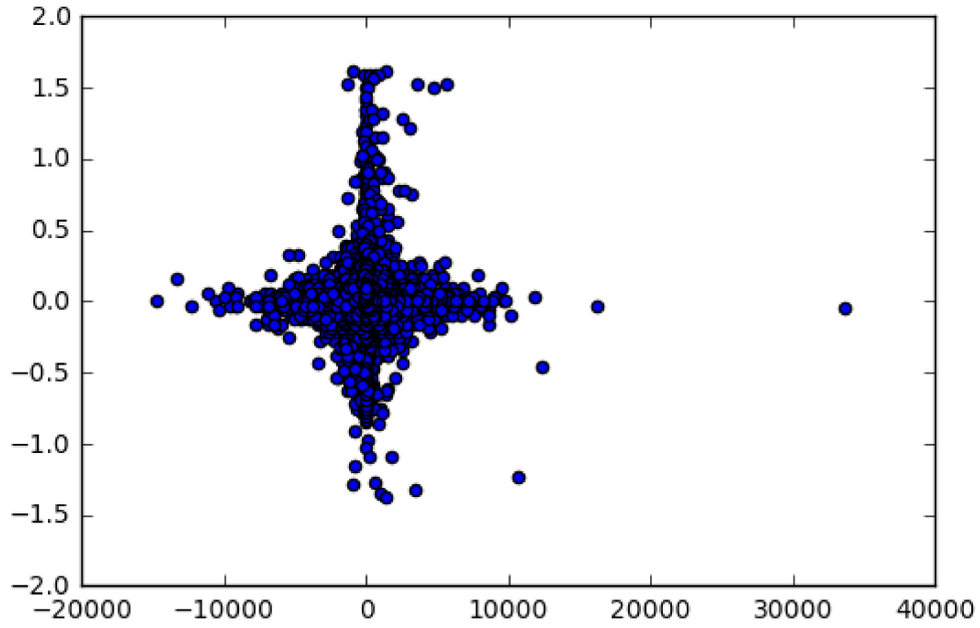
In [35]:
```
# price_inst2 vs y_value
plt.scatter(df.pre_bid_price_inst2, df.y_value)
plt.show()
```



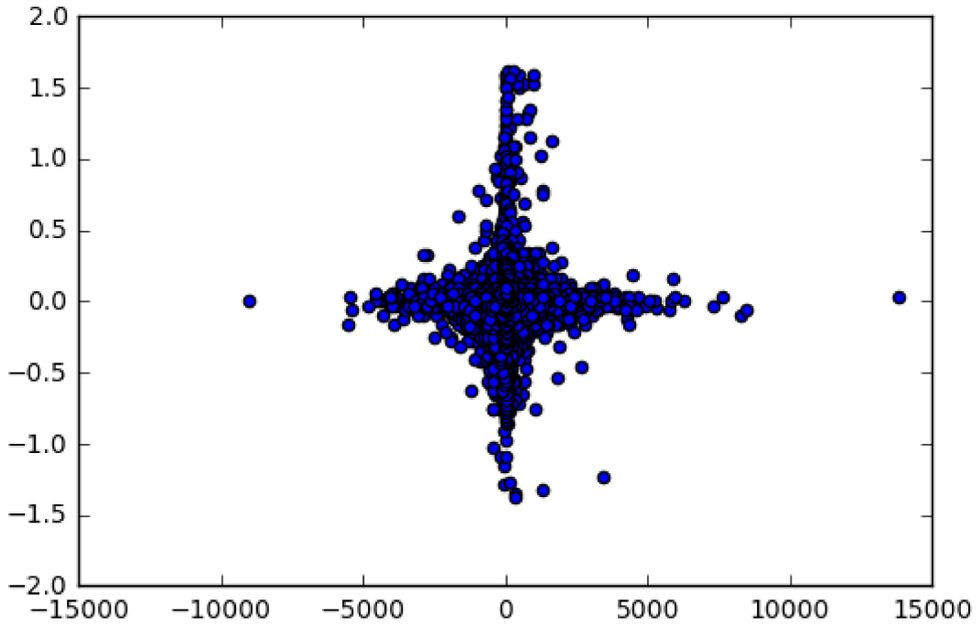
In [36]:
```
# price_inst3 vs y_value
plt.scatter(df.pre_bid_price_inst3, df.y_value)
plt.show()
```



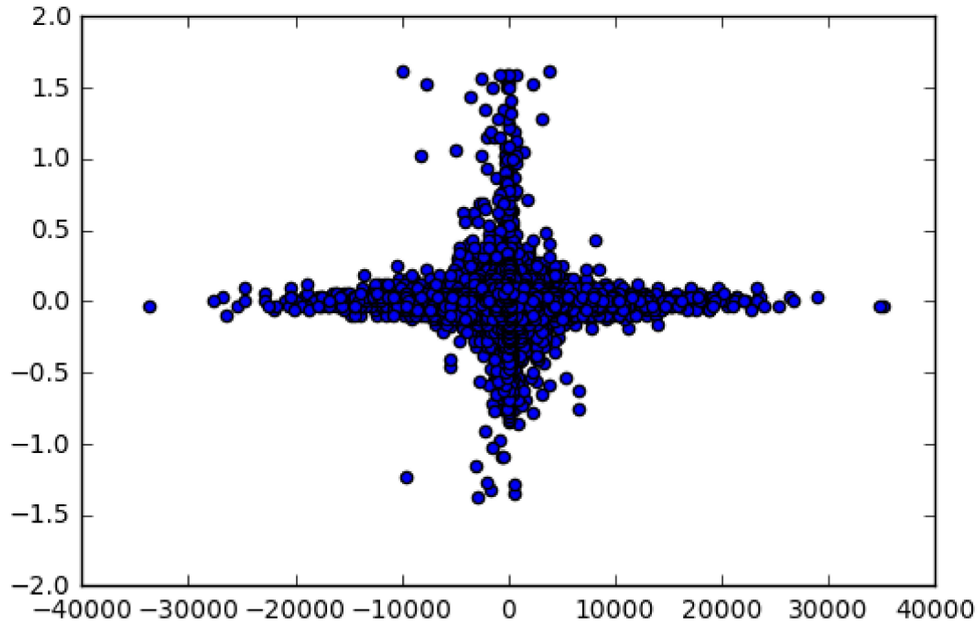
In [37]:
```
# net_traded_inst1 vs y_value
plt.scatter(df.net_traded_inst1, df.y_value)
plt.show()
```



In [38]:
```
# net_traded_inst2 vs y_value
plt.scatter(df.net_traded_inst2, df.y_value)
plt.show()
```



In [39]:
```
# net_traded_inst3 vs y_value
plt.scatter(df.net_traded_inst3, df.y_value)
plt.show()
```



In [27]:
```
df['size_diff1'] = df['pre_bid_size_inst1'] - df['pre_ask_size_inst1']
df['size_diff2'] = df['pre_bid_size_inst2'] - df['pre_ask_size_inst2']
df['size_diff3'] = df['pre_bid_size_inst3'] - df['pre_ask_size_inst3']
```

In [28]:
```
df['price_diff1']=df['pre_bid_price_inst1'] - df['pre_ask_price_inst1']
df['price_diff2']=df['pre_bid_price_inst2'] - df['pre_ask_price_inst2']
df['price_diff3']=df['pre_bid_price_inst3'] - df['pre_ask_price_inst3']
```

In [ ]:
```
```

In [63]:
```
import statsmodels.formula.api as sm
model1 = sm.ols(formula = 'y_value ~ pre_bid_price_inst1+pre_bid_price_inst2+pre_bid_price_inst3', data = df)
fitted = model1.fit()
print(fitted.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                y_value   R-squared:                       0.000
Model:                            OLS   Adj. R-squared:                  0.000
Method:                 Least Squares   F-statistic:                     74.15
Date:                Sun, 16 Apr 2017   Prob (F-statistic):           6.00e-48
Time:                        11:58:38   Log-Likelihood:             3.1814e+06
No. Observations:             1781941   AIC:                         -6.363e+06
Df Residuals:                 1781937   BIC:                         -6.363e+06
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================================
                             coef    std err          t      P>|t|      [95.0% Conf. Int.]
----------------------------------------------------------------------------------------------
Intercept                 0.0802      0.010      8.286      0.000          0.061      0.099
pre_bid_price_inst1      -0.0001   4.22e-05     -3.523      0.000         -0.000 -6.59e-05
pre_bid_price_inst2 -9.355e-05   1.68e-05     -5.553      0.000         -0.000 -6.05e-05
pre_bid_price_inst3  -2.09e-06   6.02e-07     -3.469      0.001      -3.27e-06 -9.09e-07
==============================================================================
Omnibus:                   720512.971   Durbin-Watson:                   0.072
Prob(Omnibus):                  0.000   Jarque-Bera (JB):        253238342.935
Skew:                           0.657   Prob(JB):                         0.00
Kurtosis:                      61.387   Cond. No.                     1.66e+06
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.66e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [51]:
```
import statsmodels.formula.api as sm
model3 = sm.ols(formula = 'y_value ~size_diff1+size_diff2 +size_diff3 ', data = df)
fitted = model3.fit()
print(fitted.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                y_value   R-squared:                       0.011
Model:                            OLS   Adj. R-squared:                  0.011
Method:                 Least Squares   F-statistic:                     6765.
Date:                Sun, 16 Apr 2017   Prob (F-statistic):               0.00
Time:                        10:59:47   Log-Likelihood:            3.1914e+06
No. Observations:             1781941   AIC:                        -6.383e+06
Df Residuals:                 1781937   BIC:                        -6.383e+06
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
Intercept   8.404e-05   3.02e-05      2.779      0.005      2.48e-05     0.000
size_diff1  2.822e-06   1.98e-08    142.402      0.000      2.78e-06  2.86e-06
size_diff2 -2.421e-07   2.83e-08     -8.554      0.000     -2.98e-07 -1.87e-07
size_diff3  6.161e-08    1.1e-08      5.584      0.000         4e-08  8.32e-08
==============================================================================
Omnibus:                   728251.145   Durbin-Watson:                   0.082
Prob(Omnibus):                  0.000   Jarque-Bera (JB):        265525240.549
Skew:                           0.671   Prob(JB):                         0.00
Kurtosis:                      62.786   Cond. No.                     2.74e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.74e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [54]:
```
import statsmodels.formula.api as sm
model3 = sm.ols(formula = 'y_value ~size_diff1*size_diff2 *size_diff3', data = df)
fitted = model3.fit()
print(fitted.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                y_value   R-squared:                       0.011
Model:                            OLS   Adj. R-squared:                  0.011
Method:                 Least Squares   F-statistic:                     2907.
Date:                Sun, 16 Apr 2017   Prob (F-statistic):               0.00
Time:                        11:01:58   Log-Likelihood:            3.1914e+06
No. Observations:             1781941   AIC:                        -6.383e+06
Df Residuals:                 1781933   BIC:                        -6.383e+06
Df Model:                           7
Covariance Type:            nonrobust
================================================================================================
                                   coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------------------------
Intercept                     8.487e-05   3.03e-05      2.804      0.005      2.55e-05     0.000
size_diff1                    2.818e-06   1.98e-08    142.177      0.000      2.78e-06  2.86e-06
size_diff2                   -2.468e-07   2.83e-08     -8.716      0.000     -3.02e-07 -1.91e-07
size_diff1:size_diff2        -5.568e-11   1.42e-11     -3.931      0.000     -8.34e-11 -2.79e-11
size_diff3                    6.197e-08    1.1e-08      5.614      0.000      4.03e-08  8.36e-08
size_diff1:size_diff3        -1.699e-11   7.16e-12     -2.372      0.018      -3.1e-11 -2.95e-12
size_diff2:size_diff3        -3.256e-11   1.03e-11     -3.171      0.002     -5.27e-11 -1.24e-11
size_diff1:size_diff2:size_diff3 -2.568e-14   5.06e-15     -5.076      0.000     -3.56e-14 -1.58e-14
==============================================================================
Omnibus:                   728236.412   Durbin-Watson:                   0.082
Prob(Omnibus):                  0.000   Jarque-Bera (JB):        265541193.973
Skew:                           0.671   Prob(JB):                         0.00
Kurtosis:                      62.788   Cond. No.                     6.00e+09
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large,  6e+09. This might indicate that there are
strong multicollinearity or other numerical problems.
```

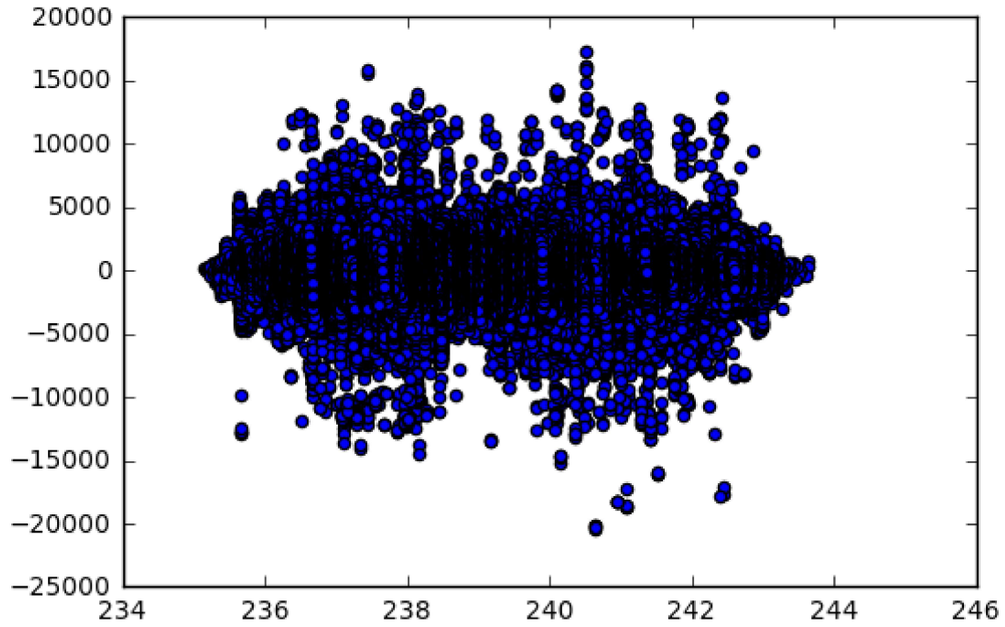
In [64]:
```
import statsmodels.formula.api as sm
model3 = sm.ols(formula = 'y_value ~price_diff1*size_diff1+price_diff2*size_diff2+price_diff3*size_diff3', data = df)
fitted = model3.fit()
print(fitted.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                y_value   R-squared:                       0.011
Model:                            OLS   Adj. R-squared:                  0.011
Method:                 Least Squares   F-statistic:                     2289.
Date:                Sun, 16 Apr 2017   Prob (F-statistic):               0.00
Time:                        12:10:30   Log-Likelihood:            3.1916e+06
No. Observations:             1781941   AIC:                        -6.383e+06
Df Residuals:                 1781931   BIC:                        -6.383e+06
Df Model:                           9
Covariance Type:            nonrobust
================================================================================================
                                   coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------------------------
Intercept                      0.0049      0.001      3.342      0.001       0.002      0.008
price_diff1                    0.2086      0.023      9.098      0.000       0.164      0.254
size_diff1                    3.1e-06   7.62e-07      4.067      0.000     1.61e-06   4.59e-06
price_diff1:size_diff1        8.91e-06   2.44e-05      0.366      0.715    -3.88e-05   5.67e-05
price_diff2                   -0.0717      0.005    -14.084      0.000      -0.082     -0.062
size_diff2                    2.6e-06    1.1e-06      2.374      0.018     4.53e-07   4.75e-06
price_diff2:size_diff2        3.03e-05   1.17e-05      2.596      0.009     7.42e-06   5.32e-05
price_diff3                    0.0050      0.001      4.138      0.000       0.003      0.007
size_diff3                -2.077e-06   5.85e-07     -3.549      0.000    -3.22e-06  -9.3e-07
price_diff3:size_diff3    -2.138e-06   5.85e-07     -3.655      0.000    -3.28e-06 -9.91e-07
==============================================================================
Omnibus:                   725733.840   Durbin-Watson:                   0.082
Prob(Omnibus):                  0.000   Jarque-Bera (JB):        264278706.726
Skew:                           0.664   Prob(JB):                         0.00
Kurtosis:                      62.646   Cond. No.                     2.94e+06
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.94e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
```

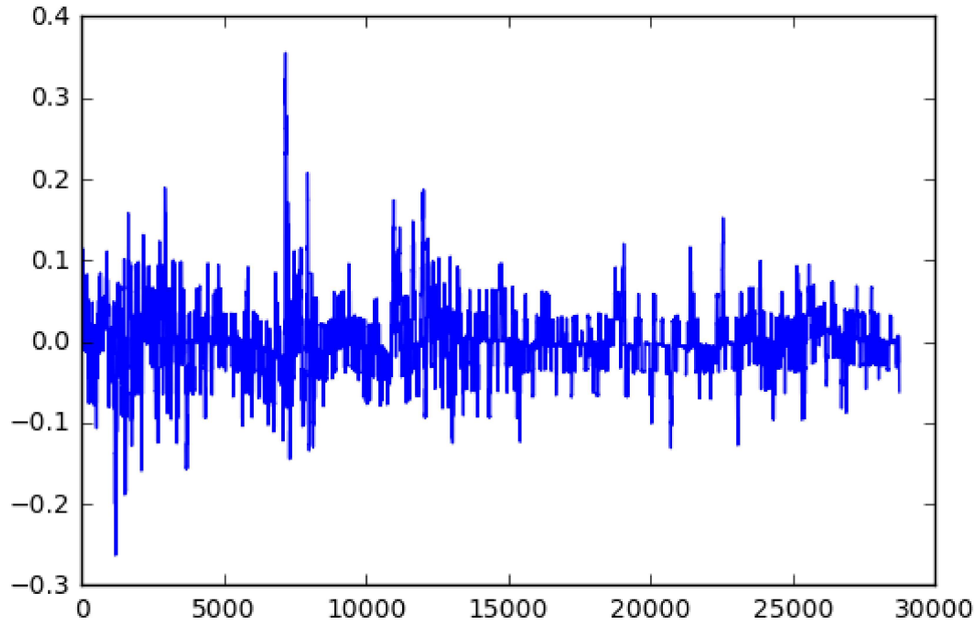
In [53]:
```
# Good model with maximum R_squared

import statsmodels.formula.api as sm
model3 = sm.ols(formula = 'y_value ~size_diff1+size_diff2 +size_diff3', data = df)
fitted = model3.fit()
print(fitted.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                y_value   R-squared:                       0.011
Model:                            OLS   Adj. R-squared:                  0.011
Method:                 Least Squares   F-statistic:                     6765.
Date:                Sun, 16 Apr 2017   Prob (F-statistic):               0.00
Time:                        11:01:36   Log-Likelihood:            3.1914e+06
No. Observations:             1781941   AIC:                        -6.383e+06
Df Residuals:                 1781937   BIC:                        -6.383e+06
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
Intercept   8.404e-05   3.02e-05      2.779      0.005      2.48e-05     0.000
size_diff1  2.822e-06   1.98e-08    142.402      0.000      2.78e-06  2.86e-06
size_diff2 -2.421e-07   2.83e-08     -8.554      0.000     -2.98e-07 -1.87e-07
size_diff3  6.161e-08    1.1e-08      5.584      0.000          4e-08  8.32e-08
==============================================================================
Omnibus:                   728251.145   Durbin-Watson:                   0.082
Prob(Omnibus):                  0.000   Jarque-Bera (JB):        265525240.549
Skew:                           0.671   Prob(JB):                         0.00
Kurtosis:                      62.786   Cond. No.                     2.74e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.74e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [61]:
```
plt.scatter(df.pre_mid_price_inst1, df.size_diff1)
plt.show()
```



In [232]:
```
plt.plot(range(1,28741), fitted.resid)
plt.show()


# Consideration procedure by implementing some other regress method.




# import matplotlib.pyplot as plt
# plt.plot(df_day0.pre_bid_size_inst2, df_day0.y_value)

# import math
# y = math.log(df_day0.y_value)
# plt.plot(df_day0.pre_bid_size_inst1 - df_day0.pre_ask_size_inst1, df_day0.y_value)

# plt.scatter((df_day0.pre_bid_price_inst1 - df_day0.pre_ask_price_inst1 + df_day0.pre_bid_price_inst2 - df_day0.pre_ask_price_inst2 + df_day0.pre_bid_price_inst3
 - df_day0.pre_ask_price_inst3)/3, df_day0.y_value)


# from sklearn.linear_model import LogisticRegression
# import statsmodels.formula.api as sm
# # model1 = sm.ols(formula = 'y_value ~ pre_mid_price_inst1 + pre_mid_price_inst2 + pre_mid_price_inst3', data = df_day0)
# fitted = model1.fit()
# print(fitted.summary())

# import statsmodels.formula.api as sm

# model1 = sm.GLM(formula = 'y_value ~ pre_mid_price_inst1 + pre_mid_price_inst2 + pre_mid_price_inst3', data = df_day0,family = sm.families.Gamma(sm.families.lin
ks.log))
# fitted = model1.fit()
# print(fitted.summary())


# # Load modules and data
# import statsmodels.api as sm
# data = sm.datasets.scotland.load()
# data.exog = sm.add_constant(data.exog)

# # Instantiate a gamma family model with the default link function.
# gamma_model = sm.GLM(data.endog, data.exog, family=sm.families.Gamma())
# gamma_results = gamma_model.fit()
```



In [ ]:
```
# df_day0.to_csv('sunrise.csv')
```