In [708]:
```
%load_ext sql
```

```
The sql extension is already loaded. To reload it, use:
  %reload_ext sql
```

In [709]:
```
%sql mysql://prod:nerd@52.2.153.189/rental_nerd
```

Out[709]:
```
u'Connected: prod@rental_nerd'
```

In [762]:
```
result = %sql (SELECT \
properties.id as "property_id", \
property_transaction_logs.id as "transaction_log_id", \
properties.*, \
property_transaction_logs.* \
FROM \
properties, \
property_transactions, \
property_transaction_logs \
WHERE \
properties.id = property_transactions.property_id AND \
property_transactions.property_transaction_log_id = property_transaction_logs.id AND \
property_transactions.transaction_type = 'rental')

data = result.DataFrame()
```

```
392 rows affected.
```

In [763]:
```
result.csv(filename="SQLdump.csv")
```

Out[763]: CSV results (./files/SQLdump.csv)

In [712]:
```
# imports
import pandas as pd
import matplotlib.pyplot as plt
# follow the usual sklearn pattern: import, instantiate, fit
from sklearn.linear_model import LinearRegression
import numpy as np

# this allows plots to appear directly in the notebook
%matplotlib inline

# read data into a DataFrame
data.head()
```

Out[712]:

	property_id	transaction_log_id	id	address	neighborhood	bedrooms	bathrooms	sqft	source	origin_url
0	1	1	1	567 Vallejo Street #PH500	San Francisco (North Beach)	3	3	2081	climbsf_renting	http://www.clir rent/567-vallejc
1	2	2	2	252 Granada Avenue	San Francisco (Ingleside)	2	2	1600	climbsf_renting	http://www.clir rent/252-grana
2	3	3	3	460 Valley Street	San Francisco (Noe Valley)	2	2	1446	climbsf_renting	http://www.clir rent/460-valley
3	4	4	4	333 Fremont Street #705	San Francisco (South Beach)	1	1	0	climbsf_renting	http://www.clir rent/333-fremo
4	5	5	5	420 Mission Bay Boulevard North #121	San Francisco (Mission Bay)	1	1	980	climbsf_renting	http://www.clir rent/420-missic

5 rows × 26 columns

In [713]:
```
import datetime

Date_final = [0.1] * len(data)

for x in range(0,len(data)):
    data
    if data["date_rented"][x] is not None :
        # print " row: "+ `x` + ": using date_rented"
        # data.ix['Date_final',x]
        Date_final[x] = data["date_rented"][x]

    elif data["date_listed"][x] is not None :
        # print " row: "+ `x` + ": using date_listed"
        Date_final[x] = data["date_listed"][x]
    else:
        print " row: "+ `x` + ": we are screwed"



data['Date'] = pd.to_datetime(Date_final)

data.head()
```

Out[713]:

	property_id	transaction_log_id	id	address	neighborhood	bedrooms	bathrooms	sqft	source	origin_url
0	1	1	1	567 Vallejo Street #PH500	San Francisco (North Beach)	3	3	2081	climbsf_renting	http://www.clir rent/567-vallejo
1	2	2	2	252 Granada Avenue	San Francisco (Ingleside)	2	2	1600	climbsf_renting	http://www.clir rent/252-grana
2	3	3	3	460 Valley Street	San Francisco (Noe Valley)	2	2	1446	climbsf_renting	http://www.clir rent/460-valley
3	4	4	4	333 Fremont Street #705	San Francisco (South Beach)	1	1	0	climbsf_renting	http://www.clir rent/333-fremo
4	5	5	5	420 Mission Bay Boulevard North #121	San Francisco (Mission Bay)	1	1	980	climbsf_renting	http://www.clir rent/420-missio

5 rows × 27 columns

In [714]:
```
# create year dummy variables (because date isn't very intuitive variable)
data["Year"] = pd.DatetimeIndex(data["Date"]).to_period('Y')

# create dummy variables using get_dummies, then exclude the first dummy column
year_dummies = pd.get_dummies(data.Year, prefix='Year').iloc[:, :-1]

# print out baseline neighborhood
base_area = pd.get_dummies(data.neighborhood, prefix='neighborhood').iloc[:, 0:1].columns[0]
print('Base neighborhood: %s' % base_area)

# create dummy variables using get_dummies, then exclude the first dummy column
area_dummies = pd.get_dummies(data.neighborhood, prefix='neighborhood').iloc[:, 1:]

# concatenate the dummy variable columns onto the original DataFrame (axis=0 means rows, axis=1 me
ans columns)
data = pd.concat([data, area_dummies, year_dummies], axis=1)

data.head()
```

```
Base neighborhood: neighborhood_East Bay (Berkeley)
```

Out[714]:

	property_id	transaction_log_id	id	address	neighborhood	bedrooms	bathrooms	sqft	source	origin_url
0	1	1	1	567 Vallejo Street #PH500	San Francisco (North Beach)	3	3	2081	climbsf_renting	http://www.clir rent/567-vallejo
1	2	2	2	252 Granada Avenue	San Francisco (Ingleside)	2	2	1600	climbsf_renting	http://www.clir rent/252-grana
2	3	3	3	460 Valley Street	San Francisco (Noe Valley)	2	2	1446	climbsf_renting	http://www.clir rent/460-valley
3	4	4	4	333 Fremont Street #705	San Francisco (South Beach)	1	1	0	climbsf_renting	http://www.clir rent/333-fremo
4	5	5	5	420 Mission Bay Boulevard North #121	San Francisco (Mission Bay)	1	1	980	climbsf_renting	http://www.clir rent/420-missi

5 rows × 91 columns

In [715]:
```
# filter out any outliers, defined as rent >$10k or >2,500 sq ft

data = data[(data.sqft <= 2500) & (data.price <= 8000) & (data.price != 0) & (data.bedrooms <= 4)
& (data.bathrooms <= 3) & (data.sqft != 0)]

# filter out listings over one month old
```

In [716]:
```
# FACTORING BY YEAR AND NEIGHBORHOOD
# Thesis: Neighborhoods influence valuations as a multiplier, rather than a constant.
# a square foot in SOMA is worth more than a square foot in Portrero by X%
# New model will look like this:
#      Price = B_1 x (SOMA Coeff * Year Coeff * Sqft) + intercept
#      $3,900 = B_1 x (1.20% * 1.15% * 2,023 sqft) + intercept
# where B_1 represents the price per square foot in base year and base neighborhood
# I will ignore intercepts for now FIXME
# calculate the coefficients for the following matrix and save them for later regressions
#                  SOMA    Mission    Portrero    Intercept
#  Price/SQFT      $1.23    $0.59       $0.88        $_.__

# create Price per square foot

price_per_foot = data.price / data.sqft
price_per_foot.name = 'price_per_foot'
data = pd.concat([data, price_per_foot], axis=1)

data.head()
```

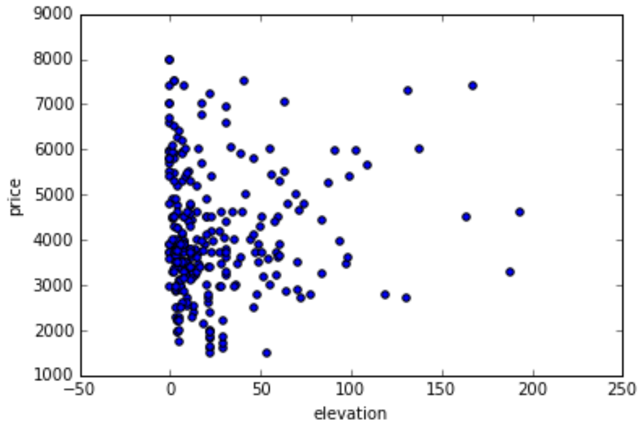
Out[716]:

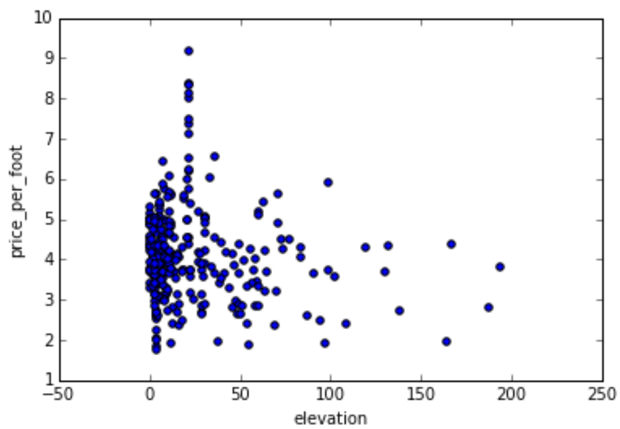
	property_id	transaction_log_id	id	address	neighborhood	bedrooms	bathrooms	sqft	source	origin_url
1	2	2	2	252 Granada Avenue	San Francisco (Ingleside)	2	2	1600	climbsf_renting	http://www.c rent/252-grar
2	3	3	3	460 Valley Street	San Francisco (Noe Valley)	2	2	1446	climbsf_renting	http://www.c rent/460-valle
4	5	5	5	420 Mission Bay Boulevard North #121	San Francisco (Mission Bay)	1	1	980	climbsf_renting	http://www.c rent/420-mis:
7	8	8	8	1160 Mission Street #1112	San Francisco (SOMA)	1	1	664	climbsf_renting	http://www.c rent/1160-mi:
11	12	12	12	655 26th Avenue	San Francisco (Central Richmond)	2	1	1300	climbsf_renting	http://www.c rent/655-26tl

5 rows × 92 columns

In [717]:
```
# visualize the relationship between the features and the response using scatterplots
data.plot(kind='scatter', x='elevation', y='price')
data.plot(kind='scatter', x='elevation', y='price_per_foot')
```

Out[717]: <matplotlib.axes._subplots.AxesSubplot at 0x10c0a0e10>





In [718]:
```
class ListTable(list):
    """ Overridden list class which takes a 2-dimensional list of
        the form [[1,2,3],[4,5,6]], and renders an HTML Table in
        IPython Notebook. """

    def _repr_html_(self):
        html = ["<table>"]
        for row in self:
            html.append("<tr>")

            for col in row:
                html.append("<td>{0}</td>".format(col))

            html.append("</tr>")
        html.append("</table>")
        return ''.join(html)
```

In [719]:
```
feature_cols = area_dummies.columns

X = data[feature_cols]
y = data.price_per_foot

# instantiate, fit
lm = LinearRegression()
lm.fit(X, y)

# print coefficients
# The mean square error
print("Residual sum of squares: %.2f"
      % np.mean((lm.predict(X) - y) ** 2))
# Explained variance score: 1 is perfect prediction
print('Variance score: %.2f' % lm.score(X, y))

# print raw results
print("Base area is %s: $%.2f" % (base_area, lm.intercept_))

zip(feature_cols,lm.coef_)

table = ListTable()

dtype = [('Neighborhood', 'S100'), ('$ per square', float)]

# round to pennies
round_coef = map(round,lm.coef_,[2]*len(lm.coef_))
x = np.array(zip(feature_cols, round_coef),dtype=dtype)
x.T
x = np.sort(x,axis=0,order='$ per square')

table.append(['Neighborhood','$ per square (+/-)'])
for i in x:
    table.append(i)

table
```

```
Residual sum of squares: 0.57
Variance score: 0.58
Base area is neighborhood_East Bay (Berkeley): $2.82
```

Out[719]:

Neighborhood	$ per square (+/-)
neighborhood_East Bay (Castro Valley)	-0.95
neighborhood_East Bay (Downtown Oakland)	-0.9
neighborhood_San Francisco (Visitacion Valley)	-0.88
neighborhood_Peninsula (Pacifica)	-0.4
neighborhood_San Francisco (Westwood Park)	-0.4
neighborhood_San Francisco (Ingleside)	-0.35
neighborhood_East Bay (West Oakland)	-0.34
neighborhood_East Bay (Walnut Creek)	-0.21
neighborhood_San Francisco (Bernal Heights)	-0.18
neighborhood_East Bay (Oakland)	-0.1
neighborhood_San Francisco (Glen Park)	-0.09
neighborhood_San Francisco (Portola)	-0.08
neighborhood_San Francisco (Diamond Heights)	-0.01

neighborhood_East Bay (Emeryville)	0.0
neighborhood_Other (Woodside)	-0.0
neighborhood_San Francisco (Clarendon Heights)	-0.0
neighborhood_San Francisco (Mission Bay/SoMa)	-0.0
neighborhood_San Francisco (Mission Terrace)	0.0
neighborhood_San Francisco (Candlestick Point)	0.1
neighborhood_San Francisco (Downtown San Francisco)	0.23
neighborhood_San Francisco (Central Richmond)	0.24
neighborhood_San Francisco (Bayview)	0.4
neighborhood_San Francisco (Excelsior)	0.4
neighborhood_San Francisco (Dogpatch)	0.44
neighborhood_San Francisco (Alamo Square)	0.47
neighborhood_San Francisco (Lone Mountain)	0.61
neighborhood_San Francisco (North Beach)	0.75
neighborhood_San Francisco (Van Ness-Civic Center)	0.76
neighborhood_San Francisco (Noe Valley)	0.79
neighborhood_San Francisco (Buena Vista)	0.88
neighborhood_San Francisco (Outer Richmond)	0.93
neighborhood_San Francisco (Golden Gate Heights)	1.01
neighborhood_San Francisco (Pacific Heights)	1.09
neighborhood_San Francisco (Western Addition)	1.11
neighborhood_San Francisco (Mission Bay)	1.19
neighborhood_San Francisco (North Panhandle)	1.22
neighborhood_San Francisco (SOMA)	1.29
neighborhood_San Francisco (Eureka Valley)	1.32
neighborhood_San Francisco (Potrero Hill)	1.33
neighborhood_San Francisco (Central Waterfront)	1.36
neighborhood_San Francisco (Eureka Valley-Dolores Heights)	1.44
neighborhood_San Francisco (Marina)	1.45
neighborhood_San Francisco (Cole Valley-Parnassus Heights)	1.49
neighborhood_San Francisco (Lower Pacific Heights)	1.54
neighborhood_San Francisco (Financial District)	1.55
neighborhood_San Francisco (South Beach)	1.56
neighborhood_San Francisco (Downtown)	1.62
neighborhood_San Francisco (Rincon Hill)	1.62
neighborhood_San Francisco (Duboce Triangle)	1.68
neighborhood_San Francisco (Russian Hill)	1.7
neighborhood_San Francisco (Inner Mission)	1.88
neighborhood_San Francisco (Nob Hill)	1.92

neighborhood_San Francisco (Yerba Buena)	2.06
neighborhood_San Francisco (South Financial District)	2.18
neighborhood_San Francisco (North Waterfront)	2.42
neighborhood_San Francisco (Mission Dolores)	2.57
neighborhood_San Francisco (Telegraph Hill)	2.73
neighborhood_San Francisco (Hayes Valley)	3.89

In [720]:

```
full_price = [lm.intercept_] * len(lm.coef_)
full_price += lm.coef_

area_price_per_foot = dict(zip(feature_cols,full_price))
area_price_per_foot[base_area] = lm.intercept_

dtype = [('Neighborhood', 'S100'), ('$ per sqft', float)]

# round to pennies
round_coef = map(round,full_price,[2]*len(full_price))
x = np.array(zip(feature_cols, full_price),dtype=dtype)
x.T
x = np.sort(x,axis=0,order='$ per sqft')

table = ListTable()

table.append(['Neighborhood','$ per sqft'])
for i in x:
    table.append(i)

table
```

Out[720]:

Neighborhood	$ per sqft
neighborhood_East Bay (Castro Valley)	1.875
neighborhood_East Bay (Downtown Oakland)	1.91666666667
neighborhood_San Francisco (Visitacion Valley)	1.93820224719
neighborhood_Peninsula (Pacifica)	2.416918429
neighborhood_San Francisco (Westwood Park)	2.41970021413
neighborhood_San Francisco (Ingleside)	2.46875
neighborhood_East Bay (West Oakland)	2.47789837974
neighborhood_East Bay (Walnut Creek)	2.61275829137
neighborhood_San Francisco (Bernal Heights)	2.64196428571
neighborhood_East Bay (Oakland)	2.72604154471
neighborhood_San Francisco (Glen Park)	2.72727272727
neighborhood_San Francisco (Portola)	2.73875140607
neighborhood_San Francisco (Diamond Heights)	2.8085106383
neighborhood_Other (Woodside)	2.82142857143
neighborhood_San Francisco (Clarendon Heights)	2.82142857143
neighborhood_San Francisco (Mission Bay/SoMa)	2.82142857143
neighborhood_San Francisco (Mission Terrace)	2.82142857143
neighborhood_East Bay (Emeryville)	2.82628422973

neighborhood_San Francisco (Candlestick Point)	2.92203694355
neighborhood_San Francisco (Downtown San Francisco)	3.05239029834
neighborhood_San Francisco (Central Richmond)	3.06552706553
neighborhood_San Francisco (Excelsior)	3.22222222222
neighborhood_San Francisco (Bayview)	3.22391991699
neighborhood_San Francisco (Dogpatch)	3.25768667643
neighborhood_San Francisco (Alamo Square)	3.28947368421
neighborhood_San Francisco (Lone Mountain)	3.43047619048
neighborhood_San Francisco (North Beach)	3.57638888889
neighborhood_San Francisco (Van Ness-Civic Center)	3.57814224333
neighborhood_San Francisco (Noe Valley)	3.60648920661
neighborhood_San Francisco (Buena Vista)	3.69863013699
neighborhood_San Francisco (Outer Richmond)	3.75
neighborhood_San Francisco (Golden Gate Heights)	3.83333333333
neighborhood_San Francisco (Pacific Heights)	3.91275451144
neighborhood_San Francisco (Western Addition)	3.92880386618
neighborhood_San Francisco (Mission Bay)	4.00986324215
neighborhood_San Francisco (North Panhandle)	4.04545454545
neighborhood_San Francisco (SOMA)	4.10829762675
neighborhood_San Francisco (Eureka Valley)	4.14285714286
neighborhood_San Francisco (Potrero Hill)	4.15169529928
neighborhood_San Francisco (Central Waterfront)	4.17900211482
neighborhood_San Francisco (Eureka Valley-Dolores Heights)	4.26485551486
neighborhood_San Francisco (Marina)	4.27103404056
neighborhood_San Francisco (Cole Valley-Parnassus Heights)	4.30769230769
neighborhood_San Francisco (Lower Pacific Heights)	4.36170254619
neighborhood_San Francisco (Financial District)	4.36781609195
neighborhood_San Francisco (South Beach)	4.38016977664
neighborhood_San Francisco (Downtown)	4.44444444444
neighborhood_San Francisco (Rincon Hill)	4.44537317129
neighborhood_San Francisco (Duboce Triangle)	4.5
neighborhood_San Francisco (Russian Hill)	4.52570564516
neighborhood_San Francisco (Inner Mission)	4.70259376879
neighborhood_San Francisco (Nob Hill)	4.73751043452
neighborhood_San Francisco (Yerba Buena)	4.87764732847
neighborhood_San Francisco (South Financial District)	5.0
neighborhood_San Francisco (North Waterfront)	5.24109014675
neighborhood_San Francisco (Mission Dolores)	5.38924963925
neighborhood_San Francisco (Telegraph Hill)	5.54711246201

neighborhood_San Francisco (Hayes Valley)	6.71633352328

In [721]:
```
# calculate the multipliers for each neighborhood relative to base area
# SOMA_mult = SOMA_per_foot / Base_per_foot

area_mults = [lm.intercept_] * len(lm.coef_)
area_mults = full_price / area_mults - [1]*len(lm.coef_)


dtype = [('Neighborhood', 'S100'), ('Multiplier', float)]

# round to pennies
round_coef = map(round,area_mults,[2]*len(area_mults))
x = np.array(zip(feature_cols, area_mults),dtype=dtype)
x.T
x = np.sort(x,axis=0,order='Multiplier')

table = ListTable()

table.append(['Neighborhood','Multiplier'])
table.append([base_area,0])
for i in x:
    table.append(i)

table
```

Out[721]:

Neighborhood	Multiplier
neighborhood_East Bay (Berkeley)	0
neighborhood_East Bay (Castro Valley)	-0.335443037975
neighborhood_East Bay (Downtown Oakland)	-0.320675105485
neighborhood_San Francisco (Visitacion Valley)	-0.313042241502
neighborhood_Peninsula (Pacifica)	-0.143370683391
neighborhood_San Francisco (Westwood Park)	-0.142384734231
neighborhood_San Francisco (Ingleside)	-0.125
neighborhood_East Bay (West Oakland)	-0.121757536295
neighborhood_East Bay (Walnut Creek)	-0.073959086603
neighborhood_San Francisco (Bernal Heights)	-0.0636075949367
neighborhood_East Bay (Oakland)	-0.0338080601037
neighborhood_San Francisco (Glen Park)	-0.0333716915995
neighborhood_San Francisco (Portola)	-0.0293032991129
neighborhood_San Francisco (Diamond Heights)	-0.00457850794506
neighborhood_Other (Woodside)	-6.32827124036e-15
neighborhood_San Francisco (Clarendon Heights)	0.0
neighborhood_San Francisco (Mission Bay/SoMa)	0.0
neighborhood_San Francisco (Mission Terrace)	2.22044604925e-16
neighborhood_East Bay (Emeryville)	0.00172099281444
neighborhood_San Francisco (Candlestick Point)	0.035658663535
neighborhood_San Francisco (Downtown San Francisco)	0.0818598525776
neighborhood_San Francisco (Central Richmond)	0.086515921959

neighborhood_San Francisco (Excelsior)	0.142053445851
neighborhood_San Francisco (Bayview)	0.142655160452
neighborhood_San Francisco (Dogpatch)	0.154623125822
neighborhood_San Francisco (Alamo Square)	0.165889407062
neighborhood_San Francisco (Lone Mountain)	0.215864978903
neighborhood_San Francisco (North Beach)	0.267580872011
neighborhood_San Francisco (Van Ness-Civic Center)	0.268202314093
neighborhood_San Francisco (Noe Valley)	0.27824933905
neighborhood_San Francisco (Buena Vista)	0.310906883995
neighborhood_San Francisco (Outer Richmond)	0.329113924051
neighborhood_San Francisco (Golden Gate Heights)	0.35864978903
neighborhood_San Francisco (Pacific Heights)	0.386799067347
neighborhood_San Francisco (Western Addition)	0.392487446241
neighborhood_San Francisco (Mission Bay)	0.421217351648
neighborhood_San Francisco (North Panhandle)	0.433831990794
neighborhood_San Francisco (SOMA)	0.456105487961
neighborhood_San Francisco (Eureka Valley)	0.46835443038
neighborhood_San Francisco (Potrero Hill)	0.471486941518
neighborhood_San Francisco (Central Waterfront)	0.481165306518
neighborhood_San Francisco (Eureka Valley-Dolores Heights)	0.511594359696
neighborhood_San Francisco (Marina)	0.513784216909
neighborhood_San Francisco (Cole Valley-Parnassus Heights)	0.526777020448
neighborhood_San Francisco (Lower Pacific Heights)	0.545919889788
neighborhood_San Francisco (Financial District)	0.548086716136
neighborhood_San Francisco (South Beach)	0.552465237289
neighborhood_San Francisco (Downtown)	0.575246132208
neighborhood_San Francisco (Rincon Hill)	0.575575301216
neighborhood_San Francisco (Duboce Triangle)	0.594936708861
neighborhood_San Francisco (Russian Hill)	0.604047570437
neighborhood_San Francisco (Inner Mission)	0.666742095268
neighborhood_San Francisco (Nob Hill)	0.679117622362
neighborhood_San Francisco (Yerba Buena)	0.728786394902
neighborhood_San Francisco (South Financial District)	0.772151898734
neighborhood_San Francisco (North Waterfront)	0.857601571
neighborhood_San Francisco (Mission Dolores)	0.91011379619
neighborhood_San Francisco (Telegraph Hill)	0.966065176407
neighborhood_San Francisco (Hayes Valley)	1.38047264116

In [722]:

```
# calculate the adjusted Sqft (Sqft * Area_mult) for the dataset and add it as a new column to dat
a

# for each property, multiplier is sum of array [area_dummies] x [area_mults]

t = data[area_dummies.columns] * area_mults
t = t.T.sum()

t.name = 'area_multiplier'
t = t + 1
data = pd.concat([data, t], axis=1)

adj_sqft = data.sqft * t
adj_sqft.name = 'area_adj_sqft'
data = pd.concat([data, adj_sqft], axis=1)

data.head()
```

Out[722]:

	property_id	transaction_log_id	id	address	neighborhood	bedrooms	bathrooms	sqft	source	origin_url
1	2	2	2	252 Granada Avenue	San Francisco (Ingleside)	2	2	1600	climbsf_renting	http://www.c rent/252-grar
2	3	3	3	460 Valley Street	San Francisco (Noe Valley)	2	2	1446	climbsf_renting	http://www.c rent/460-valle
4	5	5	5	420 Mission Bay Boulevard North #121	San Francisco (Mission Bay)	1	1	980	climbsf_renting	http://www.c rent/420-miss
7	8	8	8	1160 Mission Street #1112	San Francisco (SOMA)	1	1	664	climbsf_renting	http://www.c rent/1160-mi
11	12	12	12	655 26th Avenue	San Francisco (Central Richmond)	2	1	1300	climbsf_renting	http://www.c rent/655-26th

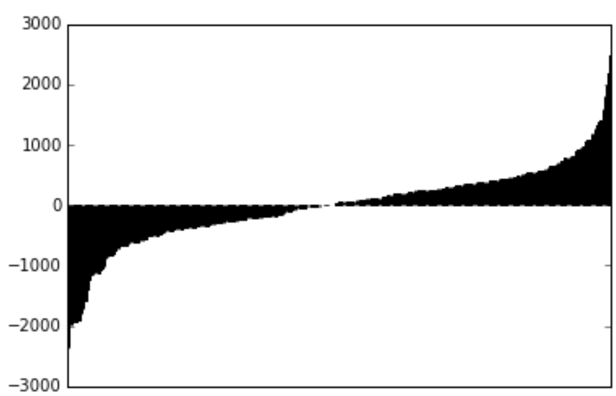
5 rows × 94 columns

In [723]:

```
# run the regression based on area_adj_sqft rather than sqft

# create X and y
feature_cols = [data.area_adj_sqft.name]

X = data[feature_cols]
y = data.price

# instantiate, fit
lm = LinearRegression()
lm.fit(X, y)

# print coefficients
print("Intercept: %.2f" % lm.intercept_)

# The mean square error
print("Residual sum of squares: %.2f"
      % np.mean((lm.predict(X) - y) ** 2))
# Explained variance score: 1 is perfect prediction
print('Variance score: %.2f' % lm.score(X, y))
zip(feature_cols, lm.coef_)

# calculate predictions for the data set and plot errors
predictions = lm.predict(X)
errors = predictions-y
errors.name = 'Error'

# visualize the relationship between the features and the response using scatterplots
errors.sort()
errors.plot(kind='bar').get_xaxis().set_ticks([])
```

```
Intercept: 1041.67
Residual sum of squares: 441256.17
Variance score: 0.76
```

Out[723]: []



In [724]:
```
feature_cols = year_dummies.columns

X = data[feature_cols]
y = data.price_per_foot

# instantiate, fit
lm = LinearRegression()
lm.fit(X, y)

# print coefficients
# The mean square error
print("Residual sum of squares: %.2f"
      % np.mean((lm.predict(X) - y) ** 2))
# Explained variance score: 1 is perfect prediction
print('Variance score: %.2f' % lm.score(X, y))

# print raw results
print lm.intercept_

zip(feature_cols,lm.coef_)
```

```
Residual sum of squares: 1.25
Variance score: 0.08
4.5007879875
```

Out[724]:
```
[(u'Year_1969', -1.6454580382583559),
 (u'Year_2011', -0.33022715359146831),
 (u'Year_2012', -1.074493166779801),
 (u'Year_2013', -0.75731724502172704),
 (u'Year_2014', -0.56961588108209849)]
```

In [725]:
```
full_price = [lm.intercept_] * len(lm.coef_)
full_price += lm.coef_

year_price_per_foot = dict(zip(feature_cols,full_price))
year_price_per_foot[base_area] = lm.intercept_

print year_price_per_foot
```

```
{u'Year_1969': 2.8553299492385769, u'Year_2012': 3.426294820717132, u'Year_2013': 3.74347074247520
61, u'Year_2011': 4.1705608339054647, 'neighborhood_East Bay (Berkeley)': 4.500787987496933, u'Yea
r_2014': 3.9311721064148344}
```

In [726]:
```
# calculate the multipliers for each year relative to base year
# 2014_mult = 2014_per_foot / 2015_per_foot

year_mults = [lm.intercept_] * len(lm.coef_)
year_mults = full_price / year_mults - [1]*len(lm.coef_)

zip(feature_cols, year_mults)
```

Out[726]:
```
[(u'Year_1969', -0.3655933233979014),
 (u'Year_2011', -0.07337096404203669),
 (u'Year_2012', -0.23873445489205758),
 (u'Year_2013', -0.16826325681759147),
 (u'Year_2014', -0.1265591453462096)]
```

In [727]:
```
# calculate the adjusted Sqft (Sqft * Year_mult) for the dataset and add it as a new column to dat
a

# for each property, multiplier is sum of array [year_dummies] x [year_mults]

t = data[year_dummies.columns] * year_mults
t = t.T.sum()

t.name = 'year_multiplier'
t = t + 1
data = pd.concat([data, t], axis=1)

year_adj_sqft = data.area_adj_sqft * t
year_adj_sqft.name = 'adj_sqft'
data = pd.concat([data, year_adj_sqft], axis=1)

data.head()
```

Out[727]:

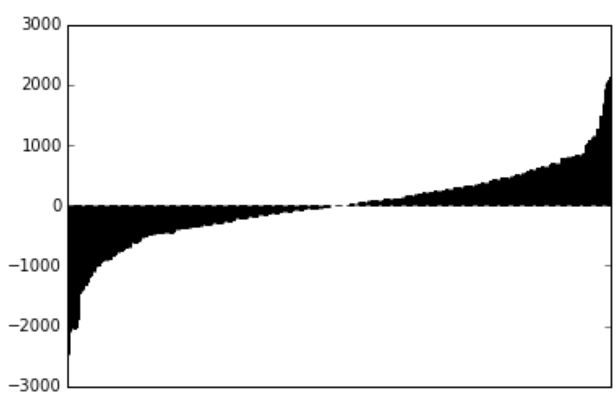
	property_id	transaction_log_id	id	address	neighborhood	bedrooms	bathrooms	sqft	source	origin_url
1	2	2	2	252 Granada Avenue	San Francisco (Ingleside)	2	2	1600	climbsf_renting	http://www.c rent/252-grar
2	3	3	3	460 Valley Street	San Francisco (Noe Valley)	2	2	1446	climbsf_renting	http://www.c rent/460-valle
4	5	5	5	420 Mission Bay Boulevard North #121	San Francisco (Mission Bay)	1	1	980	climbsf_renting	http://www.c rent/420-mis:
7	8	8	8	1160 Mission Street #1112	San Francisco (SOMA)	1	1	664	climbsf_renting	http://www.c rent/1160-mi:
11	12	12	12	655 26th Avenue	San Francisco (Central Richmond)	2	1	1300	climbsf_renting	http://www.c rent/655-26tl

5 rows × 96 columns

In [728]:
```
# run the regression based on year_and_area_adj_sqft rather than area_adj_sqft

# create X and y
feature_cols = ['adj_sqft']

X = data[feature_cols]
y = data.price

# instantiate, fit
lm = LinearRegression()
lm.fit(X, y)

# print coefficients
print lm.intercept_
# The mean square error
print("Residual sum of squares: %.2f"
      % np.mean((lm.predict(X) - y) ** 2))
# Explained variance score: 1 is perfect prediction
print('Variance score: %.2f' % lm.score(X, y))
print zip(feature_cols, lm.coef_)

# calculate predictions for the data set and plot errors
predictions = lm.predict(X)
errors = predictions-y
errors.name = 'Error'

# visualize the relationship between the features and the response using scatterplots
errors.sort(inplace=True)
errors.plot(kind='bar').get_xaxis().set_ticks([])

errors.tail(10)
```

```
941.521211083
Residual sum of squares: 419109.96
Variance score: 0.77
[('adj_sqft', 2.306505722057671)]
```

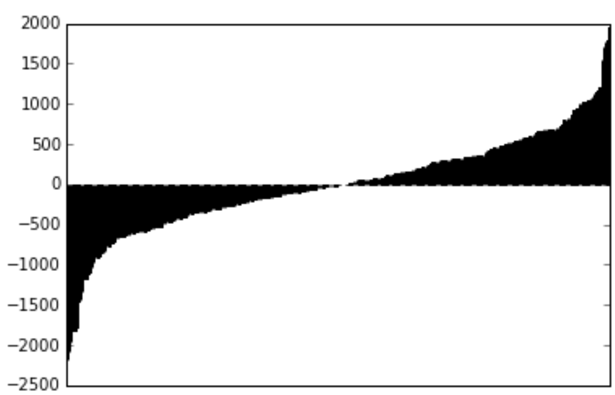
Out[728]:
```
34     1158.095914
181    1166.525374
73     1275.884300
320    1280.805609
326    1495.345136
105    1696.350944
328    1943.888012
236    2059.761250
60     2081.582673
66     2142.403403
Name: Error, dtype: float64
```



In [729]:
```
# create X and y
feature_cols = ['adj_sqft', 'bedrooms', 'bathrooms']

X = data[feature_cols]
y = data.price

# instantiate, fit
lm = LinearRegression()
lm.fit(X, y)

# print coefficients
print("Intercept: %.2f" % lm.intercept_)
# The mean square error
print("Residual sum of squares: %.2f"
      % np.mean((lm.predict(X) - y) ** 2))
# Explained variance score: 1 is perfect prediction
print('Variance score: %.2f' % lm.score(X, y))
print zip(feature_cols, lm.coef_)

# calculate predictions for the data set and plot errors
predictions = lm.predict(X)
errors = predictions-y
errors.name = 'Error'

# visualize the relationship between the features and the response using scatterplots
errors.sort()
errors.plot(kind='bar').get_xaxis().set_ticks([])
```

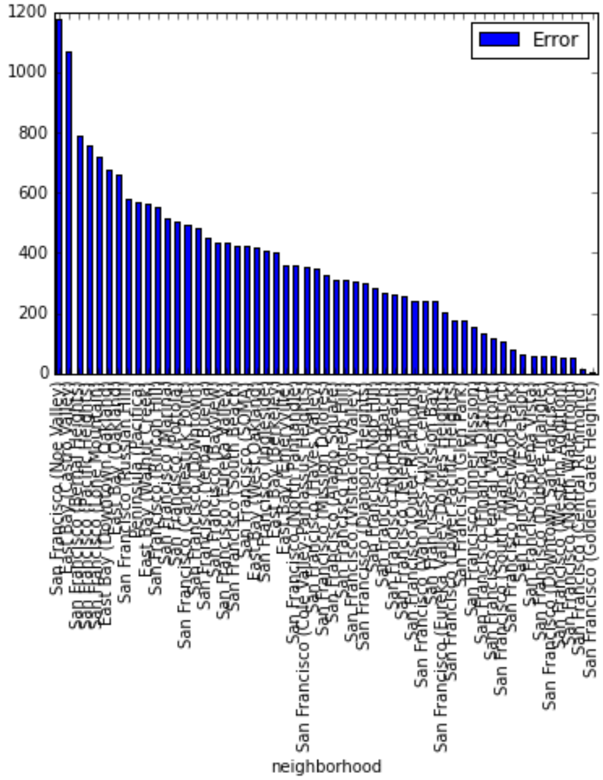
```
Intercept: 777.49
Residual sum of squares: 387000.91
Variance score: 0.79
[('adj_sqft', 2.0042956638310701), ('bedrooms', 163.98155450733458), ('bathrooms', 222.82601785404
927)]
```

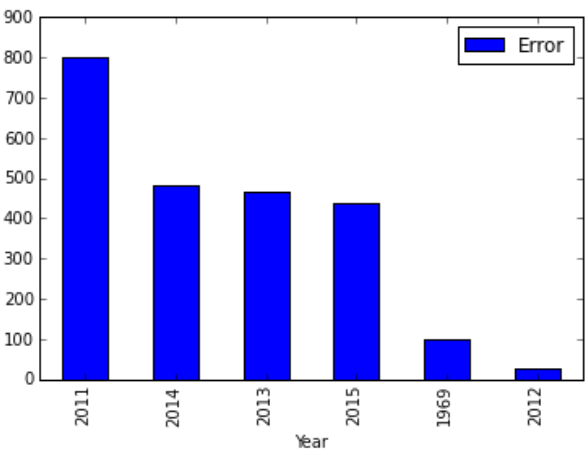
Out[729]: []

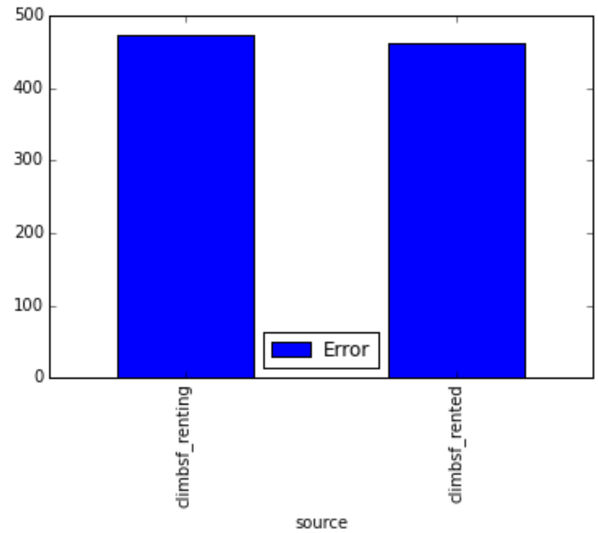


In [730]:
```
# show errors by neighborhood to see if there are any neighborhoods with funky differences

hooderrors = data[['neighborhood']]

errors = predictions-y
errors.name = 'Error'

hooderrors = pd.concat([hooderrors,errors.abs()],axis=1)

hood_group = hooderrors.groupby('neighborhood')

import numpy
def median(lst):
    return numpy.median(numpy.array(lst))

error_avg = hood_group.median()
error_avg.sort(columns='Error',ascending=False).plot(kind='bar')

# show errors by year to see if there are any years with funky differences

yearerrors = data[['Year']]

yearerrors = pd.concat([yearerrors,errors.abs()],axis=1)

year_group = yearerrors.groupby('Year')
error_avg = year_group.mean()
error_avg.sort(columns='Error',ascending=False).plot(kind='bar')

# show errors by source to see if there are any sources have noisy data

srcerrors = data[['source']]

srcerrors = pd.concat([srcerrors,errors.abs()],axis=1)

src_group = srcerrors.groupby('source')
error_avg = src_group.mean()
error_avg.sort(columns='Error',ascending=False).plot(kind='bar')
```

Out[730]: <matplotlib.axes._subplots.AxesSubplot at 0x10e6df9d0>


Error
1200
1000
800
600
400
200
0
San Francisco (Noe Valley)
San Francisco (Golden Gate Heights)
neighborhood


Error
900
800
700
600
500
400
300
200
100
0
2011
2014
2013
2015
1969
2012
Year



In [731]:

```
import csv

table = ListTable()

dtype = [('Effect', 'S100'), ('Coefficient', float)]

# round to pennies
round_coef = map(round,lm.coef_,[6]*len(lm.coef_))
x = np.array(zip(feature_cols, round_coef),dtype=dtype)
x.T
print zip(feature_cols, lm.coef_)
#x = np.sort(x,axis=0,order='Coefficient')

with open('model_features_v1.csv', 'wb') as csvfile:
    modelwriter = csv.writer(csvfile, delimiter=',', quotechar='|', quoting=csv.QUOTE_MINIMAL)

    header = ['Effect','Coefficient']
    table.append(header)
    modelwriter.writerow(header)
    for i in x:
        table.append(i)
        modelwriter.writerow(i)


    table.append(['base_rent', lm.intercept_])


    modelwriter.writerow(['base_rent',lm.intercept_])

table
```

```
[('adj_sqft', 2.0042956638310701), ('bedrooms', 163.98155450733458), ('bathrooms', 222.82601785404
927)]
```

Out[731]:

Effect	Coefficient
adj_sqft	2.004296
bedrooms	163.981555
bathrooms	222.826018
base_rent	777.494958254

In [732]:
```
table = ListTable()

dtype = [('Effect', 'S100'), ('Coefficient', float)]

# round to pennies
round_coef = map(round,(area_mults + [1]*len(area_mults)),[6]*len(area_mults))
x = np.array(zip(area_dummies.columns, round_coef),dtype=dtype)
x.T
x = np.sort(x,axis=0,order='Coefficient')

with open('model_hoods_v1.csv', 'wb') as csvfile:
    hoodwriter = csv.writer(csvfile, delimiter=',', quotechar='|', quoting=csv.QUOTE_MINIMAL)

    header = ['Neighborhood','Multiplier']
    table.append(header)
    hoodwriter.writerow(header)

    for i in x:
        table.append(i)
        hoodwriter.writerow(i)

    lastrow = [base_area, 1]
    table.append(lastrow)
    hoodwriter.writerow(lastrow)


table
```

Out[732]:

Neighborhood	Multiplier
neighborhood_East Bay (Castro Valley)	0.664557
neighborhood_East Bay (Downtown Oakland)	0.679325
neighborhood_San Francisco (Visitacion Valley)	0.686958
neighborhood_Peninsula (Pacifica)	0.856629
neighborhood_San Francisco (Westwood Park)	0.857615
neighborhood_San Francisco (Ingleside)	0.875
neighborhood_East Bay (West Oakland)	0.878242
neighborhood_East Bay (Walnut Creek)	0.926041
neighborhood_San Francisco (Bernal Heights)	0.936392
neighborhood_East Bay (Oakland)	0.966192
neighborhood_San Francisco (Glen Park)	0.966628
neighborhood_San Francisco (Portola)	0.970697
neighborhood_San Francisco (Diamond Heights)	0.995421
neighborhood_Other (Woodside)	1.0
neighborhood_San Francisco (Clarendon Heights)	1.0
neighborhood_San Francisco (Mission Bay/SoMa)	1.0
neighborhood_San Francisco (Mission Terrace)	1.0
neighborhood_East Bay (Emeryville)	1.001721
neighborhood_San Francisco (Candlestick Point)	1.035659
neighborhood_San Francisco (Downtown San Francisco)	1.08186
neighborhood_San Francisco (Central Richmond)	1.086516

neighborhood_San Francisco (Excelsior)	1.142053
neighborhood_San Francisco (Bayview)	1.142655
neighborhood_San Francisco (Dogpatch)	1.154623
neighborhood_San Francisco (Alamo Square)	1.165889
neighborhood_San Francisco (Lone Mountain)	1.215865
neighborhood_San Francisco (North Beach)	1.267581
neighborhood_San Francisco (Van Ness-Civic Center)	1.268202
neighborhood_San Francisco (Noe Valley)	1.278249
neighborhood_San Francisco (Buena Vista)	1.310907
neighborhood_San Francisco (Outer Richmond)	1.329114
neighborhood_San Francisco (Golden Gate Heights)	1.35865
neighborhood_San Francisco (Pacific Heights)	1.386799
neighborhood_San Francisco (Western Addition)	1.392487
neighborhood_San Francisco (Mission Bay)	1.421217
neighborhood_San Francisco (North Panhandle)	1.433832
neighborhood_San Francisco (SOMA)	1.456105
neighborhood_San Francisco (Eureka Valley)	1.468354
neighborhood_San Francisco (Potrero Hill)	1.471487
neighborhood_San Francisco (Central Waterfront)	1.481165
neighborhood_San Francisco (Eureka Valley-Dolores Heights)	1.511594
neighborhood_San Francisco (Marina)	1.513784
neighborhood_San Francisco (Cole Valley-Parnassus Heights)	1.526777
neighborhood_San Francisco (Lower Pacific Heights)	1.54592
neighborhood_San Francisco (Financial District)	1.548087
neighborhood_San Francisco (South Beach)	1.552465
neighborhood_San Francisco (Downtown)	1.575246
neighborhood_San Francisco (Rincon Hill)	1.575575
neighborhood_San Francisco (Duboce Triangle)	1.594937
neighborhood_San Francisco (Russian Hill)	1.604048
neighborhood_San Francisco (Inner Mission)	1.666742
neighborhood_San Francisco (Nob Hill)	1.679118
neighborhood_San Francisco (Yerba Buena)	1.728786
neighborhood_San Francisco (South Financial District)	1.772152
neighborhood_San Francisco (North Waterfront)	1.857602
neighborhood_San Francisco (Mission Dolores)	1.910114
neighborhood_San Francisco (Telegraph Hill)	1.966065
neighborhood_San Francisco (Hayes Valley)	2.380473
neighborhood_East Bay (Berkeley)	1

In [733]:
```
# show negative errors meaning we expected rents to be higher

error = predictions-y
error.name = 'error'

data = pd.concat([data,error,pd.DataFrame(predictions,columns=['predicted_price'])],axis=1)

data.head()
```

Out[733]:

	property_id	transaction_log_id	id	address	neighborhood	bedrooms	bathrooms	sqft	source	origin_url
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	2	2	2	252 Granada Avenue	San Francisco (Ingleside)	2	2	1600	climbsf_renting	http://www.c rent/252-gra
2	3	3	3	460 Valley Street	San Francisco (Noe Valley)	2	2	1446	climbsf_renting	http://www.c rent/460-vall
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	5	5	5	420 Mission Bay Boulevard North #121	San Francisco (Mission Bay)	1	1	980	climbsf_renting	http://www.c rent/420-mis

5 rows × 98 columns

In [754]:
```
# filter out overshoot error
overshoot = data[(data.error <= -500)]
columns = data.columns - ['error','latitude', 'longitude', 'address', 'origin_url','price','neighb
orhood']
overshoot = data.drop(columns,1)
overshoot.sort('error',ascending=True,inplace=True)
overshoot.head(30)
```

Out[754]:

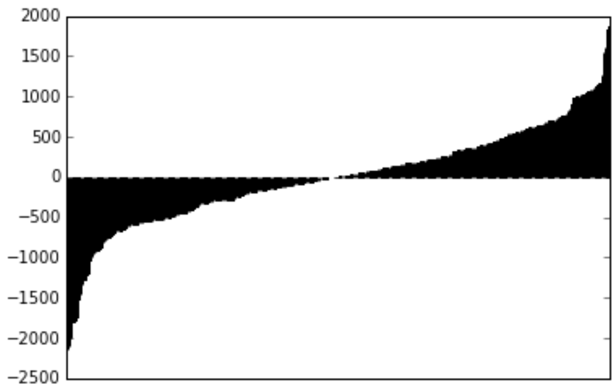
	address	neighborhood	origin_url	latitude	longitude	price	error
232	2560 Vallejo Street	San Francisco (Pacific Heights)	http://www.climbsf.com/for-rent/2560-vallejo-st/	37.7950	-122.439	7050	-2178.797974
273	301 Mission Street #29F	San Francisco (South Beach)	http://www.climbsf.com/for-rent/301-mission-st...	37.7905	-122.396	7975	-2072.695315
233	338 Spear Street #39E	San Francisco (South Beach)	http://www.climbsf.com/for-rent/338-spear-st-39e/	37.7894	-122.391	7975	-1960.351923
89	88 King Street #904	San Francisco (South Beach)	http://www.climbsf.com/for-rent/88-king-st-904/	37.7807	-122.389	6250	-1823.586896
283	234 Grand View Avenue	San Francisco (Noe Valley)	http://www.climbsf.com/for-rent/234-grand-view...	37.7545	-122.441	7300	-1821.018910
158	338 Spear Street #39A	San Francisco (South Beach)	http://www.climbsf.com/for-rent/338-spear-st-39a/	37.7894	-122.391	6700	-1784.448942
382	480 Mission Bay Boulevard North #PH1606	San Francisco (Mission Bay)	http://www.climbsf.com/for-rent/480-mission-ba...	37.7731	-122.393	7500	-1459.591211
299	401 Harrison Street #3803	San Francisco (Rincon Hill)	http://www.climbsf.com/for-rent/401-harrison-s...	37.7864	-122.392	7225	-1445.436698

381	301 Main Street #35F	San Francisco (South Beach)	http://www.climbsf.com/for-rent/301-main-st-35f/	37.7894	-122.391	7000	-1332.243966
263	163 Liberty Street	San Francisco (Noe Valley)	http://www.climbsf.com/for-rent/163-liberty-st/	37.7571	-122.425	7500	-1175.150774
364	1414 Douglass Street	San Francisco (Noe Valley)	http://www.climbsf.com/for-rent/1414-douglass-st/	37.7452	-122.438	7400	-1173.230424
113	301 Mission Street #701	San Francisco (SOMA)	http://www.climbsf.com/for-rent/301-mission-st...	37.7905	-122.396	7400	-1103.332016
119	1837 Jefferson Street	San Francisco (Marina)	http://www.climbsf.com/for-rent/1837-jefferson...	37.8045	-122.443	6200	-1084.530022
134	301 Main Street #25E	San Francisco (South Beach)	http://www.climbsf.com/for-rent/301-main-st-25e/	37.7894	-122.391	5800	-1041.888269
204	1839 Jefferson Street	San Francisco (Marina)	http://www.climbsf.com/for-rent/1839-jefferson...	37.8048	-122.443	6400	-979.770628
293	425 1st Street #3402	San Francisco (Rincon Hill)	http://www.climbsf.com/for-rent/425-1st-st-3402/	37.7858	-122.392	6600	-915.174261
369	333 Fremont Street #802	San Francisco (South Beach)	http://www.climbsf.com/for-rent/333-fremont-st...	37.7877	-122.393	4275	-913.908333
235	1880 Jackson Street	San Francisco (Pacific Heights)	http://www.climbsf.com/for-rent/1880-jackson-s...	37.7938	-122.426	4800	-892.309226
285	1160 Mission Street #1212	San Francisco (SOMA)	http://www.climbsf.com/for-rent/1160-mission-s...	37.7784	-122.412	3700	-853.286609
174	301 Main Street #14F	San Francisco (South Beach)	http://www.climbsf.com/for-rent/301-main-st-14f/	37.7894	-122.391	5950	-846.727924
72	235 Berry Street #107	San Francisco (Mission Bay)	http://www.climbsf.com/for-rent/235-berry-st-1...	37.7749	-122.394	7500	-823.611934
270	333 1st Street #N305	San Francisco (Rincon Hill)	http://www.climbsf.com/for-rent/333-1st-st-n305/	37.7870	-122.394	3900	-780.094499
161	219 Brannan Street #7E	San Francisco (South Beach)	http://www.climbsf.com/for-rent/219-brannan-st...	37.7830	-122.390	5200	-760.646739
123	480 Mission Bay Boulevard North #1608	San Francisco (Mission Bay)	http://www.climbsf.com/for-rent/480-mission-ba...	37.7711	-122.389	5475	-739.637794
70	338 Spear Street #18B	San Francisco (South Beach)	http://www.climbsf.com/for-rent/338-spear-st-18b/	37.7894	-122.391	5700	-727.512249
109	229 Brannan Street #12J	San Francisco (South Beach)	http://www.climbsf.com/for-rent/229-brannan-st...	37.7826	-122.390	5950	-710.945004
316	35 Dolores Street #410	San Francisco (Mission Dolores)	http://www.climbsf.com/for-rent/35-dolores-st-...	37.7686	-122.427	6050	-670.457098
203	461 2nd St. #557T	San Francisco (South Beach)	http://www.climbsf.com/for-rent/461-2nd-st-557t/	37.7838	-122.394	6750	-661.047512
7	1160 Mission Street #1112	San Francisco (SOMA)	http://www.climbsf.com/for-rent/1160-mission-s...	37.7784	-122.412	3750	-647.836101
13	1615 Broadway #12	East Bay (Oakland)	http://www.climbsf.com/for-rent/1615-broadway-12/	37.8062	-122.270	5500	-645.162355

In [741]:
```
data = data[(data.sqft <= 2500) & (data.price <= 8000) & (data.price != 0) & (data.bedrooms <= 4)
& (data.bathrooms <= 3) & (data.sqft != 0)]



# add squared square footage to the table
squared = data.adj_sqft ** 2
squared.name = 'sqft_squared'

squared_beds = data.bedrooms ** 2
squared_beds.name = 'beds_squared'

data = pd.concat([data, squared, squared_beds], axis=1)
#data = pd.concat([data, squared_beds], axis=1)



# create X and y
feature_cols = ['adj_sqft', 'bedrooms', 'bathrooms','sqft_squared','beds_squared']

X = data[feature_cols]
y = data.price

# instantiate, fit
lm = LinearRegression()
lm.fit(X, y)

# print coefficients
print("Intercept: %.2f" % lm.intercept_)
# The mean square error
print("Residual sum of squares: %.2f"
      % np.mean((lm.predict(X) - y) ** 2))
# Explained variance score: 1 is perfect prediction
print('Variance score: %.2f' % lm.score(X, y))
print zip(feature_cols, lm.coef_)

# calculate predictions for the data set and plot errors
predictions = lm.predict(X)
errors = predictions-y
errors.name = 'Error'

# visualize the relationship between the features and the response using scatterplots
errors.sort()
errors.plot(kind='bar').get_xaxis().set_ticks([])
```

```
Intercept: 307.20
Residual sum of squares: 381591.18
Variance score: 0.79
[('adj_sqft', 2.8234172396831432), ('bedrooms', 47.831996828432828), ('bathrooms', 237.23917675426
785), ('sqft_squared', -6.3971499961937588e-05), ('beds_squared', -6.3970609468664392e-05)]
```

Out[741]: []



In [750]:

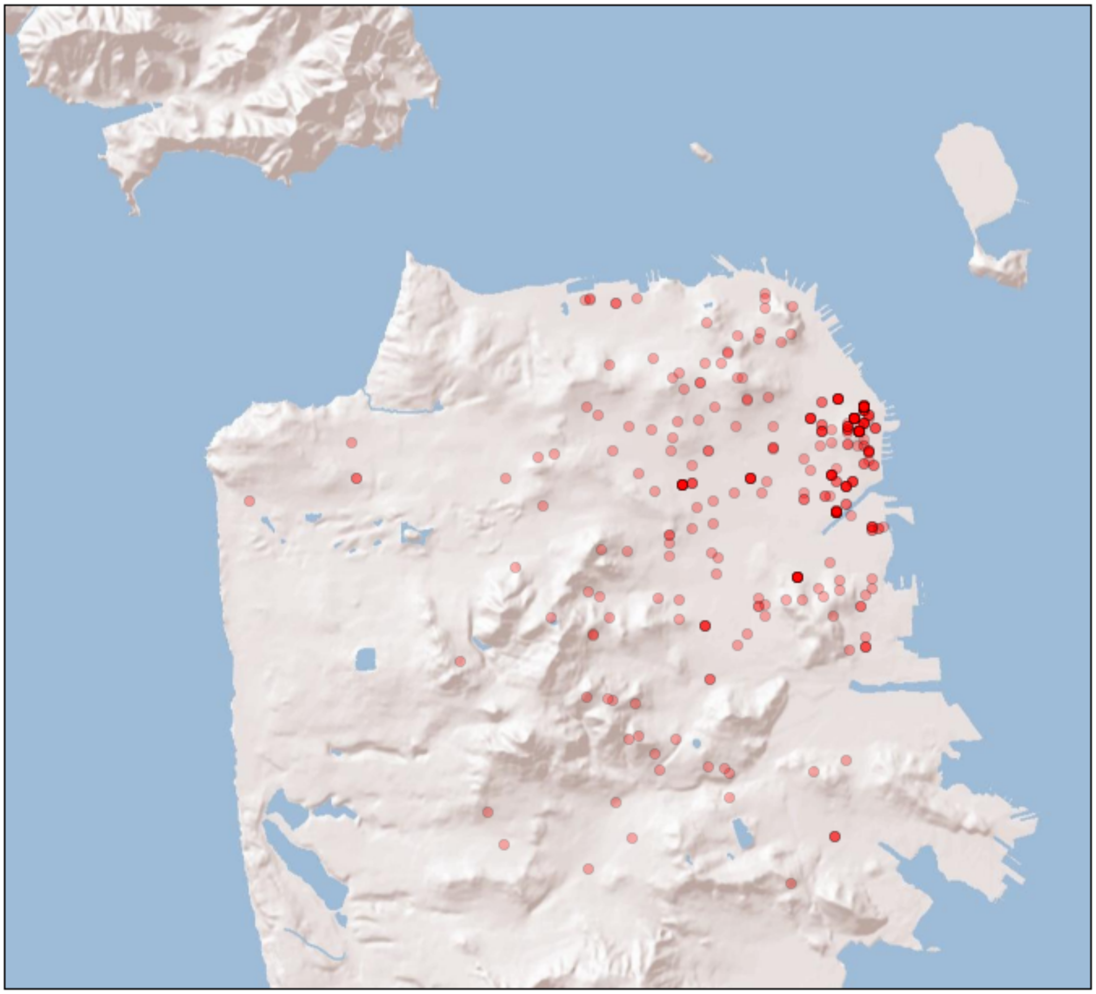
```
import statsmodels.formula.api as sm
result = sm.ols(formula="price ~ adj_sqft + bedrooms + bathrooms", data=data).fit()
print result.params
print result.summary()
```

```
Intercept    777.494958
adj_sqft       2.004296
bedrooms     163.981555
bathrooms    222.826018
dtype: float64
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.792
Model:                            OLS   Adj. R-squared:                  0.789
Method:                 Least Squares   F-statistic:                     360.6
Date:                Sun, 02 Aug 2015   Prob (F-statistic):           1.13e-96
Time:                        13:58:07   Log-Likelihood:                -2269.2
No. Observations:                 289   AIC:                             4546.
Df Residuals:                     285   BIC:                             4561.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
Intercept    777.4950    115.897      6.708      0.000       549.372  1005.618
adj_sqft       2.0043      0.094     21.240      0.000         1.819     2.190
bedrooms     163.9816     69.444      2.361      0.019        27.294   300.669
bathrooms    222.8260    100.200      2.224      0.027        25.601   420.051
==============================================================================
Omnibus:                       14.851   Durbin-Watson:                   1.707
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               29.213
Skew:                           0.244   Prob(JB):                     4.53e-07
Kurtosis:                       4.479   Cond. No.                     5.30e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 5.3e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [812]:
```
from mpl_toolkits.basemap import Basemap
import fiona
```

In [816]:
```
plt.figure(figsize=(12,12))


# Create the Basemap
event_map = Basemap(projection='merc',
                    resolution='h', epsg=2227,
                    lat_0 = 37.7, lon_0=-122.4, # Map center
                    llcrnrlon=-122.55, llcrnrlat=37.7, # Lower left corner
                    urcrnrlon=-122.35, urcrnrlat=37.85) # Upper right corner

# Draw important features
event_map.arcgisimage(service='World_Shaded_Relief', xpixels = 1500, verbose= True)

# add neighborhoods
#event_map.readshapefile(
#    'data/Realtor_Neighborhoods/Realtor_Neighborhoods', 'SF', color='none', zorder=2)

# create array storing lats and longs
listing_coords = zip(data.latitude,data.longitude)

# Draw the points on the map:
for longitude, latitude in listing_coords:
    x, y = event_map(latitude, longitude) # Convert lat, long to y,x
    event_map.plot(x,y, 'ro', alpha=0.3)
```

```
http://server.arcgisonline.com/ArcGIS/rest/services/World_Shaded_Relief/MapServer/export?bbox=5968
621.97922,2083843.65958,6027551.68158,2137245.61137&bboxSR=2227&imageSR=2227&size=1500,1359&dpi=96
&format=png32&f=image
```



In [789]:

Out[789]: 37.429900000000004

In [ ]: