In [547]:
```
%load_ext sql
```

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

In [548]:
```
%sql mysql://prod:nerd@52.2.153.189/rental_nerd
```

Out[548]:
```
u'Connected: prod@rental_nerd'
```

In [549]:
```
result = %sql (SELECT \
properties.address, \
properties.bedrooms, \
properties.bathrooms, \
properties.sqft, \
properties.source, \
properties.longitude, \
properties.latitude, \
properties.elevation, \
property_transactions.transaction_type, \
property_transaction_logs.price, \
property_transaction_logs.transaction_status, \
property_transaction_logs.days_on_market, \
property_transaction_logs.date_closed, \
property_transaction_logs.date_listed, \
neighborhoods.name as 'neighborhood', \
neighborhoods.id as 'nid' \
FROM \
properties, \
property_transactions, \
property_transaction_logs, \
property_neighborhoods, \
neighborhoods \
WHERE \
properties.id = property_transactions.property_id AND \
property_transactions.property_transaction_log_id = property_transaction_logs.id AND \
property_transactions.transaction_type = "rental" AND \
properties.id = property_neighborhoods.property_id AND \
property_neighborhoods.neighborhood_id = neighborhoods.id)

data = result.DataFrame()
```

```
727 rows affected.
```

In [550]:
```
from time import gmtime, strftime
result.csv(filename=strftime("%Y%m%d")+ " rentals.csv")
```

Out[550]: CSV results (./files/20150920 rentals.csv)

In [551]:
```
# 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

data.head()
```

Out[551]:

	address	bedrooms	bathrooms	sqft	source	longitude	latitude	elevation	transaction_type	price	transaction_status	days_on_market	da
0	814 Hayes Street #2	3	2	1200	climbsf_rented	-122.430	37.7762	42.1639	rental	5000	closed	NaN	20
1	Mcallister St San Francisco, CA 94115	4	2	1700	zillow_sf	-122.436	37.7781	60.9689	rental	8900	open	NaN	No
2	Mcallister St San Francisco, CA 94115	2	1	1150	zillow_sf	-122.436	37.7781	60.9689	rental	5900	open	NaN	No
3	1301 Fulton St APT 207, San Francisco, CA 94117	2	1	925	zillow_sf	-122.439	37.7767	63.5880	rental	3800	open	NaN	No
4	Anzavista Ave San Francisco, CA 94115	0	1	750	zillow_sf	-122.444	37.7796	106.3460	rental	2295	open	NaN	No

In [552]:
```
import datetime

Date_final = [0.1] * len(data)

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

    elif data["date_listed"][x] is not None :
        # print " row: "+ `x` + ": using date_listed"
        Date_final[x] = data["date_listed"][x]
    else:
        Date_final[x] = data["date_closed"][2]
        print " row: "+ `x` + ": we are screwed"



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

data.head()
```

Out[552]:

	address	bedrooms	bathrooms	sqft	source	longitude	latitude	elevation	transaction_type	price	transaction_status	days_on_market	da
0	814 Hayes Street #2	3	2	1200	climbsf_rented	-122.430	37.7762	42.1639	rental	5000	closed	NaN	20
1	Mcallister St San Francisco, CA 94115	4	2	1700	zillow_sf	-122.436	37.7781	60.9689	rental	8900	open	NaN	No
2	Mcallister St San Francisco, CA 94115	2	1	1150	zillow_sf	-122.436	37.7781	60.9689	rental	5900	open	NaN	No
3	1301 Fulton St APT 207, San Francisco, CA 94117	2	1	925	zillow_sf	-122.439	37.7767	63.5880	rental	3800	open	NaN	No
4	Anzavista Ave San Francisco, CA 94115	0	1	750	zillow_sf	-122.444	37.7796	106.3460	rental	2295	open	NaN	No

In [553]:
```
# create neighborhoods from lat/long coordinates
import fiona
import shapely as shapely
from geopandas import GeoSeries, GeoDataFrame
from shapely.geometry import Point
from shapely.geometry import asShape
```

In [554]:
```
# create a column of GeoSeries - each house should be represented by a point
pts = GeoSeries([Point(x, y) for x, y in zip(data['longitude'], data['latitude'])])
data['latlong'] = pts
```

In [ ]:

In [555]:
```
# filter out any outliers, defined as rent >$10k or >2,500 sq ft, or not in SF

print "Entries before filter: " + `len(data)`
data = data[  (data.sqft <= 2500)
            & (data.price <= 8000)
            & (data.price != 0)
            & (data.bedrooms <= 4)
            & (data.bathrooms <= 3)
            & (data.sqft != 0)
            & (data.address > '(Undisclosed Address) San Francisco, CA 94999')  # eliminate (Undisclosed)
            & ((data.source == 'climbsf_rented') | (data.Date > datetime.datetime(2015, 8, 1)) )] # eliminate listings old
er than 2 months

print "Entries after filter: " + `len(data)`
data.head()
```

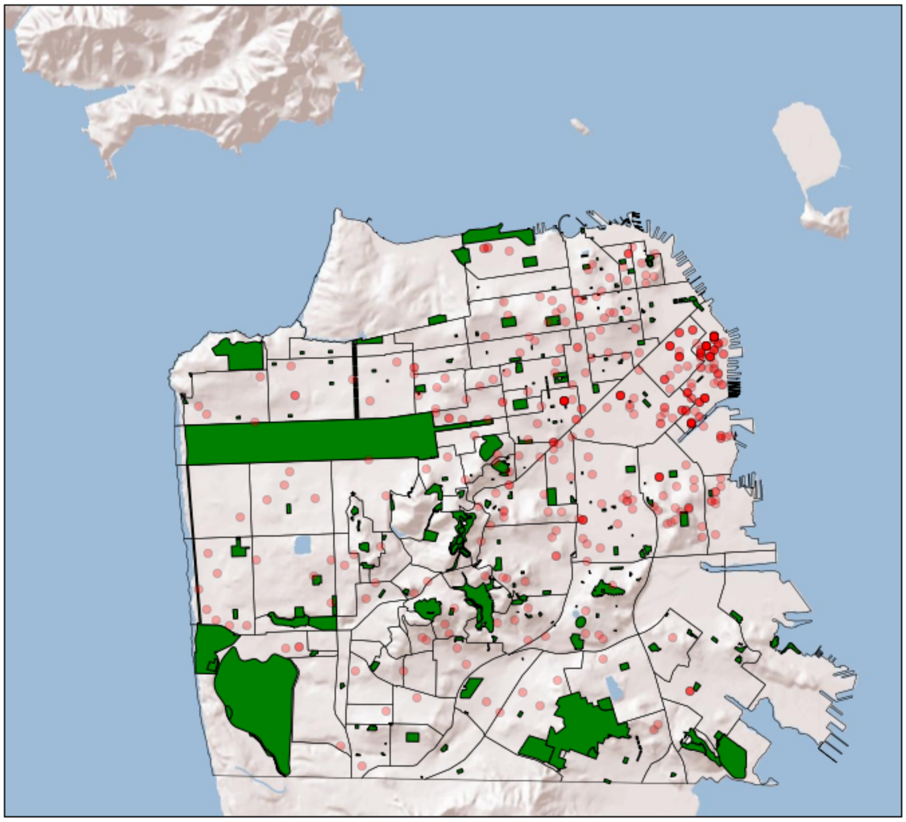
```
Entries before filter: 727
Entries after filter: 420
```

Out[555]:

	address	bedrooms	bathrooms	sqft	source	longitude	latitude	elevation	transaction_type	price	transaction_status	days_on_market	da
0	814 Hayes Street #2	3	2	1200	climbsf_rented	-122.430	37.7762	42.16390	rental	5000	closed	NaN	2(
3	1301 Fulton St APT 207, San Francisco, CA 94117	2	1	925	zillow_sf	-122.439	37.7767	63.58800	rental	3800	open	NaN	N
4	Anzavista Ave San Francisco, CA 94115	0	1	750	zillow_sf	-122.444	37.7796	106.34600	rental	2295	open	NaN	N
7	1180 Broderick St APT 304, San Francisco, CA 9...	2	2	1500	zillow_sf	-122.441	37.7809	72.96970	rental	6500	open	NaN	N
8	5800 Third Street #1109	3	3	1500	climbsf_rented	-122.395	37.7253	7.88801	rental	4500	closed	NaN	2(

In [556]:
```
from mpl_toolkits.basemap import Basemap
import fiona
from matplotlib.patches import Polygon
from matplotlib.collections import PatchCollection

fig = plt.figure(figsize=(12,12))
ax = fig.add_subplot(111)

# 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_4326/hoods_4326', 'SF', color='black', zorder=2)

# add parks
event_map.readshapefile(
    'data/RPD_Parks_4326/parks_4326', 'parks', color='none', zorder=2)

# fill in parks in green
patches   = []

for shape in event_map.parks:
    patches.append( Polygon(np.array(shape), True) )

ax.add_collection(PatchCollection(patches, facecolor= 'green', 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)

plt.show()
```

```
http://server.arcgisonline.com/ArcGIS/rest/services/World_Shaded_Relief/MapServer/export?bbox=5968621.97922,2083843.65958,
6027551.68158,2137245.61137&bboxSR=2227&imageSR=2227&size=1500,1359&dpi=96&format=png32&f=image
```



In [557]:
```
# 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 means columns)
data = pd.concat([data, area_dummies, year_dummies], axis=1)

data.head()
```

```
Base neighborhood: neighborhood_Alamo Square
```

Out[557]:

	address	bedrooms	bathrooms	sqft	source	longitude	latitude	elevation	transaction_type	price	...	neighborhood_West Portal	neighborhood_ Addition
0	814 Hayes Street #2	3	2	1200	climbsf_rented	-122.430	37.7762	42.16390	rental	5000	...	0	0
3	1301 Fulton St APT 207, San Francisco, CA 94117	2	1	925	zillow_sf	-122.439	37.7767	63.58800	rental	3800	...	0	0
4	Anzavista Ave San Francisco, CA 94115	0	1	750	zillow_sf	-122.444	37.7796	106.34600	rental	2295	...	0	0
7	1180 Broderick St APT 304, San Francisco, CA 9...	2	2	1500	zillow_sf	-122.441	37.7809	72.96970	rental	6500	...	0	0
8	5800 Third Street #1109	3	3	1500	climbsf_rented	-122.395	37.7253	7.88801	rental	4500	...	0	0

5 rows × 89 columns

In [558]:
```
# 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()
```

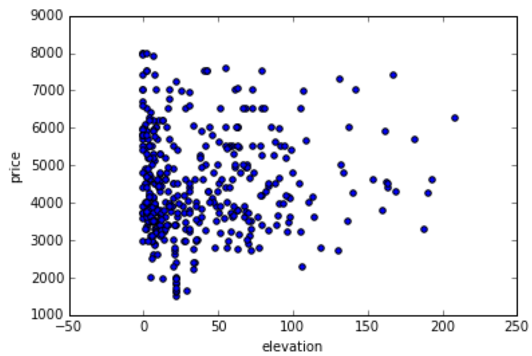
Out[558]:

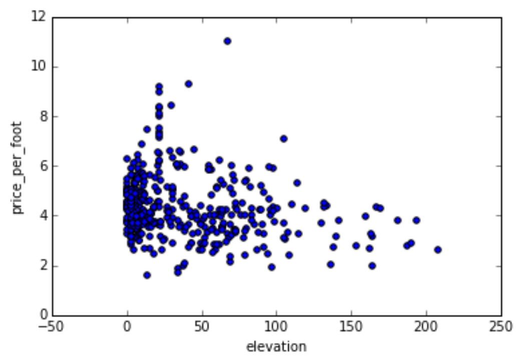
	address	bedrooms	bathrooms	sqft	source	longitude	latitude	elevation	transaction_type	price	...	neighborhood_Western Addition	neighborho Highlands
0	814 Hayes Street #2	3	2	1200	climbsf_rented	-122.430	37.7762	42.16390	rental	5000	...	0	0
3	1301 Fulton St APT 207, San Francisco, CA 94117	2	1	925	zillow_sf	-122.439	37.7767	63.58800	rental	3800	...	0	0
4	Anzavista Ave San Francisco, CA 94115	0	1	750	zillow_sf	-122.444	37.7796	106.34600	rental	2295	...	0	0
7	1180 Broderick St APT 304, San Francisco, CA 9...	2	2	1500	zillow_sf	-122.441	37.7809	72.96970	rental	6500	...	0	0
8	5800 Third Street #1109	3	3	1500	climbsf_rented	-122.395	37.7253	7.88801	rental	4500	...	0	0

5 rows × 90 columns

In [559]:
```
# 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[559]: <matplotlib.axes._subplots.AxesSubplot at 0x11dfe3050>





In [560]:
```
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 [561]:
```
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: 0.97
Variance score: 0.38
Base area is neighborhood_Alamo Square: $4.14
```

Out[561]:

Neighborhood	$ per square (+/-)
neighborhood_Westwood Park	-1.72
neighborhood_Ingleside	-1.67
neighborhood_Visitacion Valley	-1.63
neighborhood_Portola	-1.49
neighborhood_Inner Richmond	-1.42
neighborhood_Stonestown	-1.42
neighborhood_Westwood Highlands	-1.4
neighborhood_Diamond Heights	-1.33
neighborhood_Silver Terrace	-1.3
neighborhood_Central Richmond	-1.27
neighborhood_Outer Richmond	-1.16
neighborhood_Central Sunset	-1.14
neighborhood_Merced Heights	-1.03
neighborhood_Forest Hill	-0.98
neighborhood_Miraloma Park	-0.98
neighborhood_Parkside	-0.98
neighborhood_Bayview	-0.96
neighborhood_Mission Terrace	-0.95
neighborhood_West Portal	-0.81
neighborhood_Lake Shore	-0.78
neighborhood_Ingleside Heights	-0.69
neighborhood_Excelsior	-0.67

neighborhood_Forest Hills Extension	-0.64
neighborhood_Inner Parkside	-0.62
neighborhood_Inner Sunset	-0.62
neighborhood_Haight Ashbury	-0.5
neighborhood_Anza Vista	-0.44
neighborhood_Bernal Heights	-0.42
neighborhood_Van Ness/Civic Center	-0.41
neighborhood_Glen Park	-0.39
neighborhood_Downtown	-0.37
neighborhood_Western Addition	-0.37
neighborhood_Sunnyside	-0.34
neighborhood_Golden Gate Heights	-0.3
neighborhood_Mission Bay	-0.29
neighborhood_North Beach	-0.29
neighborhood_Lone Mountain	-0.27
neighborhood_Central Waterfront/Dogpatch	-0.14
neighborhood_Buena Vista Park/Ashbury Heights	-0.1
neighborhood_South of Market	-0.01
neighborhood_Potrero Hill	0.02
neighborhood_Eureka Valley / Dolores Heights	0.07
neighborhood_Lower Pacific Heights	0.09
neighborhood_Cole Valley/Parnassus Heights	0.14
neighborhood_Outer Parkside	0.18
neighborhood_South Beach	0.23
neighborhood_Jordan Park / Laurel Heights	0.27
neighborhood_Noe Valley	0.28
neighborhood_Twin Peaks	0.36
neighborhood_Marina	0.39
neighborhood_Pacific Heights	0.46
neighborhood_Inner Mission	0.49
neighborhood_Outer Sunset	0.64
neighborhood_Yerba Buena	0.66
neighborhood_Corona Heights	0.73
neighborhood_Telegraph Hill	0.74
neighborhood_Nob Hill	0.79
neighborhood_North Panhandle	1.09
neighborhood_North Waterfront	1.1
neighborhood_Russian Hill	1.16
neighborhood_Duboce Triangle	1.41
neighborhood_Mission Dolores	1.53
neighborhood_Ingleside Terrace	1.85
neighborhood_Tenderloin	2.43
neighborhood_Hayes Valley	2.58

In [562]:
```
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[562]:

Neighborhood	$ per sqft
neighborhood_Westwood Park	2.41970021413
neighborhood_Ingleside	2.46875
neighborhood_Visitacion Valley	2.51138053536
neighborhood_Portola	2.64285714286
neighborhood_Inner Richmond	2.7149321267
neighborhood_Stonestown	2.71739130435
neighborhood_Westwood Highlands	2.73333333333
neighborhood_Diamond Heights	2.8085106383
neighborhood_Silver Terrace	2.83464566929
neighborhood_Central Richmond	2.86287503734
neighborhood_Outer Richmond	2.97578486133
neighborhood_Central Sunset	3.00224338741
neighborhood_Merced Heights	3.10344827586
neighborhood_Miraloma Park	3.15637492916
neighborhood_Parkside	3.1588500265
neighborhood_Forest Hill	3.15985130112
neighborhood_Bayview	3.17631505701
neighborhood_Mission Terrace	3.18513033175
neighborhood_West Portal	3.33095238095
neighborhood_Lake Shore	3.3597866078
neighborhood_Ingleside Heights	3.4516765286
neighborhood_Excelsior	3.46732026144
neighborhood_Forest Hills Extension	3.49304851557
neighborhood_Inner Parkside	3.5148488121
neighborhood_Inner Sunset	3.51773066953
neighborhood_Haight Ashbury	3.63636363636
neighborhood_Anza Vista	3.69666666667
neighborhood_Bernal Heights	3.71361480988
neighborhood_Van Ness/Civic Center	3.72264292176
neighborhood_Glen Park	3.74975024975
neighborhood_Downtown	3.7644395011
neighborhood_Western Addition	3.76897132069
neighborhood_Sunnyside	3.79500437012

neighborhood_Golden Gate Heights	3.83333333333
neighborhood_Mission Bay	3.85094171788
neighborhood_North Beach	3.851726278
neighborhood_Lone Mountain	3.86835131596
neighborhood_Central Waterfront/Dogpatch	3.99275046752
neighborhood_Buena Vista Park/Ashbury Heights	4.03881278539
neighborhood_South of Market	4.12424899483
neighborhood_Potrero Hill	4.15950525812
neighborhood_Eureka Valley / Dolores Heights	4.20614100903
neighborhood_Lower Pacific Heights	4.23156414839
neighborhood_Cole Valley/Parnassus Heights	4.27572115385
neighborhood_Outer Parkside	4.31838192588
neighborhood_South Beach	4.3660239643
neighborhood_Jordan Park / Laurel Heights	4.40412895928
neighborhood_Noe Valley	4.42230152536
neighborhood_Twin Peaks	4.500450045
neighborhood_Marina	4.53031759648
neighborhood_Pacific Heights	4.59407317018
neighborhood_Inner Mission	4.63204917563
neighborhood_Outer Sunset	4.7775
neighborhood_Yerba Buena	4.79430413324
neighborhood_Corona Heights	4.86243722756
neighborhood_Telegraph Hill	4.87803918764
neighborhood_Nob Hill	4.92263404674
neighborhood_North Panhandle	5.223665503
neighborhood_North Waterfront	5.24109014675
neighborhood_Russian Hill	5.29620601173
neighborhood_Duboce Triangle	5.54750351629
neighborhood_Mission Dolores	5.67139509236
neighborhood_Ingleside Terrace	5.99
neighborhood_Tenderloin	6.57142857143
neighborhood_Hayes Valley	6.71721249918

In [563]:

```
# 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[563]:

Neighborhood	Multiplier
neighborhood_Alamo Square	0
neighborhood_Westwood Park	-0.41516227813
neighborhood_Ingleside	-0.403307022319
neighborhood_Visitacion Valley	-0.393003289222
neighborhood_Portola	-0.361225600747
neighborhood_Inner Richmond	-0.343805190934
neighborhood_Stonestown	-0.343210811578
neighborhood_Westwood Highlands	-0.33935764834
neighborhood_Diamond Heights	-0.321187412409
neighborhood_Silver Terrace	-0.314870616676
neighborhood_Central Richmond	-0.308047622984
neighborhood_Outer Richmond	-0.280757496773
neighborhood_Central Sunset	-0.274362512787
neighborhood_Merced Heights	-0.249901451016
neighborhood_Miraloma Park	-0.237109162469
neighborhood_Parkside	-0.236510935349
neighborhood_Forest Hill	-0.236268928854
neighborhood_Bayview	-0.232289665044
neighborhood_Mission Terrace	-0.230159027056
neighborhood_West Portal	-0.194914067968
neighborhood_Lake Shore	-0.187944880858
neighborhood_Ingleside Heights	-0.165735232064
neighborhood_Excelsior	-0.161954166533
neighborhood_Forest Hills Extension	-0.155735688126
neighborhood_Inner Parkside	-0.150466590871
neighborhood_Inner Sunset	-0.149770050478
neighborhood_Haight Ashbury	-0.121096649676
neighborhood_Anza Vista	-0.10652150245
neighborhood_Bernal Heights	-0.102425162991
neighborhood_Van Ness/Civic Center	-0.100243082602
neighborhood_Glen Park	-0.0936912842193
neighborhood_Downtown	-0.0901409153562
neighborhood_Western Addition	-0.0890455817174
neighborhood_Sunnyside	-0.0827534347661
neighborhood_Golden Gate Heights	-0.0734893848666
neighborhood_Mission Bay	-0.0692334661201
neighborhood_North Beach	-0.0690438391769
neighborhood_Lone Mountain	-0.0650255937483
neighborhood_Central Waterfront/Dogpatch	-0.0349585151984
neighborhood_Buena Vista Park/Ashbury Heights	-0.0238253256874
neighborhood_South of Market	-0.00317552873947
neighborhood_Potrero Hill	0.00534585444026
neighborhood_Eureka Valley / Dolores Heights	0.0166176418117
neighborhood_Lower Pacific Heights	0.0227623744619
neighborhood_Cole Valley/Parnassus Heights	0.033435052971
neighborhood_Outer Parkside	0.0437460942253

neighborhood_South Beach	0.0552610997006
neighborhood_Jordan Park / Laurel Heights	0.064471016831
neighborhood_Noe Valley	0.0688632973654
neighborhood_Twin Peaks	0.0877516711932
neighborhood_Marina	0.0949706112333
neighborhood_Pacific Heights	0.110380232749
neighborhood_Inner Mission	0.119558973316
neighborhood_Outer Sunset	0.154714207948
neighborhood_Yerba Buena	0.158775740424
neighborhood_Corona Heights	0.175243401763
neighborhood_Telegraph Hill	0.179014370884
neighborhood_Nob Hill	0.189792877927
neighborhood_North Panhandle	0.262551705679
neighborhood_North Waterfront	0.266763214566
neighborhood_Russian Hill	0.280084632122
neighborhood_Duboce Triangle	0.340822842261
neighborhood_Mission Dolores	0.370767240615
neighborhood_Ingleside Terrace	0.447773543821
neighborhood_Tenderloin	0.588303911657
neighborhood_Hayes Valley	0.623539656851

In [564]:

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

# 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[564]:

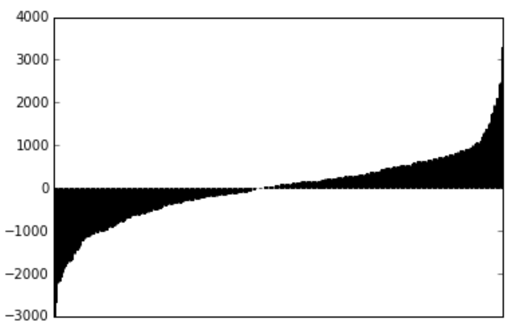
	address	bedrooms	bathrooms	sqft	source	longitude	latitude	elevation	transaction_type	price	...	neighborhood_Westwood Park	neighborl Buena
0	814 Hayes Street #2	3	2	1200	climbsf_rented	-122.430	37.7762	42.16390	rental	5000	...	0	0
3	1301 Fulton St APT 207, San Francisco, CA 94117	2	1	925	zillow_sf	-122.439	37.7767	63.58800	rental	3800	...	0	0
4	Anzavista Ave San Francisco, CA 94115	0	1	750	zillow_sf	-122.444	37.7796	106.34600	rental	2295	...	0	0
7	1180 Broderick St APT 304, San Francisco, CA 9...	2	2	1500	zillow_sf	-122.441	37.7809	72.96970	rental	6500	...	0	0
8	5800 Third Street #1109	3	3	1500	climbsf_rented	-122.395	37.7253	7.88801	rental	4500	...	0	0

5 rows × 92 columns

In [565]:
```
# 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: 1602.60
Residual sum of squares: 657185.67
Variance score: 0.64
```

Out[565]: []



In [566]:
```
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: 1.48
Variance score: 0.05
4.52008932173
```

Out[566]:
```
[(u'Year_1969', -1.6647593724935521),
 (u'Year_2011', -0.34952848782667051),
 (u'Year_2012', -1.0937945010150021),
 (u'Year_2013', -0.71231298438035173),
 (u'Year_2014', -0.4166576501058542)]
```

In [567]:
```
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': 2.8553299492385809, 'neighborhood_Alamo Square': 4.520089321732133, u'Year_2012': 3.4262948207171311, u'Yea
r_2013': 3.8077763373517812, u'Year_2011': 4.1705608339054621, u'Year_2014': 4.1034316716262786}
```

In [568]:
```
# 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[568]:
```
[(u'Year_1969', -0.36830231749836384),
 (u'Year_2011', -0.077327783357327262),
 (u'Year_2012', -0.24198515187656766),
 (u'Year_2013', -0.15758825405410082),
 (u'Year_2014', -0.092179074449393328)]
```

In [569]:
```
# calculate the adjusted Sqft (Sqft * Year_mult) for the dataset and add it as a new column to data

# 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[569]:

	address	bedrooms	bathrooms	sqft	source	longitude	latitude	elevation	transaction_type	price	...	Year_1969	Year_2011	Year_2012	Ye
0	814 Hayes Street #2	3	2	1200	climbsf_rented	-122.430	37.7762	42.16390	rental	5000	...	0	0	0	0
3	1301 Fulton St APT 207, San Francisco, CA 94117	2	1	925	zillow_sf	-122.439	37.7767	63.58800	rental	3800	...	0	0	0	0
4	Anzavista Ave San Francisco, CA 94115	0	1	750	zillow_sf	-122.444	37.7796	106.34600	rental	2295	...	0	0	0	0
7	1180 Broderick St APT 304, San Francisco, CA 9...	2	2	1500	zillow_sf	-122.441	37.7809	72.96970	rental	6500	...	0	0	0	0
8	5800 Third Street #1109	3	3	1500	climbsf_rented	-122.395	37.7253	7.88801	rental	4500	...	0	0	0	0

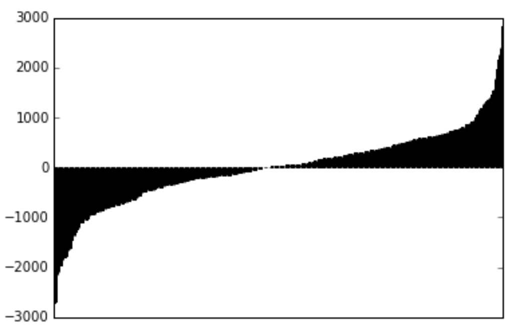
5 rows × 94 columns

In [ ]:

In [570]:
```
# 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([])
```

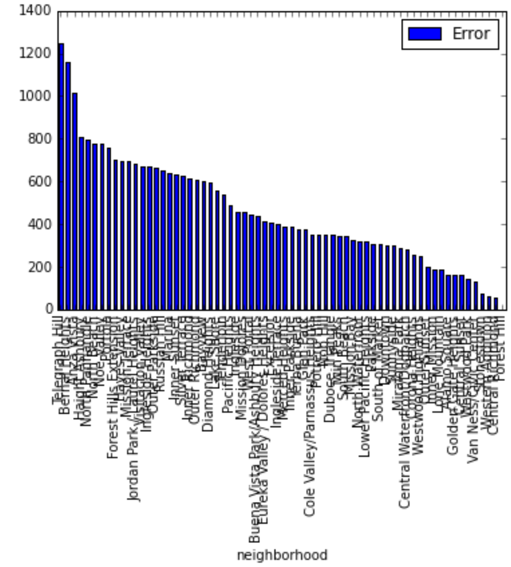
```
Intercept: 1281.34
Residual sum of squares: 560352.26
Variance score: 0.69
[('adj_sqft', 2.2214484217866937), ('bedrooms', 182.95494231541647), ('bathrooms', 321.09594348281797)]
```

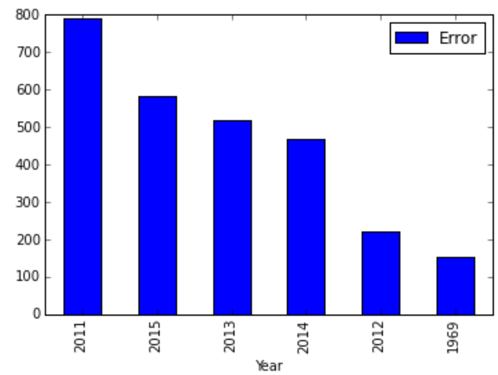
Out[570]: []

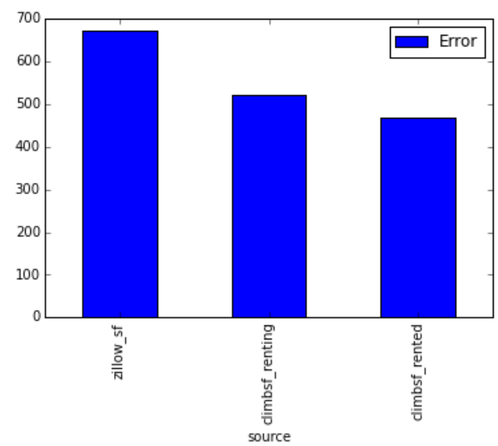


In [571]:
```
# show errors by neighborhood to see if there are any neighborhoods with funky differences

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[571]: <matplotlib.axes._subplots.AxesSubplot at 0x11f5d4150>







In [572]:
```
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', 2.2214484217866937), ('bedrooms', 182.95494231541647), ('bathrooms', 321.09594348281797)]
```

Out[572]:

Effect	Coefficient
adj_sqft	2.221448
bedrooms	182.954942
bathrooms	321.095943
base_rent	1281.34097396

In [573]:
```
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:
        i[0] = i[0][13:]
        table.append(i)
        hoodwriter.writerow(i)

    lastrow = [base_area[13:], 1]
    table.append(lastrow)
    hoodwriter.writerow(lastrow)


table
```

Out[573]:

Neighborhood	Multiplier
Westwood Park	0.584838
Ingleside	0.596693
Visitacion Valley	0.606997
Portola	0.638774

Inner Richmond	0.656195
Stonestown	0.656789
Westwood Highlands	0.660642
Diamond Heights	0.678813
Silver Terrace	0.685129
Central Richmond	0.691952
Outer Richmond	0.719243
Central Sunset	0.725637
Merced Heights	0.750099
Miraloma Park	0.762891
Parkside	0.763489
Forest Hill	0.763731
Bayview	0.76771
Mission Terrace	0.769841
West Portal	0.805086
Lake Shore	0.812055
Ingleside Heights	0.834265
Excelsior	0.838046
Forest Hills Extension	0.844264
Inner Parkside	0.849533
Inner Sunset	0.85023
Haight Ashbury	0.878903
Anza Vista	0.893478
Bernal Heights	0.897575
Van Ness/Civic Center	0.899757
Glen Park	0.906309
Downtown	0.909859
Western Addition	0.910954
Sunnyside	0.917247
Golden Gate Heights	0.926511
Mission Bay	0.930767
North Beach	0.930956
Lone Mountain	0.934974
Central Waterfront/Dogpatch	0.965041
Buena Vista Park/Ashbury Heights	0.976175
South of Market	0.996824
Potrero Hill	1.005346
Eureka Valley / Dolores Heights	1.016618
Lower Pacific Heights	1.022762
Cole Valley/Parnassus Heights	1.033435
Outer Parkside	1.043746
South Beach	1.055261
Jordan Park / Laurel Heights	1.064471
Noe Valley	1.068863
Twin Peaks	1.087752
Marina	1.094971
Pacific Heights	1.11038

Inner Mission	1.119559
Outer Sunset	1.154714
Yerba Buena	1.158776
Corona Heights	1.175243
Telegraph Hill	1.179014
Nob Hill	1.189793
North Panhandle	1.262552
North Waterfront	1.266763
Russian Hill	1.280085
Duboce Triangle	1.340823
Mission Dolores	1.370767
Ingleside Terrace	1.447774
Tenderloin	1.588304
Hayes Valley	1.62354
Alamo Square	1

In [574]:

```
# 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[574]:

	address	bedrooms	bathrooms	sqft	source	longitude	latitude	elevation	transaction_type	price	...	Year_2012	Year_2013	Year_2014	pri
0	814 Hayes Street #2	3	2	1200	climbsf_rented	-122.430	37.7762	42.1639	rental	5000	...	0	0	1	4.1
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	Na
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	Na
3	1301 Fulton St APT 207, San Francisco, CA 94117	2	1	925	zillow_sf	-122.439	37.7767	63.5880	rental	3800	...	0	0	0	4.1
4	Anzavista Ave San Francisco, CA 94115	0	1	750	zillow_sf	-122.444	37.7796	106.3460	rental	2295	...	0	0	0	3.0

5 rows × 96 columns

In [575]:
```
# filter out overshoot error
overshoot = data[(data.error <= -500)]
columns = data.columns - ['error','latitude', 'longitude', 'address', 'origin_url','price','neighborhood']
overshoot = data.drop(columns,1)
overshoot.sort('error',ascending=True,inplace=True)
overshoot.head(30)
```

Out[575]:

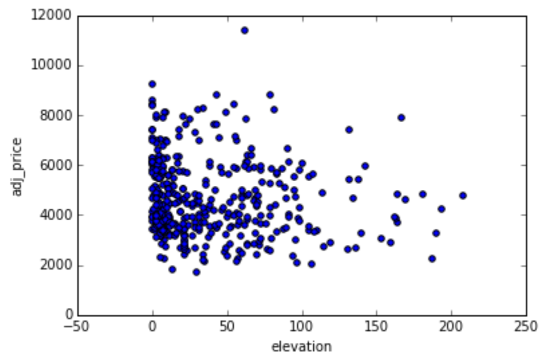
	address	longitude	latitude	price	neighborhood	error
651	55 Rodgers St, San Francisco, CA 94103	-122.409	37.7750	7900	South of Market	-2720.781683
610	301 Main St UNIT 35A, San Francisco, CA 94105	-122.391	37.7894	7950	South Beach	-2688.101378
209	20th St San Francisco, CA 94110	-122.416	37.7588	6200	Inner Mission	-2176.269877
530	338 Spear Street #39E	-122.391	37.7894	7975	South Beach	-2135.914619
384	1840 Broadway, San Francisco, CA 94109	-122.428	37.7953	7580	Pacific Heights	-2083.909114
288	480 Mission Bay Boulevard North #PH1606	-122.393	37.7731	7500	Mission Bay	-1951.941094
104	301 Mission Street #29F	-122.396	37.7905	7975	Yerba Buena	-1944.217492
668	525 Greenwich St, San Francisco, CA 94133	-122.408	37.8025	6000	Telegraph Hill	-1857.400488
474	338 Spear Street #39A	-122.391	37.7894	6700	South Beach	-1843.332280
647	Bluxome St San Francisco, CA 94107	-122.398	37.7760	8000	South of Market	-1806.580371
560	401 Harrison Street #3803	-122.392	37.7864	7225	South Beach	-1796.662603
472	88 King Street #904	-122.389	37.7807	6250	South Beach	-1766.567295
376	2560 Vallejo Street	-122.439	37.7950	7050	Pacific Heights	-1666.540029
329	296 Francisco Street	-122.410	37.8053	5675	North Beach	-1665.467028
16	212 Cortland Ave, San Francisco, CA 94110	-122.419	37.7394	6500	Bernal Heights	-1634.902866
590	301 Main Street #35F	-122.391	37.7894	7000	South Beach	-1609.169932
343	209 Ashbury St, San Francisco, CA 94117	-122.448	37.7736	6500	North Panhandle	-1446.490355
69	234 Grand View Avenue	-122.441	37.7545	7300	Eureka Valley / Dolores Heights	-1379.183095
43	16 Flint St, San Francisco, CA 94114	-122.437	37.7643	7500	Corona Heights	-1369.951929
322	45 Clipper St, San Francisco, CA 94114	-122.426	37.7491	5500	Noe Valley	-1359.054611
320	1151 Church St, San Francisco, CA 94114	-122.427	37.7525	5475	Noe Valley	-1274.693994
547	425 1st Street #3402	-122.392	37.7858	6600	South Beach	-1241.988846
7	1180 Broderick St APT 304, San Francisco, CA 9...	-122.441	37.7809	6500	Anza Vista	-1233.332657
123	1st St San Francisco, CA 94105	-122.395	37.7881	5100	Yerba Buena	-1206.370658
522	461 2nd St. #557T	-122.394	37.7838	6750	South Beach	-1108.452578
237	143 Riverton Dr, San Francisco, CA 94132	-122.487	37.7316	5250	Lake Shore	-1102.495414
721	Hyde St San Francisco, CA 94109	-122.418	37.7962	4999	Nob Hill	-1099.157332
344	Fell St San Francisco, CA 94117	-122.442	37.7735	3595	North Panhandle	-1075.428310
256	2309A California St, San Francisco, CA 94115	-122.433	37.7888	6500	Lower Pacific Heights	-1055.471988
594	425 1st Street #2005	-122.392	37.7858	4500	South Beach	-1050.220386

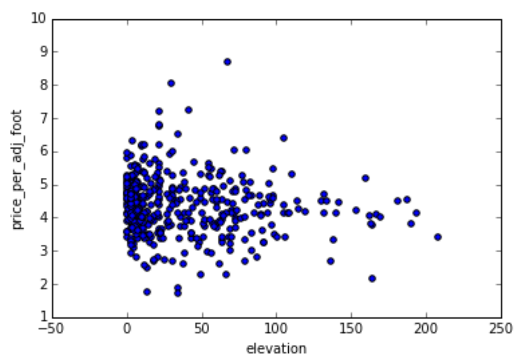
In [578]:
```
import statsmodels.formula.api as sm
result = sm.ols(formula="price ~ adj_sqft + bedrooms + bathrooms", data=data).fit()
print result.params
print result.summary()
```

```
Intercept    1281.340974
adj_sqft        2.221448
bedrooms      182.954942
bathrooms     321.095943
dtype: float64
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.689
Model:                            OLS   Adj. R-squared:                  0.687
Method:                 Least Squares   F-statistic:                     307.2
Date:                Sun, 20 Sep 2015   Prob (F-statistic):          4.31e-105
Time:                        12:21:39   Log-Likelihood:                -3375.6
No. Observations:                 420   AIC:                             6759.
Df Residuals:                     416   BIC:                             6775.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
Intercept   1281.3410    117.273     10.926      0.000      1050.820  1511.862
adj_sqft       2.2214      0.122     18.147      0.000         1.981     2.462
bedrooms     182.9549     59.978      3.050      0.002        65.057   300.853
bathrooms    321.0959     85.209      3.768      0.000       153.602   488.590
==============================================================================
Omnibus:                       21.362   Durbin-Watson:                   1.828
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               55.101
Skew:                           0.153   Prob(JB):                     1.08e-12
Kurtosis:                       4.748   Cond. No.                     3.98e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.98e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [577]:
```
price_per_adj_foot = data['price'] / data['adj_sqft']
price_per_adj_foot.name = 'price_per_adj_foot'
adj_price = data['price'] * data['area_multiplier']
adj_price.name = 'adj_price'
data = pd.concat([data, price_per_adj_foot, adj_price], axis=1)

# visualize the relationship between the features and the response using scatterplots
data.plot(kind='scatter', x='elevation', y='adj_price')
data.plot(kind='scatter', x='elevation', y='price_per_adj_foot')
```

Out[577]: <matplotlib.axes._subplots.AxesSubplot at 0x117e57b90>





In [ ]: