In [3]:
```
import pandas as pd
import numpy as np
import os
```

In [4]:
```
path = r"C:\Data\Waste_Intervention\Census_Tables\Cleaned"
fList = os.listdir(path)
```

In [5]:
```
dfs = {}

for doc in fList:
    i = "{}\\{}".format(path, doc)
    j = doc.split("_")[-1][:-4]

    dfs[j] = pd.read_csv(i, na_values=("-"))

    df = pd.read_csv(i, na_values=("-"))
    #print dfs

    df = df.fillna(0)
    print df.head(10)
```

	Id2	med_age
0	60855001001	27.2
1	60855001002	33.7
2	60855001003	26.9
3	60855001004	34.1
4	60855002001	30.6
5	60855002002	36.9
6	60855002003	36.9
7	60855002004	29.2
8	60855003001	37.3
9	60855003002	35.4

	Id2	White	AfAm	NatAm	Asian	Hawaiin	Other	Inter
0	60855001001	310	0	0	112	0	500	10
1	60855001002	397	18	12	727	0	717	65
2	60855001003	731	0	0	403	0	1418	68
3	60855001004	755	150	22	559	0	756	94
4	60855002001	796	43	50	212	63	154	250
5	60855002002	337	118	12	46	0	123	58
6	60855002003	812	83	0	250	0	376	190
7	60855002004	895	88	29	629	0	602	53
8	60855003001	265	65	7	54	0	35	127
9	60855003002	1865	215	20	726	0	202	209

	Id2	Total_HH	FamHH	NonFM	NoFamAloneHH
0	60855001001	286	174	112	86
1	60855001002	664	436	228	142
2	60855001003	561	364	197	79
3	60855001004	884	512	372	289
4	60855002001	526	256	270	171
5	60855002002	0	0	0	0
6	60855002003	615	291	324	229
7	60855002004	1004	159	845	496
8	60855003001	142	61	81	73
9	60855003002	1322	623	699	313

	Id2	HH1	HH2	HH3	HH4	HH5	HH6	HH7
0	60855001001	86	67	69	16	0	0	48
1	60855001002	142	135	157	211	19	0	0
2	60855001003	79	65	33	85	152	49	98
3	60855001004	289	296	67	114	67	24	27
4	60855002001	171	108	71	175	1	0	0
5	60855002002	0	0	0	0	0	0	0
6	60855002003	229	95	142	94	29	0	26
7	60855002004	496	241	107	55	36	69	0
8	60855003001	73	8	26	29	6	0	0
9	60855003002	313	647	164	104	58	0	36

	Id2	MedHHInc
0	60855001001	59118.0
1	60855001002	110714.0
2	60855001003	62730.0
3	60855001004	85648.0
4	60855002001	103265.0
5	60855002003	62067.0
6	60855002004	99821.0
7	60855003001	30625.0
8	60855003002	114904.0
9	60855004001	54489.0

	Id2	Vacant
0	60855001001	0

```
1  60855001002       0
2  60855001003       0
3  60855001004      89
4  60855002001      15
5  60855002002       0
6  60855002003      34
7  60855002004      44
8  60855003001       7
9  60855003002      46
           Id2  TotalHH  OwnerHH  RenterHH
0  60855001001      888      290       598
1  60855001002     1883      607      1276
2  60855001003     2620      999      1621
3  60855001004     2336      760      1576
4  60855002001     1424      611       813
5  60855002002        0        0         0
6  60855002003     1711      573      1138
7  60855002004     2296      440      1856
8  60855003001      312       88       224
9  60855003002     3237     1628      1609
           Id2  AvgHHSize  AvOwnHHsize  AvRenHHsize
0  60855001001       3.10         3.02         3.15
1  60855001002       2.84         2.21         3.28
2  60855001003       4.67         4.22         5.00
3  60855001004       2.64         2.43         2.76
4  60855002001       2.71         3.20         2.43
5  60855002003       2.78         4.34         2.36
6  60855002004       2.29         1.68         2.50
7  60855003001       2.20         2.15         2.22
8  60855003002       2.45         2.39         2.51
9  60855004001       2.33         2.60         2.25
           Id2  DetachUnit  AttachUnit  TwoUnits  Units_3_4  Units_5_9  \
0  60855001001         185          17         0         15         43
1  60855001002         189         269        61        114         16
2  60855001003         273          75        84          0          0
3  60855001004         287         176        77         93         52
4  60855002001         298          16        79         40         59
5  60855002002           0           0         0          0          0
6  60855002003         119          63         0         65        266
7  60855002004         137          76        31         15        137
8  60855003001          54          18        42         26          9
9  60855003002         422         302        14         55        120

   Units_10_19  Units_20_49  Units_50plus  UnitMobile  Unit_Other
0           26            0             0           0           0
1           15            0             0           0           0
2           47           82             0           0           0
3            0           66           222           0           0
4            0           12            37           0           0
5            0            0             0           0           0
6           88           38            10           0           0
7           13          166           473           0           0
8            0            0             0           0           0
9           45            8           402           0           0
           Id2  HouseYear
0  60855001001       1948
1  60855001002       1968
```

```
2  60855001003       1986
3  60855001004       1999
4  60855002001       1947
5  60855002003       1963
6  60855002004       1998
7  60855003001       1956
8  60855003002       1997
9  60855004001       1957
           Id2  studio  bed1  bed2  bed3  bed4  bed5
0  60855001001      14    12   204    19    37     0
1  60855001002       0    64   251   320    29     0
2  60855001003       0    12   321   197    18    13
3  60855001004      17   190   390   244   108    24
4  60855002001      12   112   267   124    26     0
5  60855002002       0     0     0     0     0     0
6  60855002003      86   239   257    41     0    26
7  60855002004      51   400   362   184    36    15
8  60855003001       0    48    88     0    13     0
9  60855003002      67   338   534   262   139    28
           Id2  R100  R150  R199  R249  R299  R349  R399  R449  R499  \
0  60855001001     0     0     0     0     0     0     0     0     0
1  60855001002     0     0     0     0     0     0     0     0    17
2  60855001003     0     0     0     0     0     0     0     0     0
3  60855001004     0    35     0     0    17     0     0     0     0
4  60855002001     0     0     0     0    31     0     0     0     0
5  60855002002     0     0     0     0     0     0     0     0     0
6  60855002003     0     0     0     0     0     0     0     0     0
7  60855002004     0     0     0     0     0     0     0     0     0
8  60855003001     0     0     0     0     0     0     0     0     0
9  60855003002     0     9     0     0     0     0     0     0     0

     ...      R799  R899  R999  R1249  R1499  R1999  R2499  R2999  R3499  \
0    ...         0     0    60     43      0     57     15      0      0
1    ...        12    49     0     46     97     71     49      0     48
2    ...         0     0     0     12    137    161     14      0      0
3    ...         0     0     0     90    136     58    107     34     47
4    ...         0     0    22     26     73     96     42      9      0
5    ...         0     0     0      0      0      0      0      0      0
6    ...         0    61     0    113    116     96      0     14      0
7    ...         0    12     0      0    112    275    148    139     24
8    ...         0    29     0     24     48      0      0      0      0
9    ...        18     8     0     50     52    111    108    172    102

   R3500plus
0          0
1          0
2          0
3          0
4          0
5          0
6          0
7          0
8          0
9          0

[10 rows x 25 columns]
           Id2  Total_Bachelors  STEM  STEM_rel  Business  Education  \
```

```
0  60855001001              192    88         0         38          0
1  60855001002              573   368         0         98         14
2  60855001003              360   127        81         84         17
3  60855001004              710   387         0        122         10
4  60855002001              523   309        12         30         12
5  60855002002               64    13        11         31          0
6  60855002003              295   138        37         38         11
7  60855002004              964   543        28        153         13
8  60855003001               53    32         0          0          0
9  60855003002             1606   594       159        332         61

   Humanities
0          66
1          93
2          51
3         191
4         160
5           9
6          71
7         227
8          21
9         460
           Id2  Population
0  60855001001         932
1  60855001002        1936
2  60855001003        2620
3  60855001004        2336
4  60855002001        1568
5  60855002002         694
6  60855002003        1711
7  60855002004        2296
8  60855003001         553
9  60855003002        3237
```

In [6]:
```
tabs = dfs.values()
df = tabs.pop()
for x in tabs:
    df = df.merge(x, how='outer')

df.to_csv(r"C:\Data\CO_054\census_merged_2.csv", index=False)
```

In [110]:
```
rts = pd.read_csv(r"C:\Data\CO_054\MtVwResRCY.csv")
```

In [111]:
```
rts.columns = [x.lower() for x in rts.columns]
rts.head(2)
```

Out[111]:

	fid	loc_name	status	score	match_addr	pct_along	side	ref_id	re
0	569	CO_052_053_054	M	100.0	1097 MERCY ST	97.959184	L	23,708,145.00	23
1	1,648	NaN	M	0.0	NaN	0.000000	R	23,749,160.00	23

2 rows × 75 columns

In [11]:
```
check = rts.groupby(['ref_id']).apply(np.mode)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-11-627755d9e89f> in <module>()
----> 1 check = rts.groupby(['ref_id']).apply(np.mode)

AttributeError: 'module' object has no attribute 'mode'
```

In [112]:
```
rts_f1 = rts.loc[rts['z1tims'] < 2]
mxRt = rts_f1.groupby(['ref_id_side', 'route']).size().reset_index(name='coun
t')
mxRt = mxRt.sort_values(by=(['ref_id_side', 'count']))
mxRt = mxRt.drop_duplicates('ref_id_side', keep='last')
#count = mxRt['ref_id_side']
mxRt.head(5)
mxRt.to_csv(r"C:\Data\CO_054\maxRCY.csv", index=False)
type(mxRt)
```

Out[112]: pandas.core.frame.DataFrame

In [113]:
```
len(mxRt)
```

Out[113]: 2827

In [94]:
```
def get_mode(group):
    return group.mode()[0]
```

In [95]:
```
rts_f1 = rts.loc[rts['z1tims'] < 2]
merged = rts_f1.merge(rts_f1.groupby(['ref_id_side'])['route'].apply(get_mode)
.reset_index(), how='left', indicator=True)
merged[merged['_merge'] != 'both'].groupby(['ref_id_side', 'route']).size().to
_csv(r"C:\Data\CO_054\noMaxYW.csv")
```

In [96]:
```
fix = merged[merged['_merge'] != 'both'].groupby(['ref_id_side', 'route']).siz
e()
```

In [97]:
```
final = rts.merge(mxRts)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-97-1871867f1e91> in <module>()
----> 1 final = rts.merge(mxRts)

NameError: name 'mxRts' is not defined
```

In [9]:
```
# Group by BG ID and route to get the count of rount numbers in a block group

bgRoute = pd.read_csv(r"C:\Data\Waste_Intervention\Mountain_View\AllComs.csv",
 low_memory=False)
#create series that gets number of route stops in a BG, include route to do a
 join later as we will be using the route count
bg = bgRoute.groupby(['GEOID','Z1COMM','Route']).size().reset_index(name='coun
t')

# mkae a dataframe getting the count of stops per commodity in each BG
bgCom = bgRoute.groupby(['GEOID','Z1COMM']).size().reset_index(name='count')

# create a dateframe that gets a count of the route that can be joined back to
 the BG table
rt = bgRoute.groupby(['Route', 'Z1COMM']).size().reset_index(name='count')

bgRt = pd.merge(bg, rt, on = 'Route')
bgRtCom = pd.merge(bgRt, bgCom, left_on=(['GEOID','Z1COMM_x']), right_on=(['GE
OID', 'Z1COMM']))
```

In [10]:
```
bgRtCom.head()
```

Out[10]:

	GEOID	Z1COMM_x	Route	count_x	Z1COMM_y	count_y	Z1COMM	count
0	60855046011	G	303X	4	G	622	G	367
1	60855046011	G	404X	363	G	817	G	367
2	60855092023	G	303X	148	G	622	G	152
3	60855092023	G	305X	4	G	725	G	152
4	60855093031	G	303X	9	G	622	G	162

In [11]:
```
bgRtCom['perRt'] = bgRtCom['count_x'] / bgRtCom['count_y']
```

In [12]:
```
bgRtCom.head()
```

Out[12]:

	GEOID	Z1COMM_x	Route	count_x	Z1COMM_y	count_y	Z1COMM	count	
0	60855046011	G	303X	4	G	622	G	367	0.0
1	60855046011	G	404X	363	G	817	G	367	0.4
2	60855092023	G	303X	148	G	622	G	152	0.2
3	60855092023	G	305X	4	G	725	G	152	0.0
4	60855093031	G	303X	9	G	622	G	162	0.0

In [13]:
```
wt = pd.read_csv(r"C:\Data\Waste_Intervention\Mountain_View\Rts_Wts.csv")
wt.head()
```

Out[13]:

	Route	Tons
0	101X	4.470000
1	102X	6.063750
2	103X	8.566000
3	104X	7.159412
4	105X	7.731053

In [14]:
```
y_raw = pd.merge(bgRtCom, wt, on='Route')
```

In [15]:
```
y_raw['bgTons'] = y_raw['perRt'] * y_raw['Tons']
y_raw.head()
```

Out[15]:

	GEOID	Z1COMM_x	Route	count_x	Z1COMM_y	count_y	Z1COMM	count	
0	60855046011	G	303X	4	G	622	G	367	0.0
1	60855092023	G	303X	148	G	622	G	152	0.2
2	60855093031	G	303X	9	G	622	G	162	0.0
3	60855093041	G	303X	336	G	622	G	337	0.5
4	60855093042	G	303X	115	G	622	G	115	0.1

In [16]:
```
y_con = y_raw.groupby(['GEOID', 'Z1COMM_x'])[['bgTons']].sum().reset_index()
```

In [17]:
```
y_G = y_con[y_con['Z1COMM_x'] == 'G']
y_O = y_con[y_con['Z1COMM_x'] == 'O']
y_R = y_con[y_con['Z1COMM_x'] == 'R']
```

In [18]:
```
y_G.head()
```

Out[18]:

	GEOID	Z1COMM_x	bgTons
0	60855046011	G	3.417532
3	60855091051	G	6.038411
6	60855091052	G	2.820326
9	60855091053	G	3.001942
12	60855091081	G	1.776370

In [19]:
```
y_GO = pd.merge(y_G, y_O, on='GEOID')
```

In [20]:
```
y_GO.head()# y =  pd.merge(y_GO, y_R, on='GEOID')
```

Out[20]:

	GEOID	Z1COMM_x_x	bgTons_x	Z1COMM_x_y	bgTons_y
0	60855046011	G	3.417532	O	2.715642
1	60855091051	G	6.038411	O	3.142073
2	60855091052	G	2.820326	O	2.109063
3	60855091053	G	3.001942	O	1.826215
4	60855091081	G	1.776370	O	1.170674

In [21]:
```
 y =  pd.merge(y_GO, y_R, on='GEOID')
```

In [22]:
```
y.head()
```

Out[22]:

	GEOID	Z1COMM_x_x	bgTons_x	Z1COMM_x_y	bgTons_y	Z1COMM_x	bgTon
0	60855046011	G	3.417532	O	2.715642	R	1.41906
1	60855091051	G	6.038411	O	3.142073	R	3.06261
2	60855091052	G	2.820326	O	2.109063	R	1.79439
3	60855091053	G	3.001942	O	1.826215	R	1.10015
4	60855091081	G	1.776370	O	1.170674	R	0.48814

In [23]:
```
y['totalWaste'] = y['bgTons_x'] + y['bgTons_y'] + y['bgTons']
```

In [24]:
```
y.head()
```

Out[24]:

	GEOID	Z1COMM_x_x	bgTons_x	Z1COMM_x_y	bgTons_y	Z1COMM_x	bgTon
0	60855046011	G	3.417532	O	2.715642	R	1.41906
1	60855091051	G	6.038411	O	3.142073	R	3.06261
2	60855091052	G	2.820326	O	2.109063	R	1.79439
3	60855091053	G	3.001942	O	1.826215	R	1.10015
4	60855091081	G	1.776370	O	1.170674	R	0.48814

In [25]:
```
y['nonLF'] = y['totalWaste'] - y['bgTons_x']
```

In [26]:
```
y.head()
```

Out[26]:

	GEOID	Z1COMM_x_x	bgTons_x	Z1COMM_x_y	bgTons_y	Z1COMM_x	bgTon
0	60855046011	G	3.417532	O	2.715642	R	1.41906
1	60855091051	G	6.038411	O	3.142073	R	3.06261
2	60855091052	G	2.820326	O	2.109063	R	1.79439
3	60855091053	G	3.001942	O	1.826215	R	1.10015
4	60855091081	G	1.776370	O	1.170674	R	0.48814

In [27]:
```
y['per_NLF'] = y['nonLF']/y['totalWaste']
```

In [28]:
```
y.head()
```

Out[28]:

	GEOID	Z1COMM_x_x	bgTons_x	Z1COMM_x_y	bgTons_y	Z1COMM_x	bgTon
0	60855046011	G	3.417532	O	2.715642	R	1.41906
1	60855091051	G	6.038411	O	3.142073	R	3.06261
2	60855091052	G	2.820326	O	2.109063	R	1.79439
3	60855091053	G	3.001942	O	1.826215	R	1.10015
4	60855091081	G	1.776370	O	1.170674	R	0.48814

In [33]:
```
x = pd.read_csv(r"C:\Data\Waste_Intervention\Mountain_View\census_merged_3.csv")
x.head()
```

Out[33]:

	Id2	med_hh_inc	med_age	p_vacant	avg_hh_size	p_renters	p_r1000_plus
0	60855001001	59118.0	27.2	0.000000	3.10	0.673423	0.611888
1	60855001002	110714.0	33.7	0.000000	2.84	0.677642	0.468373
2	60855001003	62730.0	26.9	0.000000	4.67	0.618702	0.577540
3	60855001004	85648.0	34.1	0.100679	2.64	0.674658	0.533937
4	60855002001	103265.0	30.6	0.028517	2.71	0.570927	0.509506

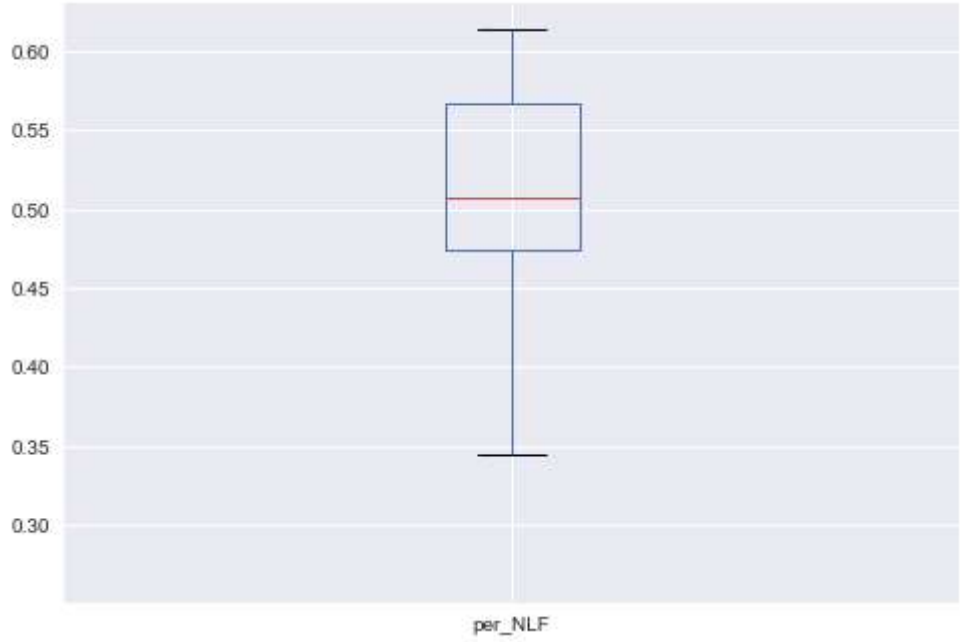
In [34]:
```
div_rate = pd.merge(x, y, right_on='GEOID', left_on='Id2')

div_rate.to_csv(r"C:\Data\Waste_Intervention\Mountain_View\div_rate_all.csv")
```

In [98]:
```
df = pd.read_csv(r"C:\Data\Waste_Intervention\Mountain_View\div_rate_all.csv")
df['per_NLF'].describe()
```

Out[98]:
```
count    56.000000
mean      0.509190
std       0.070802
min       0.268017
25%       0.474059
50%       0.506763
75%       0.566571
max       0.613565
Name: per_NLF, dtype: float64
```

In [100]:
```
df.boxplot('per_NLF')
```

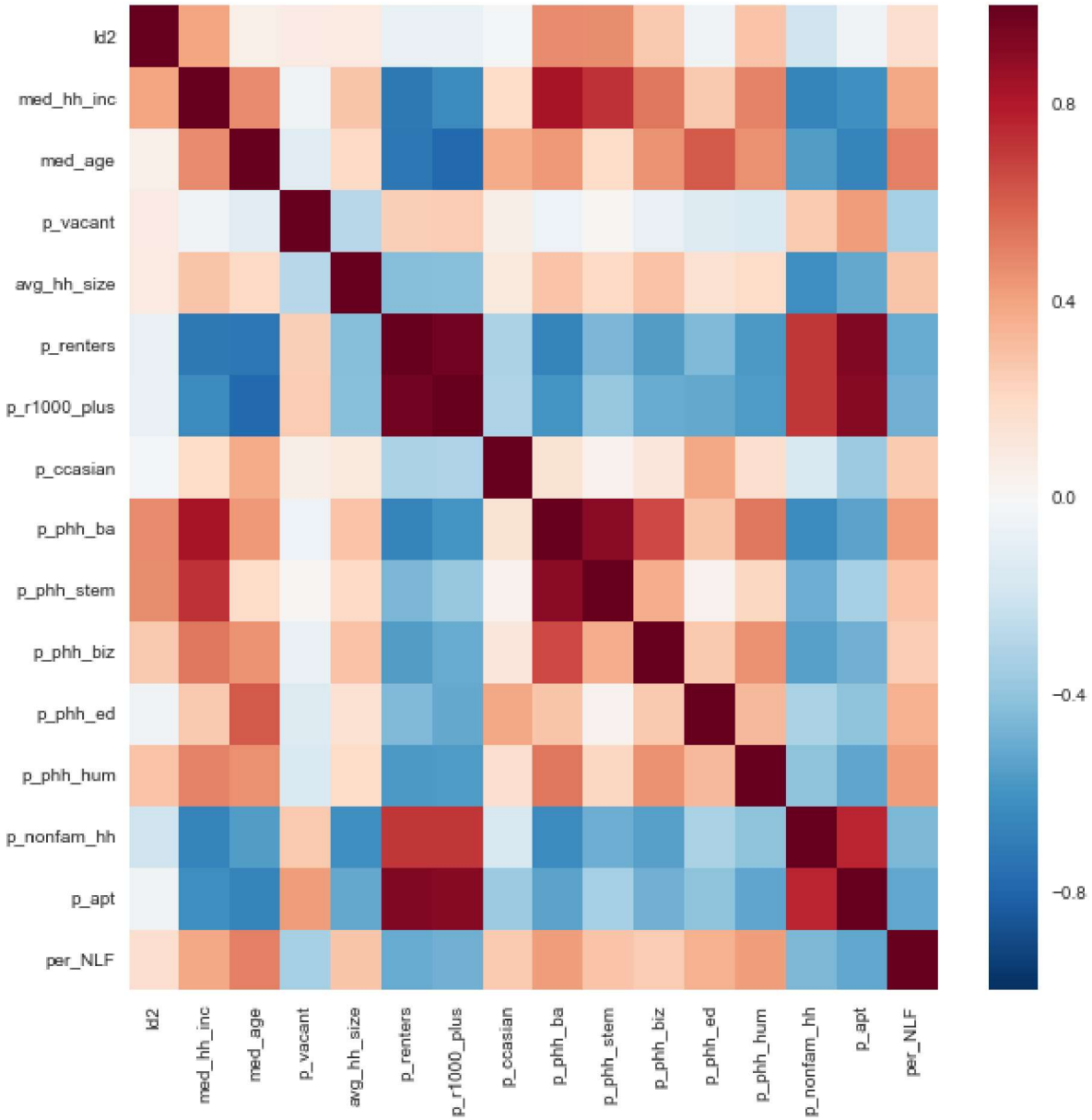
Out[100]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x25288e10>
```



In [47]:
```
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

import re

from sklearn.model_selection import cross_val_score
from sklearn import (
    metrics,
    linear_model,
    ensemble,
    neighbors,
)

from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn import feature_selection
```

In [49]:
```
fig, ax = plt.subplots(figsize=(10,10))

sns.heatmap(df.corr(), ax=ax)
```

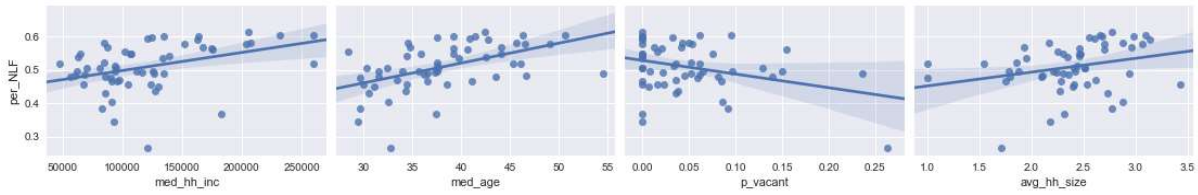
Out[49]: <matplotlib.axes._subplots.AxesSubplot at 0xd899438>



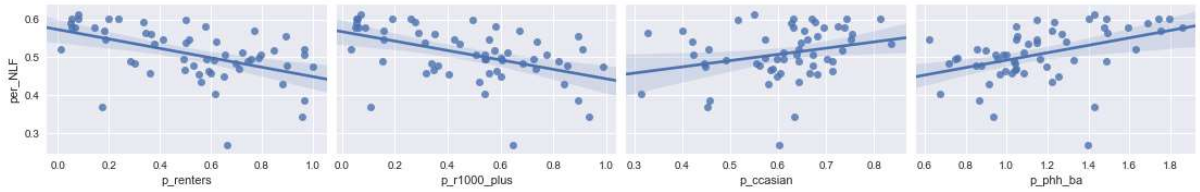
In [118]:
```
features_all = df.drop(['per_NLF', 'Id2'], axis=1)
target =  df['per_NLF']

X = features

y = target

# Initialize and fit the model
linreg = LinearRegression()
linreg.fit(X, y)

# Print the intercept and coefficients
print 'Intercept:', linreg.intercept_
print 'Coefficients:',linreg.coef_
print 'R-Squared:', linreg.score(X, y)
```

```
Intercept: 0.185102394096
Coefficients: [ -1.91479979e-07   4.60612136e-03  -2.67727530e-01   1.4898735
9e-02
  -9.87613773e-02   1.52087203e-01   6.34688744e-02   1.89259083e+07
  -1.89259083e+07  -1.89259084e+07  -1.89259082e+07  -1.89259080e+07
  -1.64076285e-02  -1.72521845e-02]
R-Squared: 0.477963521181
```

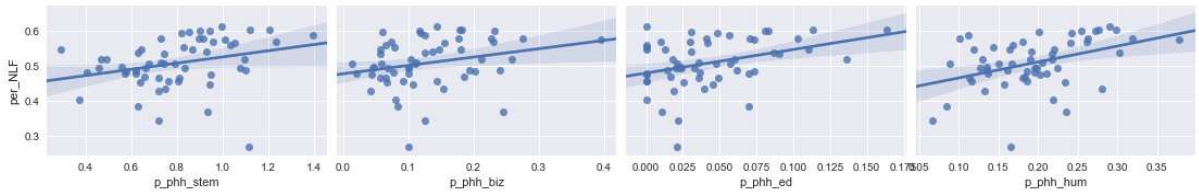
In [103]:
```
import matplotlib.pyplot as plt
%matplotlib inline
plt.style.use('seaborn')


g = sns.pairplot(df, x_vars=['med_hh_inc', 'med_age', 'p_vacant', 'avg_hh_siz
e'], y_vars='per_NLF',  size=15, aspect=0.7, kind='reg')

g.fig.set_size_inches(15,2)
```



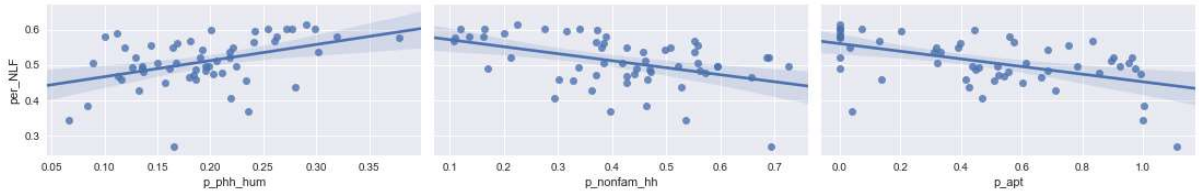
In [104]:
```
g = sns.pairplot(df, x_vars=['p_renters', 'p_r1000_plus', 'p_ccasian', 'p_phh_
ba'], y_vars='per_NLF',  size=15, aspect=0.7, kind='reg')

g.fig.set_size_inches(15,2)
```



In [107]:
```
g = sns.pairplot(df, x_vars=['p_phh_stem',
       'p_phh_biz', 'p_phh_ed', 'p_phh_hum'], y_vars='per_NLF',  size=15, aspe
ct=0.7, kind='reg')

g.fig.set_size_inches(15,2)
```



In [106]:
```
g = sns.pairplot(df, x_vars=[ 'p_phh_hum', 'p_nonfam_hh', 'p_apt'], y_vars='pe
r_NLF',  size=15, aspect=0.7, kind='reg')

g.fig.set_size_inches(15,2)
```



In [131]:
```
from sklearn.ensemble import RandomForestRegressor

cls = RandomForestRegressor(n_estimators=50)
cls.fit(X,y)

print cls.score(X,y)
print cross_val_score(cls, X, y )
```

```
0.925335251199
[-0.30020396 -0.40860798 -4.77100593]
```

In [138]:
```
feature_importances = cls.feature_importances_
f_zip = zip(features, feature_importances)
```

In [139]:
```
for i in f_zip:
    print i
```

```
('med_hh_inc', 0.021833734605622643)
('med_age', 0.14188542782600144)
('p_vacant', 0.029195688081678238)
('avg_hh_size', 0.06336913162749272)
('p_renters', 0.028119084295113086)
('p_r1000_plus', 0.062194992132091301)
('p_ccasian', 0.041866644171379877)
('p_phh_ba', 0.094887654359521734)
('p_phh_stem', 0.049112841478810759)
('p_phh_biz', 0.012325230754034138)
('p_phh_ed', 0.021225180313106849)
('p_phh_hum', 0.10680549438436307)
('p_nonfam_hh', 0.030785699807264168)
('p_apt', 0.29639319616352)
```

In [ ]: