In [127]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
%matplotlib inline
df = pd.read_csv('../../Documents/Python Scripts/data_flats.csv',sep=';')
```

In [128]:
```
df.info()
```

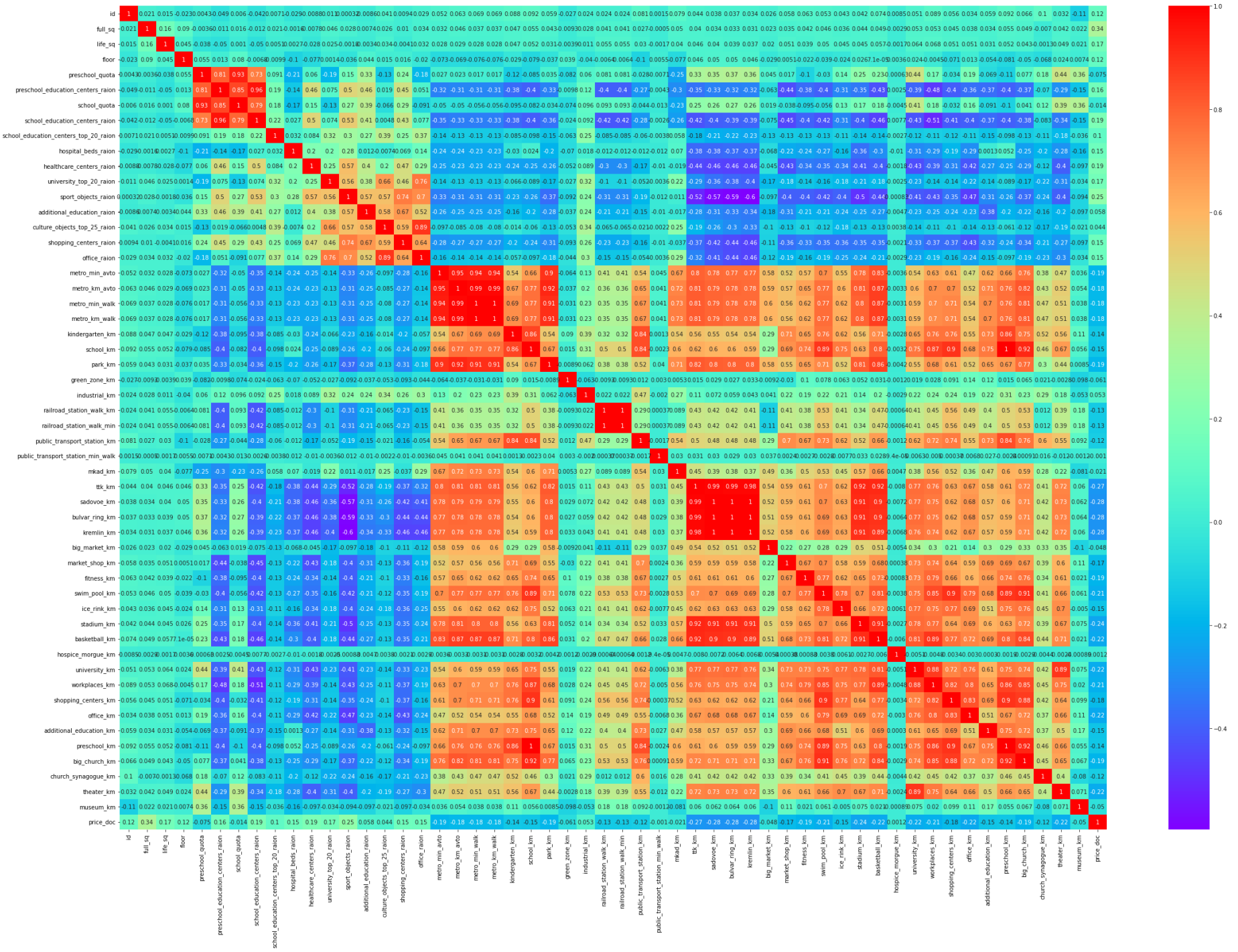
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30469 entries, 0 to 30468
Data columns (total 56 columns):
 #   Column                                Non-Null Count  Dtype  
---  ------                                --------------  -----  
 0   id                                    30469 non-null  int64  
 1   full_sq                               30469 non-null  int64  
 2   life_sq                               24086 non-null  float64
 3   floor                                 30302 non-null  float64
 4   sub_area                              30469 non-null  object 
 5   preschool_quota                       23781 non-null  float64
 6   preschool_education_centers_raion     30469 non-null  int64  
 7   school_quota                          23784 non-null  float64
 8   school_education_centers_raion        30469 non-null  int64  
 9   school_education_centers_top_20_raion 30469 non-null  int64  
 10  hospital_beds_raion                   16029 non-null  float64
 11  healthcare_centers_raion              30469 non-null  int64  
 12  university_top_20_raion               30469 non-null  int64  
 13  sport_objects_raion                   30469 non-null  int64  
 14  additional_education_raion            30469 non-null  int64  
 15  culture_objects_top_25_raion          30469 non-null  int64  
 16  shopping_centers_raion                30469 non-null  int64  
 17  office_raion                          30469 non-null  int64  
 18  metro_min_avto                        30469 non-null  float64
 19  metro_km_avto                         30469 non-null  float64
 20  metro_min_walk                        30444 non-null  float64
 21  metro_km_walk                         30444 non-null  float64
 22  kindergarten_km                       30469 non-null  float64
 23  school_km                             30469 non-null  float64
 24  park_km                               30469 non-null  float64
 25  green_zone_km                         30469 non-null  float64
 26  industrial_km                         30469 non-null  float64
 27  railroad_station_walk_km              30444 non-null  float64
 28  railroad_station_walk_min             30444 non-null  float64
 29  public_transport_station_km           30469 non-null  float64
 30  public_transport_station_min_walk     30469 non-null  float64
 31  mkad_km                               30469 non-null  float64
 32  ttk_km                                30469 non-null  float64
 33  sadovoe_km                            30469 non-null  float64
 34  bulvar_ring_km                        30469 non-null  float64
 35  kremlin_km                            30469 non-null  float64
 36  big_market_km                         30469 non-null  float64
 37  market_shop_km                        30469 non-null  float64
 38  fitness_km                            30469 non-null  float64
 39  swim_pool_km                          30469 non-null  float64
 40  ice_rink_km                           30469 non-null  float64
 41  stadium_km                            30469 non-null  float64
 42  basketball_km                         30469 non-null  float64
 43  hospice_morgue_km                     30469 non-null  float64
 44  university_km                         30469 non-null  float64
 45  workplaces_km                         30469 non-null  float64
 46  shopping_centers_km                   30469 non-null  float64
 47  office_km                             30469 non-null  float64
 48  additional_education_km               30469 non-null  float64
 49  preschool_km                          30469 non-null  float64
 50  big_church_km                         30469 non-null  float64
 51  church_synagogue_km                   30469 non-null  float64
 52  theater_km                            30469 non-null  float64
 53  museum_km                             30469 non-null  float64
 54  ecology                               30469 non-null  object 
 55  price_doc                             30469 non-null  int64  
dtypes: float64(41), int64(13), object(2)
memory usage: 13.0+ MB
```

In [23]:
```
df['sub_area']
```

Out[23]:
```
0                  Bibirevo
1           Nagatinskij Zaton
2              Tekstil'shhiki
3                      Mitino
4                   Basmannoe
                 ...
30464                Otradnoe
30465                Tverskoe
30466    Poselenie Vnukovskoe
30467             Obruchevskoe
30468              Novogireevo
Name: sub_area, Length: 30469, dtype: object
```

In [24]:
```
correlation = df.corr()
fig = plt.figure()
axes = fig.add_axes([0,0,5,5])
sns.heatmap(correlation, annot=True, cmap='rainbow')
```

Out[24]:
```
<Axes:>
```



In [129]:
```
df1 = df[['full_sq','life_sq','sport_objects_raion','kremlin_km','healthcare_centers_raion',
          'school_education_centers_raion','preschool_education_centers_raion','hospital_beds_raion',
          'university_top_20_raion','shopping_centers_raion','metro_min_walk','ecology', 'price_doc']].copy()
```

In [130]:
```
df1.loc[df1['ecology'] == 'no data', 'ecology'] = np.nan
df1 = df1.loc[df1['life_sq'] > 9]
df1 = df1.loc[df1['full_sq'] > 9]
```

In [131]:
```
df1['ecology'].value_counts()
```

Out[131]:
```
poor            7669
good            5553
excellent       3370
satisfactory    3295
Name: ecology, dtype: int64
```

In [132]:
```
df1['ecology'] = df1['ecology'].astype('category')
df1['ecology'] = df1['ecology'].cat.reorder_categories(['poor', 'satisfactory', 'good', 'excellent'], ordered=True)
df1['ecology'] = df1['ecology'].cat.codes
```

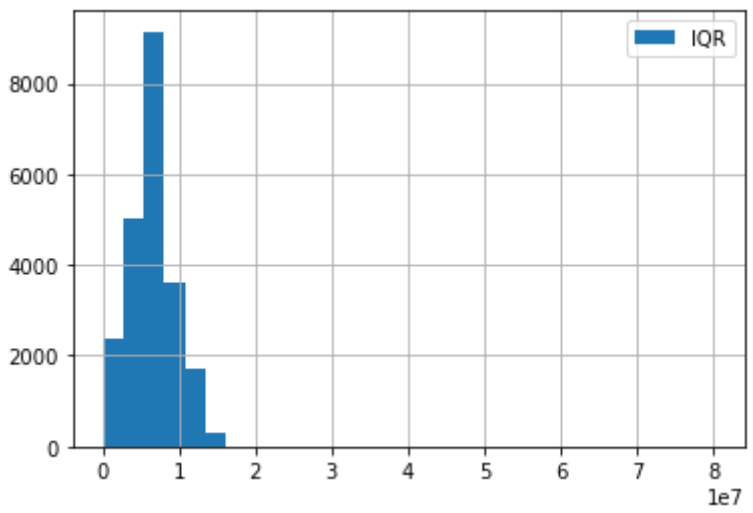
In [133]:
```
df1=df1.fillna(df1.median())
```

In [134]:
```
median = df1.price_doc.median()
print(median)
IQR = df1.price_doc.quantile(0.75, interpolation='midpoint') - df1.price_doc.quantile(0.25, interpolation='midpoint'
perc25 = df1.price_doc.quantile(0.25, interpolation='midpoint')
perc75 = df1.price_doc.quantile(0.75, interpolation='midpoint')
print('25-й перцентиль:{},'.format(perc25),
      '75-й перцентиль: {},'.format(perc75),
      "IQR: {}, ".format(IQR),"Гарницы выбросов: [{f}, {l}].".format(f=perc25 - 1.5*IQR,
                                                                    l=perc75+1.5*IQR))

df1.price_doc.loc[df1.price_doc.between(perc25-1.5*IQR,
                                        perc75+1.5*IQR)].hist(bins=30,
                                                              range=(1e+5,8e+7),
                                                              label='IQR')
plt.legend()
df1 = df1.loc[df1.price_doc.between(perc25-1.5*IQR,
                                    perc75+1.5*IQR)]
df1.price_doc.describe()
```

```
6500000.0
25-й перцентиль:5000000.0, 75-й перцентиль: 8650000.0, IQR: 3650000.0,  Гарницы выбросов: [-475000.0, 14125000.0].
```

Out[134]:
```
count    2.218300e+04
mean     6.502109e+06
std      2.883633e+06
min      1.000000e+05
25%      4.900000e+06
50%      6.300000e+06
75%      8.103500e+06
max      1.410360e+07
Name: price_doc, dtype: float64
```



In [135]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics
```

In [138]:
```
X = df1.iloc[:,0:12].values
Y = df1.iloc[:,12].values
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.25)
my_model = LinearRegression()
my_model.fit(X_train, Y_train)
y_pred = my_model.predict(X_test)
#print(my_model.intercept_,my_model.coef_)
print('MAE:', metrics.mean_absolute_error(Y_test,y_pred))
print('MSE:', metrics.mean_squared_error(Y_test, y_pred))
print('R_2:', metrics.r2_score(Y_test,y_pred))
```

```
MAE: 2017935.5928098806
MSE: 7266508037672.396
R_2: 0.1299483126075065
```

In [116]:
```
1 df1.describe()
```

Out[116]:

	full_sq	life_sq	sport_objects_raion	kremlin_km	healthcare_centers_raion	school_education_centers_raion	preschool_education
count	22183.000000	22183.000000	22183.000000	22183.000000	22183.000000	22183.000000	
mean	50.354370	33.289501	7.170761	15.141552	1.508948	5.293197	
std	40.313478	53.348209	5.914148	7.941736	1.494708	3.322322	
min	10.000000	10.000000	0.000000	0.072897	0.000000	0.000000	
25%	38.000000	20.000000	3.000000	10.102446	0.000000	3.000000	
50%	45.000000	30.000000	6.000000	13.610218	1.000000	5.000000	
75%	59.000000	41.000000	10.000000	18.017726	3.000000	7.000000	
max	5326.000000	7478.000000	29.000000	70.738769	6.000000	14.000000	