Exploratory Data Analysis

This notebook focuses the EDA on sold condos and the data scraped/proccessed from Redfin.

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt

color = sns.color_palette()
%matplotlib inline
```

In [2]:
```
def price_vs_house_feature(df, feature, subtitle, ax):
    sns.regplot(x=df[feature], y=df.LOG_SOLDPRICE, scatter_kws={'alpha':0.2}, color=color[0], ax=ax)
    # set titles
    ax.set_title("Log Sold Price vs. " + feature + "\n" + subtitle, fontsize=16)
    # label axes
    ax.set_ylabel("Log Sold Price", fontsize=14)
    ax.set_xlabel(feature, fontsize=14)

def dom_vs_house_feature(df, feature, subtitle, ax):
    sns.regplot(x=df[feature], y=df.LOG_DOM, scatter_kws={'alpha':0.2}, color=color[0], ax=ax)
    # set titles
    ax.set_title("Log DOM vs. " + feature + "\n" + subtitle, fontsize=16)
    # label axes
    ax.set_ylabel("Log DOM", fontsize=14)
    ax.set_xlabel(feature, fontsize=14)

def plot_sold_price_distribution(df, subtitle):
    fig, ax = plt.subplots(1, 2, figsize=(14, 5))
    sns.distplot(df.SOLDPRICE, kde=False, ax=ax[0])
    sns.distplot(df.LOG_SOLDPRICE, kde=False, ax=ax[1])

    ax[0].set_title("Distribution of Sold Price \n" + subtitle, fontsize=16)
    ax[1].set_title("Distribution of Log Sold Price \n" + subtitle, fontsize=16)

    ax[0].set_xlabel("Sold Price", fontsize=14)
    ax[1].set_xlabel("Log Sold Price", fontsize=14)
    ax[0].set_ylabel("Frequency", fontsize=14)
    ax[1].set_ylabel("Frequency", fontsize=14)

def plot_dom_distribution(df, subtitle):
    fig, ax = plt.subplots(1, 2, figsize=(14, 5))
    sns.distplot(df.DOM, kde=False, ax=ax[0])
    sns.distplot(df.LOG_DOM, kde=False, ax=ax[1])

    ax[0].set_title("Distribution of DOM \n" + subtitle, fontsize=16)
    ax[1].set_title("Distribution of Log (Log DOM) \n" + subtitle, fontsize=16)

    ax[0].set_xlabel("DOM", fontsize=14)
    ax[1].set_xlabel("Log DOM", fontsize=14)
    ax[0].set_ylabel("Frequency", fontsize=14)
    ax[1].set_ylabel("Frequency", fontsize=14)

```

In [3]:
```
df_con = pd.read_csv("data/features/CON_feats_no_img.csv", index_col=0)
df_con = df_con.fillna(-1)
df_con['LOG_SOLDPRICE'] = np.log(df_con['SOLDPRICE'])
df_con['LOG_DOM'] = np.log(np.log(df_con['DOM']+20))
df_con['log_sqft_finished'] = np.log(df_con['sqft_finished']+2)
df_con['log_sqft_unfinished'] = np.log(df_con['sqft_unfinished']+2)
df_con.head()
```

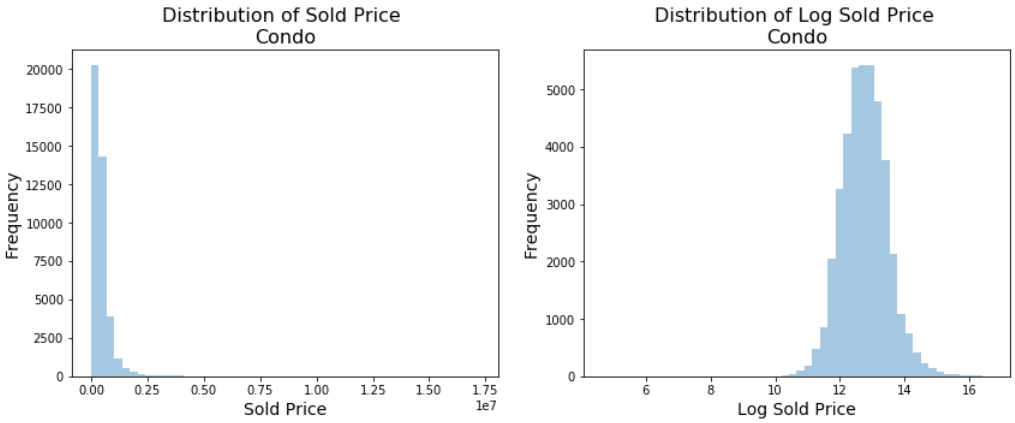
Out[3]:

	MLSNUM	LISTPRICE	SOLDPRICE	DOM	DTO	AGE	LOTSIZE	GARAGE	LISTMONTH	SOLDMONTH	...	school_distances_high_min	walk_score	transit_score	bike_score	nu
0	71498924	169900.0	177500.0	709	618.0	11.0	-1.0	2.0	3	1	...	0.1	34.0	-1.0	-1.0	
1	71905628	242000.0	235000.0	91	81.0	14.0	-1.0	1.0	9	1	...	0.3	31.0	-1.0	-1.0	
2	71918879	209000.0	209000.0	78	37.0	32.0	-1.0	0.0	10	1	...	0.5	14.0	-1.0	-1.0	
3	71952614	339900.0	350695.0	1	1.0	3.0	-1.0	1.0	1	1	...	0.9	17.0	-1.0	-1.0	
4	71912071	299900.0	280000.0	83	71.0	10.0	-1.0	1.0	9	2	...	0.9	16.0	-1.0	-1.0	

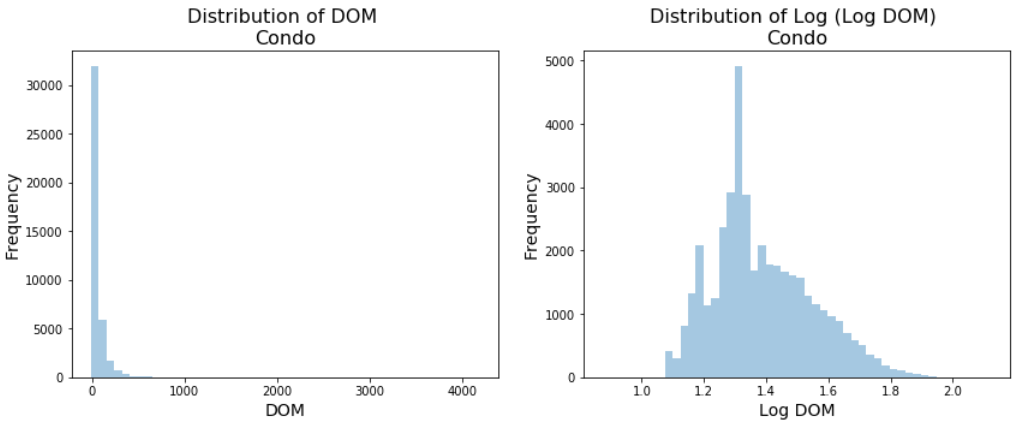
5 rows × 42 columns

Response Distribution

In [4]:
```
1 plot_sold_price_distribution(df_con, "Condo")
```



In [5]:
```
1 plot_dom_distribution(df_con, "Condo")
```

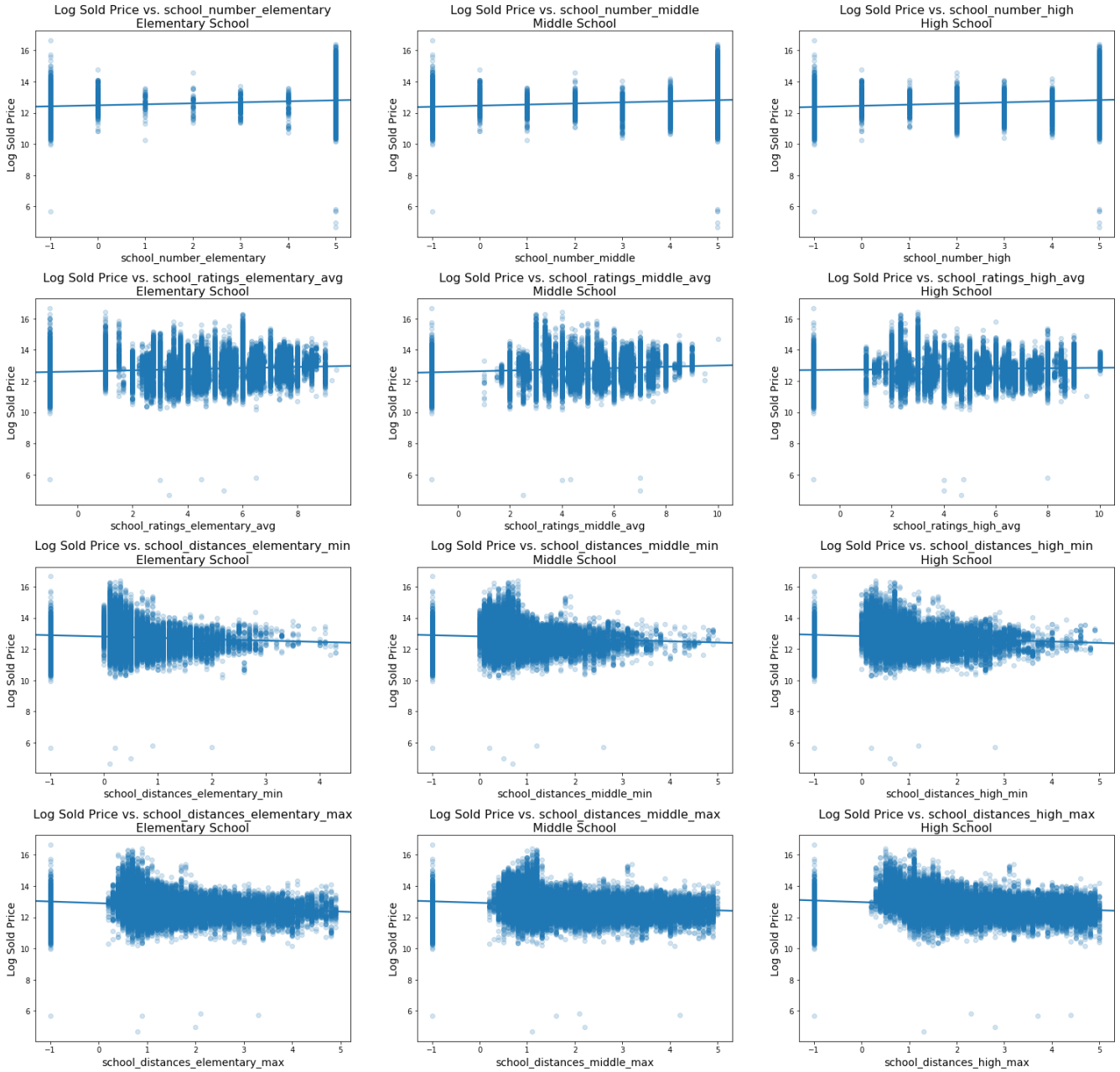


Price Trends

Price vs. School information

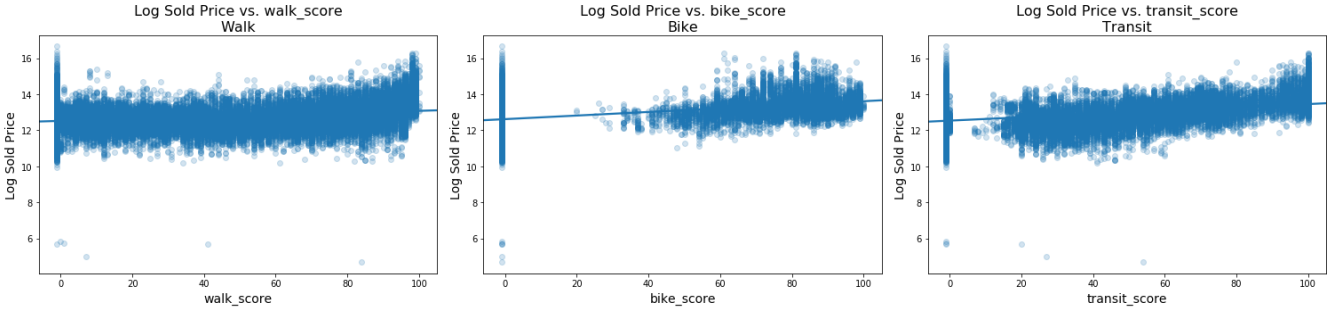
In [6]:

```
# price vs school information
fig, ax = plt.subplots(4, 3, figsize=(21, 20))
# number of schools
price_vs_house_feature(df_con, 'school_number_elementary', 'Elementary School', ax=ax[0][0])
price_vs_house_feature(df_con, 'school_number_middle', 'Middle School', ax=ax[0][1])
price_vs_house_feature(df_con, 'school_number_high', 'High School', ax=ax[0][2])

# average school rating
price_vs_house_feature(df_con, 'school_ratings_elementary_avg', 'Elementary School', ax=ax[1][0])
price_vs_house_feature(df_con, 'school_ratings_middle_avg', 'Middle School', ax=ax[1][1])
price_vs_house_feature(df_con, 'school_ratings_high_avg', 'High School', ax=ax[1][2])

# min school distance
price_vs_house_feature(df_con, 'school_distances_elementary_min', 'Elementary School', ax=ax[2][0])
price_vs_house_feature(df_con, 'school_distances_middle_min', 'Middle School', ax=ax[2][1])
price_vs_house_feature(df_con, 'school_distances_high_min', 'High School', ax=ax[2][2])

# max school distance
price_vs_house_feature(df_con, 'school_distances_elementary_max', 'Elementary School', ax=ax[3][0])
price_vs_house_feature(df_con, 'school_distances_middle_max', 'Middle School', ax=ax[3][1])
price_vs_house_feature(df_con, 'school_distances_high_max', 'High School', ax=ax[3][2])
plt.tight_layout()
```



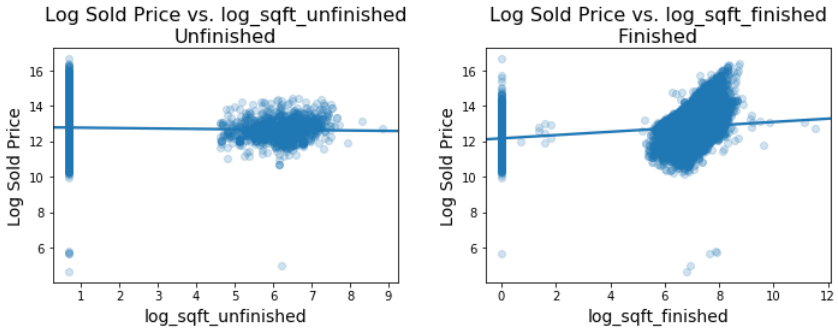
Price vs. Convenience scores

In [7]:
```
# price vs convenience scores
fig, ax = plt.subplots(1, 3, figsize=(21, 5))
# number of schools
price_vs_house_feature(df_con, 'walk_score', 'Walk', ax=ax[0])
price_vs_house_feature(df_con, 'bike_score', 'Bike', ax=ax[1])
price_vs_house_feature(df_con, 'transit_score', 'Transit', ax=ax[2])

plt.tight_layout()
```



Price vs. Square footage

In [8]:
```
# price vs sqft
fig, ax = plt.subplots(1, 2, figsize=(10, 4))
# number of schools
price_vs_house_feature(df_con, 'log_sqft_unfinished', 'Unfinished', ax=ax[0])
price_vs_house_feature(df_con, 'log_sqft_finished', 'Finished', ax=ax[1])

plt.tight_layout()
```



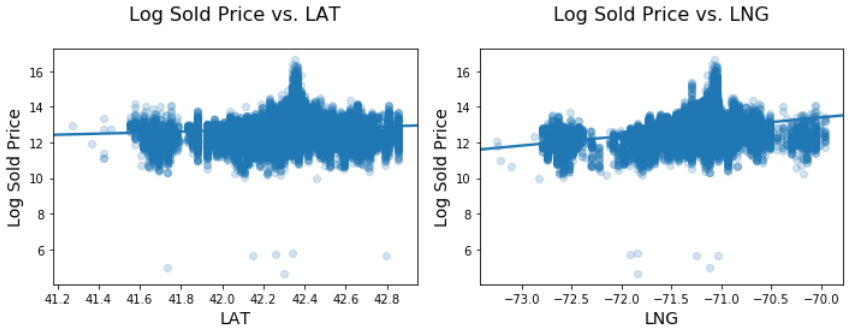
Price vs. number of photos on redfin

In [9]:
```
# price vs number of images posted on redfin
fig, ax = plt.subplots(1, 1, figsize=(5, 4))
# number of schools
price_vs_house_feature(df_con, 'num_photo', '', ax=ax)

plt.tight_layout()
```

Price vs. latitude/longitude

In [10]:

```
# price vs latitude/longitude
fig, ax = plt.subplots(1, 2, figsize=(10, 4))
# number of schools
price_vs_house_feature(df_con, 'LAT', '', ax=ax[0])
price_vs_house_feature(df_con, 'LNG', '', ax=ax[1])

plt.tight_layout()
```

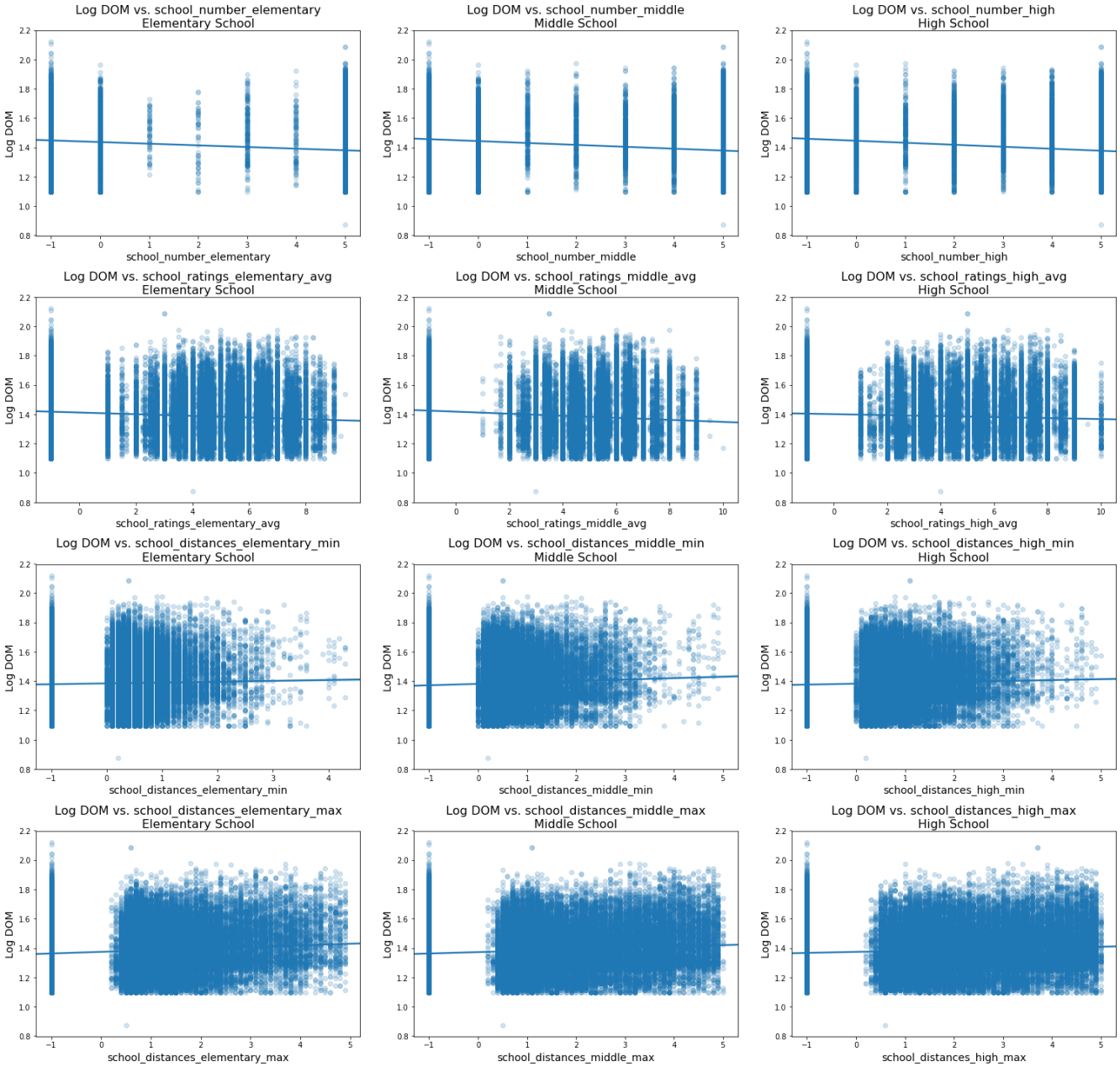


DOM Trend

DOM vs. School information

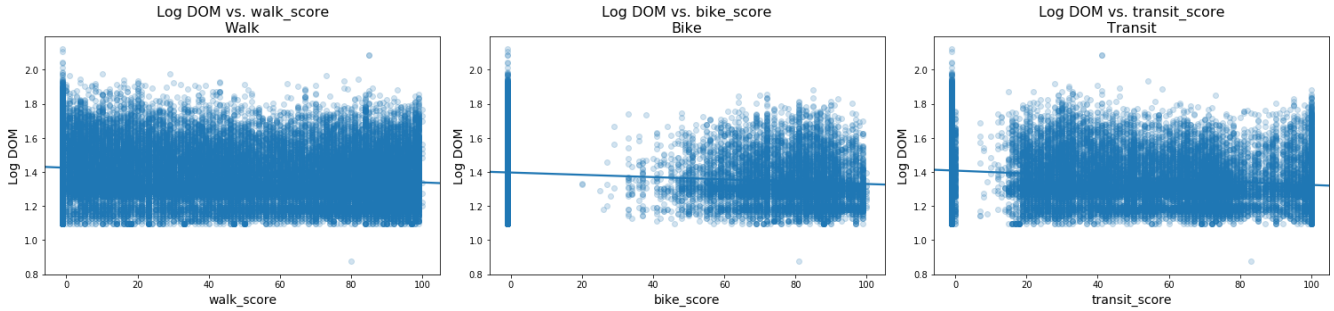
In [11]:

```
# dom vs school information
fig, ax = plt.subplots(4, 3, figsize=(21, 20))
# number of schools
dom_vs_house_feature(df_con, 'school_number_elementary', 'Elementary School', ax=ax[0][0])
dom_vs_house_feature(df_con, 'school_number_middle', 'Middle School', ax=ax[0][1])
dom_vs_house_feature(df_con, 'school_number_high', 'High School', ax=ax[0][2])

# average school rating
dom_vs_house_feature(df_con, 'school_ratings_elementary_avg', 'Elementary School', ax=ax[1][0])
dom_vs_house_feature(df_con, 'school_ratings_middle_avg', 'Middle School', ax=ax[1][1])
dom_vs_house_feature(df_con, 'school_ratings_high_avg', 'High School', ax=ax[1][2])

# min school distance
dom_vs_house_feature(df_con, 'school_distances_elementary_min', 'Elementary School', ax=ax[2][0])
dom_vs_house_feature(df_con, 'school_distances_middle_min', 'Middle School', ax=ax[2][1])
dom_vs_house_feature(df_con, 'school_distances_high_min', 'High School', ax=ax[2][2])

# max school distance
dom_vs_house_feature(df_con, 'school_distances_elementary_max', 'Elementary School', ax=ax[3][0])
dom_vs_house_feature(df_con, 'school_distances_middle_max', 'Middle School', ax=ax[3][1])
dom_vs_house_feature(df_con, 'school_distances_high_max', 'High School', ax=ax[3][2])
plt.tight_layout()
```



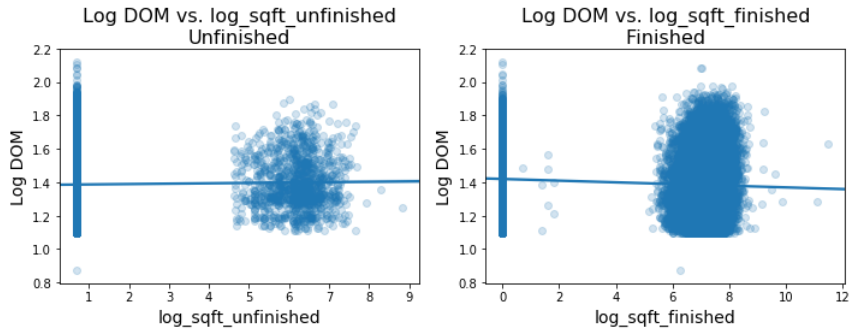
DOM vs. convenience scores

In [12]:
```
# dom vs convenience scores
fig, ax = plt.subplots(1, 3, figsize=(21, 5))
# number of schools
dom_vs_house_feature(df_con, 'walk_score', 'Walk', ax=ax[0])
dom_vs_house_feature(df_con, 'bike_score', 'Bike', ax=ax[1])
dom_vs_house_feature(df_con, 'transit_score', 'Transit', ax=ax[2])

plt.tight_layout()
```



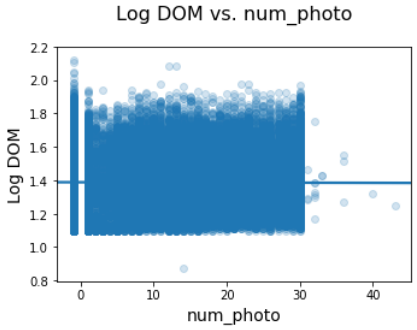
DOM vs. Square Footage

In [13]:
```
# dom vs sqft
fig, ax = plt.subplots(1, 2, figsize=(10, 4))
# number of schools
dom_vs_house_feature(df_con, 'log_sqft_unfinished', 'Unfinished', ax=ax[0])
dom_vs_house_feature(df_con, 'log_sqft_finished', 'Finished', ax=ax[1])

plt.tight_layout()
```

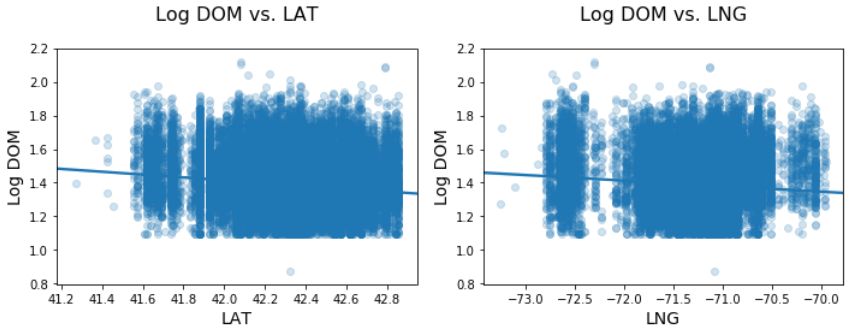


DOM vs. number of photos on redfin

In [14]:
```
# dom vs number of images posted on redfin
fig, ax = plt.subplots(1, 1, figsize=(5, 4))
# number of schools
dom_vs_house_feature(df_con, 'num_photo', '', ax=ax)

plt.tight_layout()
```



In [15]:
```
# price vs latitude/longitude
fig, ax = plt.subplots(1, 2, figsize=(10, 4))
# number of schools
dom_vs_house_feature(df_con, 'LAT', '', ax=ax[0])
dom_vs_house_feature(df_con, 'LNG', '', ax=ax[1])

plt.tight_layout()
```



In [ ]:
```

```