Freddy-2021 / **dsc-phase-2-project-v2-3** Public

forked from learn-co-curriculum/dsc-phase-2-project-v2-3

View license

0 stars 71 forks

Star Watch

Code Pull requests Actions Projects Wiki Security Insights

main

This branch is 1 commit ahead of learn-co-curriculum:main. Contribute Fetch upstream

Freddy-2021 phase 2 project ... 28 minutes ago 17

View code

README.md

Author: Freddy Abrahamson

Overview

For this project, I will use multiple linear regression modeling to analyze house sales in King County, in Washington state.

Business Problem

The goal of this project is to to provide advice to homeowners about how home renovations may increase the value of their homes, and by what amount. The information for this project is derived from data comprised of the different characteristics of over 20,000 homes in King County,which is located in Washington State. I will use this information gain a better understanding about how different remodels, or renovations to the homes listed, impact their price.

Data Understanding

The data comes from the King County House Sales dataset, in the form of a 'csv' file. The file will be converted into a pandas dataframe. It contains information about the different characteristics of the homes in the King County area,including the number of bedrooms, building grades, square footage, and price. King County is located in Washington State, and has a size of approximately 2300 square miles, per the U.S Census Bureau:

kc_house_data.csv

I will be giving this dataframe a brief overview of its different characteristics, with a view toward using its columns as variables in a regression model. These include:

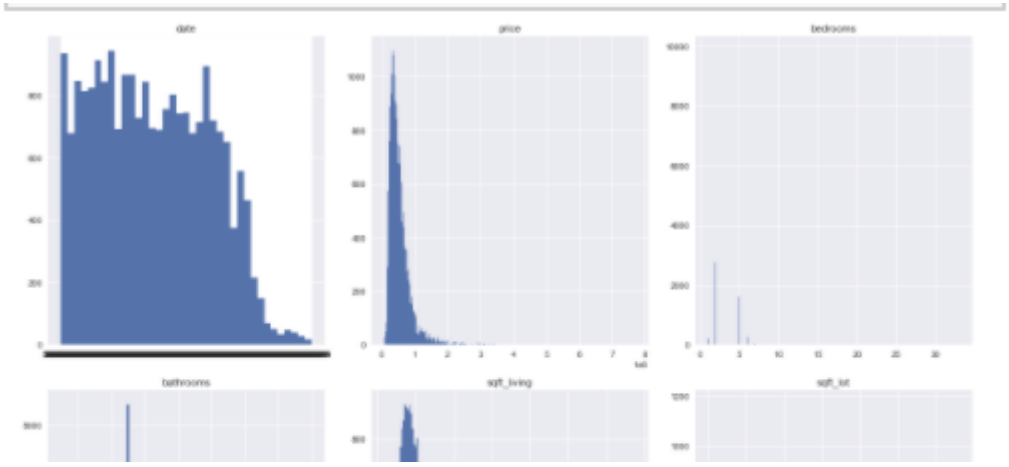
- dataframe shape: the number of rows and columns in the dataframe
- any missing/null values
- continuous variables
- categorical variables
- binary variables
- zero inflated variables
- outliers

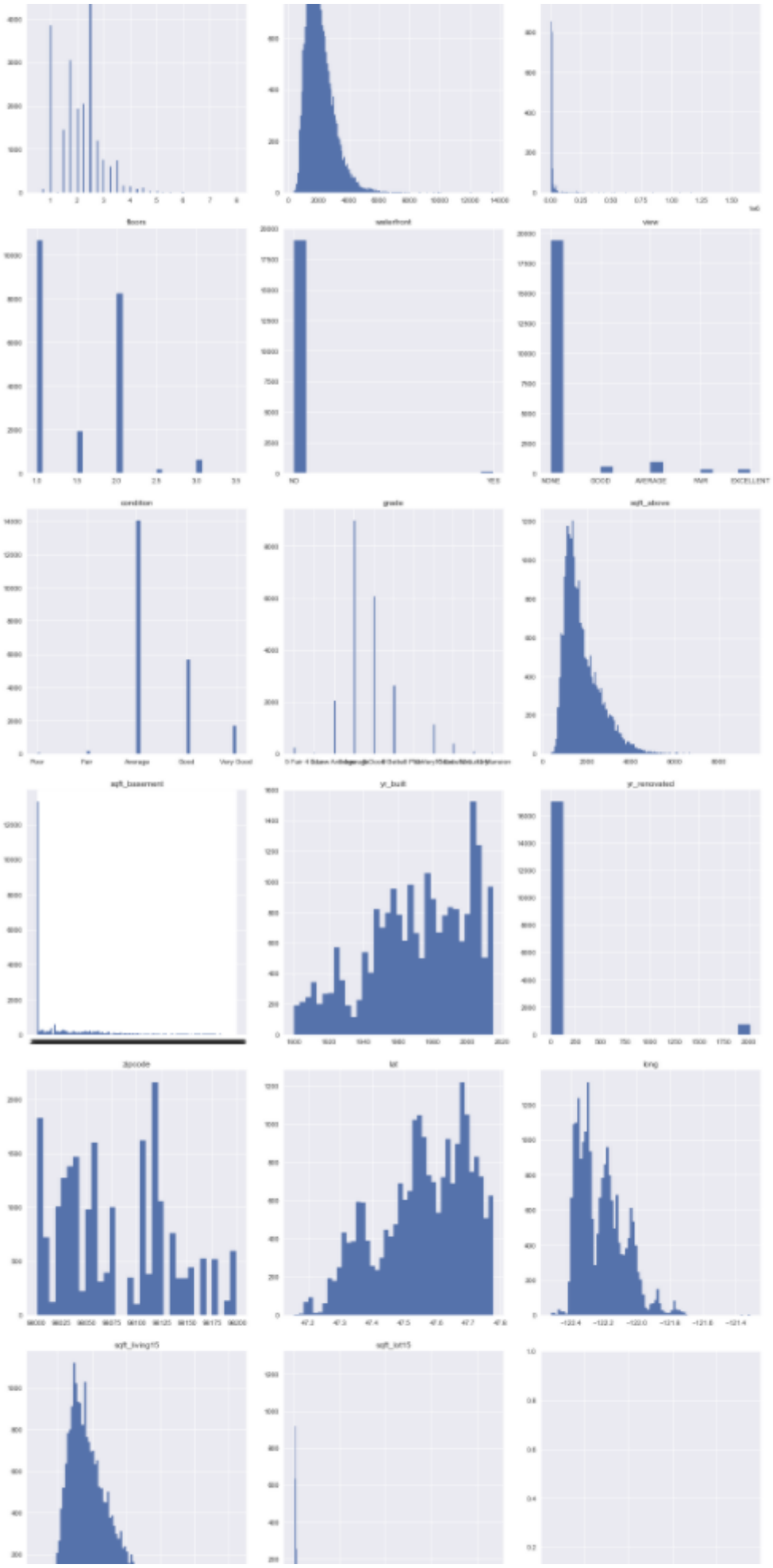
Since the goal is to try to gain insights, as to how much a particular upgrade or remodel can the impact the price of a house, as opposed to predicting home prices, I will be placing an emphasis on choosing features with the least explanatory overlap. To that end, for instance, I would favor a feature such as a bedroom, or a bathroom over square footage.

The following includes dataframe information, as well as plots and histograms describing the data:

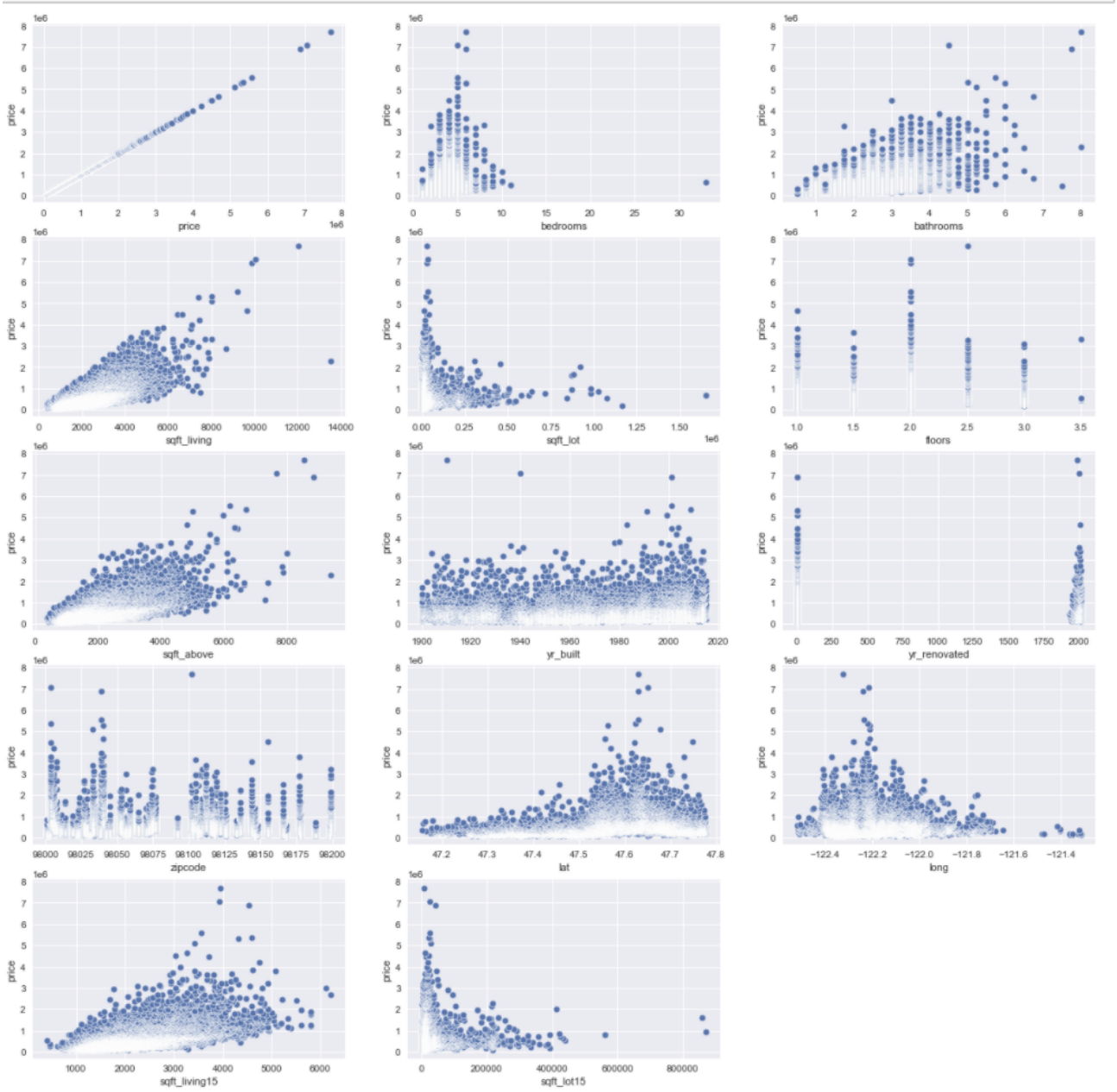
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 21597 entries, 15279 to 7245
Data columns (total 20 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   date           21597 non-null  object 
 1   price          21597 non-null  float64
 2   bedrooms       21597 non-null  int64  
 3   bathrooms      21597 non-null  float64
 4   sqft_living    21597 non-null  int64  
 5   sqft_lot       21597 non-null  int64  
 6   floors         21597 non-null  float64
 7   waterfront     19221 non-null  object 
 8   view           21534 non-null  object 
 9   condition      21597 non-null  object 
 10  grade          21597 non-null  object 
 11  sqft_above     21597 non-null  int64  
 12  sqft_basement  21597 non-null  object 
 13  yr_built       21597 non-null  int64  
 14  yr_renovated   17755 non-null  float64
 15  zipcode        21597 non-null  int64  
 16  lat            21597 non-null  float64
 17  long           21597 non-null  float64
 18  sqft_living15  21597 non-null  int64  
 19  sqft_lot15     21597 non-null  int64  
dtypes: float64(6), int64(8), object(6)
memory usage: 3.5+ MB
```

Above is information about the dataframe (after 'id' column was dropped).




floors
waterfront
view
NO
YES
NONE
GOOD
AVERAGE
FAIR
EXCELLENT
condition
grade
sqft_above
sqft_basement
yr_built
yr_renovated
zipcode
lat
long
sqft_living15
sqft_lot15

Above are the histograms of the distribtutions of the different features.



Above are scatter plots of the change of the different features, relative to the price.

Data Understanding Take-aways So Far:

1. There appear to be six categorical variables:

- bedrooms
- bathrooms

- floors
- view
- condition
- grade

2. There is atleast one binary variable: - waterfront
3. There appear to be several potential zero inflated variables: - sqft_lot - sqft_basement - yr_renovated - sqft_lot15
4. There are also several right-skewed distributions, that may be normalized by a log transformation
5. Sqft_living appears to have the strongest correlation with price. Sqft_above and bathrooms, also appear to have a pretty strong linear relationship with price.
6. Since the relationship between the ordinal variables (grade,condition, bathrooms,bedrooms) are not repectively monotonic, some of the results given by the regression model may appear non-sensical.
7. The descriptive analysis, as well as the visualizations show that there are outliers. I will begin by removing these.

Preprocessing Data
In this section, I will deal with incorrect data types, as well as missing , or 'zero-inflated' data. I will also create the categorical 'dummy' columns , as well as perform transformations on continuous data if necessary. I also created two 'preprocessed' dataframes, one that is scaled, and one that is not. Once I have all my final features, I will create another model with the identical set of features that are not scaled. It makes for an easier interpretation of the individual variables.**

The following is a description of the scaled dataframe after the preprocessing was completed:

```
In [30]: X_data_preproc_cmplt_scaled.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 21428 entries, 15279 to 7245
Data columns (total 71 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   view_EXCELLENT      21428 non-null  uint8
 1   view_FAIR           21428 non-null  uint8
 2   view_GOOD           21428 non-null  uint8
 3   view_NONE           21428 non-null  uint8
 4   view_missing        21428 non-null  uint8
 5   cond_Fair           21428 non-null  uint8
 6   cond_Good           21428 non-null  uint8
 7   cond_Poor           21428 non-null  uint8
 8   cond_Very Good      21428 non-null  uint8
 9   grd_11 Excellent    21428 non-null  uint8
 10  grd_12 Luxury       21428 non-null  uint8
 11  grd_13 Mansion      21428 non-null  uint8
 12  grd_3 Poor          21428 non-null  uint8
 13  grd_4 Low           21428 non-null  uint8
 14  grd_5 Fair          21428 non-null  uint8
 15  grd_6 Low Average   21428 non-null  uint8
 16  grd_7 Average       21428 non-null  uint8
 17  grd_8 Good          21428 non-null  uint8
 18  grd_9 Better        21428 non-null  uint8
 19  wtrfrnt_YES         21428 non-null  uint8
 20  wtrfrnt_missing     21428 non-null  uint8
 21  bdrm__2             21428 non-null  uint8
 22  bdrm__3             21428 non-null  uint8
 23  bdrm__4             21428 non-null  uint8
 24  bdrm__5             21428 non-null  uint8
 25  bdrm__6             21428 non-null  uint8
 26  bdrm__7             21428 non-null  uint8
 27  bdrm__8             21428 non-null  uint8
 28  bdrm__9             21428 non-null  uint8
 29  bdrm__10            21428 non-null  uint8
 30  bdrm__11            21428 non-null  uint8
 31  bdrm__33            21428 non-null  uint8
 32  bthrm__0.75         21428 non-null  uint8
 33  bthrm__1.0          21428 non-null  uint8
 34  bthrm__1.25         21428 non-null  uint8
 35  bthrm__1.5          21428 non-null  uint8
 36  bthrm__1.75         21428 non-null  uint8
 37  bthrm__2.0          21428 non-null  uint8
 38  bthrm__2.25         21428 non-null  uint8
 39  bthrm__2.5          21428 non-null  uint8
 40  bthrm__2.75         21428 non-null  uint8
 41  bthrm__3.0          21428 non-null  uint8
 42  bthrm__3.25         21428 non-null  uint8
 43  bthrm__3.5          21428 non-null  uint8
 44  bthrm__3.75         21428 non-null  uint8
 45  bthrm__4.0          21428 non-null  uint8
 46  bthrm__4.25         21428 non-null  uint8
```

```
 47  bthrm__4.5          21428 non-null  uint8
 48  bthrm__4.75         21428 non-null  uint8
 49  bthrm__5.0          21428 non-null  uint8
 50  bthrm__5.25         21428 non-null  uint8
 51  bthrm__5.5          21428 non-null  uint8
 52  bthrm__5.75         21428 non-null  uint8
 53  bthrm__6.0          21428 non-null  uint8
 54  bthrm__6.25         21428 non-null  uint8
 55  bthrm__6.5          21428 non-null  uint8
 56  bthrm__6.75         21428 non-null  uint8
 57  bthrm__7.5          21428 non-null  uint8
 58  bthrm__7.75         21428 non-null  uint8
 59  bthrm__8.0          21428 non-null  uint8
 60  flr__1.5            21428 non-null  uint8
 61  flr__2.0            21428 non-null  uint8
 62  flr__2.5            21428 non-null  uint8
 63  flr__3.0            21428 non-null  uint8
 64  flr__3.5            21428 non-null  uint8
 65  lat                 21428 non-null  float64
 66  log_sqft_living     21428 non-null  float64
 67  log_sqft_above      21428 non-null  float64
 68  log_sqft_lot        21428 non-null  float64
 69  log_sqft_living15   21428 non-null  float64
 70  log_sqft_lot15      21428 non-null  float64
dtypes: float64(6), uint8(65)
memory usage: 2.5 MB
```

Feature Selection

In this section I created the following three baseline models:
1. The feature with the highest correlation to the dependent variable (price)
2. The dataframe with all the features, as well as 'dummy' features, log transormations
3. In this dataframe I have dropped some 'generalized' features with high explanatory overlap.

Continuing with the third baseline model, I repeatedly remove features with p-values over.05, and features with high multi-colinearity. The following is the final model before I begin to confirm the assumptions of linear regression. At this point there are actually two models, that are identical except one has the 'lat' feature and one does not. This is because at this point I am not sure if I should remove the 'lat' feature in order to lower the Jarque Bera score.

Model with 'lat' feature:

Dep. Variable:	price	**R-squared:**	0.656
Model:	OLS	**Adj. R-squared:**	0.656
Method:	Least Squares	**F-statistic:**	907.4
Date:	Mon, 31 Jan 2022	**Prob (F-statistic):**	0.00
Time:	10:30:33	**Log-Likelihood:**	-5196.8
No. Observations:	21428	**AIC:**	1.049e+04
Df Residuals:	21382	**BIC:**	1.085e+04
Df Model:	45		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	12.8777	0.007	1789.974	0.000	12.864	12.892
view_EXCELLENT	0.4491	0.022	20.731	0.000	0.407	0.492
view_FAIR	0.2748	0.017	15.958	0.000	0.241	0.309
view_GOOD	0.3290	0.014	23.187	0.000	0.301	0.357
cond_Fair	-0.1126	0.024	-4.674	0.000	-0.160	-0.065
cond_Good	0.0935	0.005	18.122	0.000	0.083	0.104
cond_Very Good	0.1801	0.008	22.008	0.000	0.164	0.196
grd_11 Excellent	0.5794	0.018	33.051	0.000	0.545	0.614
grd_12 Luxury	0.7869	0.035	22.248	0.000	0.718	0.856
grd_13 Mansion	1.1282	0.096	11.765	0.000	0.940	1.316
grd_4 Low	-0.5897	0.060	-9.868	0.000	-0.707	-0.473
grd_5 Fair	-0.4066	0.021	-19.716	0.000	-0.447	-0.366
grd_6 Low Average	-0.2241	0.008	-27.016	0.000	-0.240	-0.208
grd_8 Good	0.1112	0.005	20.630	0.000	0.101	0.122
grd_9 Better	0.3141	0.008	41.622	0.000	0.299	0.329
wtrfrnt_YES	0.4002	0.031	12.714	0.000	0.339	0.462
bdrm__2	0.0261	0.007	3.594	0.000	0.012	0.040
bdrm__4	0.0760	0.005	14.613	0.000	0.066	0.086
bdrm__5	0.0922	0.009	10.324	0.000	0.075	0.110
bdrm__6	0.0434	0.020	2.197	0.028	0.005	0.082
bthrm__1.0	-0.2089	0.009	-23.280	0.000	-0.227	-0.191
bthrm__1.5	-0.1609	0.010	-16.041	0.000	-0.181	-0.141
bthrm__1.75	-0.0880	0.008	-10.821	0.000	-0.104	-0.072
bthrm__2.0	-0.0753	0.009	-8.343	0.000	-0.093	-0.058
bthrm__2.25	-0.0343	0.008	-4.161	0.000	-0.050	-0.018
bthrm__2.75	0.0756	0.010	7.418	0.000	0.056	0.096

bthrm__3.0	0.1147	0.012	9.328	0.000	0.091	0.139
bthrm__3.25	0.2693	0.014	19.554	0.000	0.242	0.296
bthrm__3.5	0.2830	0.013	22.360	0.000	0.258	0.308
bthrm__3.75	0.4121	0.026	15.811	0.000	0.361	0.463
bthrm__4.0	0.4193	0.028	15.008	0.000	0.365	0.474
bthrm__4.25	0.5334	0.036	14.814	0.000	0.463	0.604
bthrm__4.5	0.4171	0.033	12.751	0.000	0.353	0.481
bthrm__4.75	0.5971	0.066	9.108	0.000	0.469	0.726
bthrm__5.0	0.4942	0.068	7.216	0.000	0.360	0.628
bthrm__5.25	0.5476	0.087	6.324	0.000	0.378	0.717
bthrm__5.5	0.6505	0.100	6.476	0.000	0.454	0.847
bthrm__6.0	0.7073	0.130	5.456	0.000	0.453	0.961
bthrm__6.25	0.5680	0.224	2.534	0.011	0.129	1.007
bthrm__7.75	0.9592	0.324	2.957	0.003	0.323	1.595
bthrm__8.0	0.5088	0.225	2.261	0.024	0.068	0.950
flr__1.5	0.1560	0.008	19.960	0.000	0.141	0.171
flr__2.0	0.0954	0.006	16.180	0.000	0.084	0.107
flr__2.5	0.3229	0.025	12.884	0.000	0.274	0.372
lat	0.2210	0.002	101.018	0.000	0.217	0.225
log_sqft_lot	0.0531	0.002	23.346	0.000	0.049	0.058

Omnibus:	762.776	**Durbin-Watson:**	1.280
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	1154.282
Skew:	0.345	**Prob(JB):**	2.24e-251
Kurtosis:	3.903	**Cond. No.**	196.

Model without 'lat' feature:

Dep. Variable:	price	**R-squared:**	0.492
Model:	OLS	**Adj. R-squared:**	0.491
Method:	Least Squares	**F-statistic:**	471.2
Date:	Mon, 31 Jan 2022	**Prob (F-statistic):**	0.00
Time:	10:30:33	**Log-Likelihood:**	-9377.2
No. Observations:	21428	**AIC:**	1.884e+04
Df Residuals:	21383	**BIC:**	1.920e+04
Df Model:	44		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	12.8555	0.009	1470.896	0.000	12.838	12.873
view_EXCELLENT	0.4772	0.026	18.126	0.000	0.426	0.529
view_FAIR	0.2881	0.021	13.767	0.000	0.247	0.329
view_GOOD	0.2952	0.017	17.125	0.000	0.261	0.329
cond_Fair	-0.1385	0.029	-4.731	0.000	-0.196	-0.081
cond_Good	0.0679	0.006	10.843	0.000	0.056	0.080
cond_Very Good	0.1855	0.010	18.645	0.000	0.166	0.205
grd_11 Excellent	0.6795	0.021	31.946	0.000	0.638	0.721
grd_12 Luxury	0.8945	0.043	20.816	0.000	0.810	0.979
grd_13 Mansion	1.2653	0.117	10.857	0.000	1.037	1.494
grd_4 Low	-0.6990	0.073	-9.625	0.000	-0.841	-0.557
grd_5 Fair	-0.5281	0.025	-21.104	0.000	-0.577	-0.479
grd_6 Low Average	-0.2971	0.010	-29.581	0.000	-0.317	-0.277
grd_8 Good	0.1364	0.007	20.843	0.000	0.124	0.149
grd_9 Better	0.3794	0.009	41.519	0.000	0.362	0.397
wtrfrnt_YES	0.3448	0.038	9.015	0.000	0.270	0.420
bdrm__2	0.0854	0.009	9.694	0.000	0.068	0.103
bdrm__4	0.0815	0.006	12.890	0.000	0.069	0.094
bdrm__5	0.1095	0.011	10.091	0.000	0.088	0.131
bdrm__6	0.0714	0.024	2.978	0.003	0.024	0.118
bthrm__1.0	-0.1755	0.011	-16.098	0.000	-0.197	-0.154
bthrm__1.5	-0.1466	0.012	-12.030	0.000	-0.170	-0.123
bthrm__1.75	-0.0585	0.010	-5.919	0.000	-0.078	-0.039
bthrm__2.0	-0.0630	0.011	-5.748	0.000	-0.084	-0.042
bthrm__2.25	-0.0030	0.010	-0.301	0.763	-0.023	0.017
bthrm__2.75	0.1064	0.012	8.590	0.000	0.082	0.131

bthrm__3.0	0.1549	0.015	10.370	0.000	0.126	0.184
bthrm__3.25	0.3453	0.017	20.659	0.000	0.313	0.378
bthrm__3.5	0.3433	0.015	22.340	0.000	0.313	0.373
bthrm__3.75	0.5125	0.032	16.194	0.000	0.451	0.575
bthrm__4.0	0.5075	0.034	14.953	0.000	0.441	0.574
bthrm__4.25	0.6076	0.044	13.889	0.000	0.522	0.693
bthrm__4.5	0.5026	0.040	12.644	0.000	0.425	0.581
bthrm__4.75	0.7081	0.080	8.888	0.000	0.552	0.864
bthrm__5.0	0.4961	0.083	5.959	0.000	0.333	0.659
bthrm__5.25	0.5802	0.105	5.513	0.000	0.374	0.787
bthrm__5.5	0.7151	0.122	5.858	0.000	0.476	0.954
bthrm__6.0	0.7434	0.158	4.718	0.000	0.435	1.052
bthrm__6.25	0.6886	0.272	2.528	0.011	0.155	1.223
bthrm__7.75	1.0014	0.394	2.540	0.011	0.229	1.774
bthrm__8.0	0.6557	0.274	2.397	0.017	0.120	1.192
flr__1.5	0.2014	0.009	21.242	0.000	0.183	0.220
flr__2.0	0.0458	0.007	6.412	0.000	0.032	0.060
flr__2.5	0.2985	0.030	9.802	0.000	0.239	0.358
log_sqft_lot	0.0192	0.003	7.008	0.000	0.014	0.025

Omnibus:	150.455	**Durbin-Watson:**	0.932
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	155.406
Skew:	0.194	**Prob(JB):**	1.79e-34
Kurtosis:	3.153	**Cond. No.**	196.

Assumptions of Multiple Linear Regression

In this section I had to remove some rows in order to satisfy the assumptions of homoscedasticity, and normality of the residuals. This is the final model after all the assumptions were satisfied:

Dep. Variable:	price	**R-squared:**	0.653
Model:	OLS	**Adj. R-squared:**	0.652
Method:	Least Squares	**F-statistic:**	943.7
Date:	Mon, 31 Jan 2022	**Prob (F-statistic):**	0.00
Time:	10:30:36	**Log-Likelihood:**	-2586.6
No. Observations:	20590	**AIC:**	5257.
Df Residuals:	20548	**BIC:**	5590.
Df Model:	41		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	12.8553	0.007	1971.557	0.000	12.843	12.868
view_EXCELLENT	0.4225	0.021	20.154	0.000	0.381	0.464
view_FAIR	0.2550	0.016	16.168	0.000	0.224	0.286
view_GOOD	0.3205	0.013	24.471	0.000	0.295	0.346
cond_Fair	-0.0772	0.023	-3.305	0.001	-0.123	-0.031
cond_Good	0.0855	0.005	18.244	0.000	0.076	0.095
cond_Very Good	0.1658	0.007	22.293	0.000	0.151	0.180
grd_11 Excellent	0.5802	0.017	34.108	0.000	0.547	0.614
grd_12 Luxury	0.7530	0.039	19.542	0.000	0.677	0.828
grd_13 Mansion	1.1681	0.275	4.248	0.000	0.629	1.707
grd_4 Low	-0.3119	0.071	-4.385	0.000	-0.451	-0.172
grd_5 Fair	-0.2908	0.021	-13.877	0.000	-0.332	-0.250
grd_6 Low Average	-0.1990	0.008	-26.198	0.000	-0.214	-0.184
grd_8 Good	0.1329	0.005	27.317	0.000	0.123	0.142
grd_9 Better	0.3420	0.007	49.922	0.000	0.329	0.355
wtrfrnt_YES	0.3045	0.033	9.291	0.000	0.240	0.369
bdrm__2	0.0343	0.007	5.178	0.000	0.021	0.047
bdrm__4	0.0758	0.005	16.135	0.000	0.067	0.085
bdrm__5	0.0854	0.008	10.476	0.000	0.069	0.101
bdrm__6	0.0496	0.018	2.731	0.006	0.014	0.085
bthrm__1.0	-0.1775	0.008	-21.764	0.000	-0.194	-0.162
bthrm__1.5	-0.1482	0.009	-16.336	0.000	-0.166	-0.130
bthrm__1.75	-0.0742	0.007	-10.116	0.000	-0.089	-0.060
bthrm__2.0	-0.0651	0.008	-8.016	0.000	-0.081	-0.049

bthrm__2.25	-0.0408	0.007	-5.484	0.000	-0.055	-0.026
bthrm__2.75	0.0765	0.009	8.331	0.000	0.058	0.094
bthrm__3.0	0.1013	0.011	9.071	0.000	0.079	0.123
bthrm__3.25	0.2245	0.013	17.537	0.000	0.199	0.250
bthrm__3.5	0.2534	0.012	21.774	0.000	0.231	0.276
bthrm__3.75	0.3793	0.025	15.397	0.000	0.331	0.428
bthrm__4.0	0.3416	0.027	12.566	0.000	0.288	0.395
bthrm__4.25	0.4654	0.037	12.443	0.000	0.392	0.539
bthrm__4.5	0.3499	0.031	11.147	0.000	0.288	0.411
bthrm__4.75	0.2176	0.092	2.357	0.018	0.037	0.398
bthrm__5.0	0.2973	0.074	4.002	0.000	0.152	0.443
bthrm__5.25	0.4710	0.092	5.112	0.000	0.290	0.652
bthrm__5.5	0.5602	0.138	4.060	0.000	0.290	0.831
flr__1.5	0.1595	0.007	22.606	0.000	0.146	0.173
flr__2.0	0.0864	0.005	16.151	0.000	0.076	0.097
flr__2.5	0.2707	0.024	11.314	0.000	0.224	0.318
lat	0.2084	0.002	105.639	0.000	0.205	0.212
log_sqft_lot	0.0472	0.002	23.018	0.000	0.043	0.051

Omnibus:	64.244	**Durbin-Watson:**	0.005
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	64.770
Skew:	0.136	**Prob(JB):**	8.62e-15
Kurtosis:	3.042	**Cond. No.**	181.

Renovation and Remodel Recommendations

Baseline model:
'const': 1
'lat': 47.56
'view': 'AVERAGE' (default)
'condition': 'AVERAGE' (default)
'grade': '10 Very Good' (default)
'waterfront': 'NO' (default)
'bedrooms': 1 (default)
'bathrooms': 0.5 (default)
'floors': 1 (default)
'log_sqft_lot': 9
'price' : 12.855537576059533 aprox.: 382903.231151

1. Adding a Second Floor:

The following is the price predicted by the model when the baseline model is changed from 1 floor, to 2 floors. The result is the power to which Euler's number (e) should be raised, in order to derive the price:

```
model_final_no_scale.predict(house_features)[0]
```

```
12.941941308413988
```

In this case, we have a log-level regression, where the dependent variable 'price', is log transformed, and the independent variable 'flr__2.0' is not. It takes the following form:
ln y = b0 + b1(x) + E
where b0 is the constant, b1 is the slope coefficient, and E is the error term.
We can calculate the change in price, based on the change in x with the following formula:
%(change in y) = 100 * ((e^b1) - 1)
Based on this formula, with b1 = 0.0864, the change in y would be equal to 9.02423%. If we multiply 382,903.231151 by 1.0902423, we get 417,457.299407. If we compare this to the price predicted by the model in terms of an exponent,e^12.941941308413988 = 417458.872072. The log-level interpretation matches the predicted value with a precision of 99.99962328%.

2. Upgrading Building Grade to 'grd_11 Excellent':

The following is the price predicted by the model when the baseline model is changed from grade 10 to grade 11. The result is the power to which Euler's number (e) should be raised, in order to derive the price:

```
model_final_no_scale.predict(house_features)[0]
```

```
13.43576963632787
```

In this case, we have a log-level regression, where the dependent variable 'price', is log transformed, and the independent variable 'grd_11 Excellent' is not. It takes the following form:
ln y = b0 + b1(x) + E
where b0 is the constant, b1 is the slope coefficient, and E is the error term.
We can calculate the change in price, based on the change in x with the following formula:
%(change in y) = 100 * ((e^b1) - 1)
Based on this formula, with b1 = 0.5802, the change in y would be equal to 78.639567%. If we multiply 382,903.231151 by 1.78639567, we get 684,016.674157. If we compare this to the price predicted by the model in terms of an exponent, e^13.43576963632787 = 684,038.60586. The log-level interpretation matches the predicted value with a precision of 99.996793791%.

3. Upgrading to Four Bedrooms:

The following is the price predicted by the model when the baseline model is changed from 1 bedroom to 4 bedrooms. The result is the power to which Euler's number (e) should be raised, in order to derive the price:

```
model_final_no_scale.predict(house_features)[0]
```

```
12.931293218903669
```

In this case, we have a log-level regression, where the dependent variable 'price', is log transformed, and the independent variable 'bdrm__4' is not. It takes the following form:
ln y = b0 + b1(x) + E
where b0 is the constant, b1 is the slope coefficient, and E is the error term.
We can calculate the change in price, based on the change in x with the following formula:
%(change in y) = 100 * ((e^b1) - 1)
Based on this formula, with b1 = 0.0758, the change in y would be equal to 7.87468%. If we multiply 382,903.231151 by 1.0787468, we get 413,055.635314. If we compare this to the price predicted by the model in terms of an exponent, e^12.931293218903669 = 413,037.31498. The log-level interpretation matches the predicted value with a precision of 99.995564681%.

Project Conclusion: Main Take-aways

1. It is critical to have some understanding of the subject matter one is dealing with, in order to be able to select features, more effectively, and root out non-sensical results.
2. If the ordinal data in one's data set does not have a monotonic relationship, one may wind up with non-sensical results. For example, recommending a downgrade from 'grd_10 Very Good' to 'grd_9 Better', in order to increase the price of the home.
3. High multi-colinearity between features, may lead to unexpected results. For example, one may have two features that are highly correlated, where both have positive Pearson correlations, but one has a negative regression coefficient.
4. The features with the higher Perason correlation coefficient, do not necessarily have a higher regression coefficient. I found this particularly in the 'dummy' variables. I think this may have something to do with these variables not having an equal number of occurences, since only one can be chosen at a time, for each row of data.
5. I understand that the distribution of the residuals do not necessarily have to be perfectly normal, but I am not clear as to what is defined as 'normal enough'.
6. I may have been able to further improve the R-squared score by adding 'sqft_bsmnt' as a binary variable.
7. I originally planned to associate the zip codes with their corresponding average household incomes (per the U.S census bureau),then binning by income groups, and creating interactions, but ultimately decicded against it because it generated too many columns, and didn't lend itself to learning.
8. Log transforming the dependent variable can help approximate a normal residual distribution.
9. Remove rows with outliers in the dependent variable can help approximate a normal residual distribution.

For More Information

Please review my full analysis in my Jupyter Notebook or mypresentation.
For any additional questions, please contact **Freddy Abrahamson at** fred0421@hotmail.com,

Repository Structure

```
├── README.md                                   <- The top-level README
for reviewers of this project
├── student.ipynb                               <- Narrative
documentation of analysis in Jupyter notebook
├── Phase_2_Project_Presentation.pdf            <- PDF version of project
presentation
├── CONTRIBUTING.md                             <- Contributing to
Learn.co Curriculum
├── LICENSE.md                                  <- Learn.co Educational
Content License
└── images                                      <- Images used for this
project
```

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

- **Jupyter Notebook** 100.0%