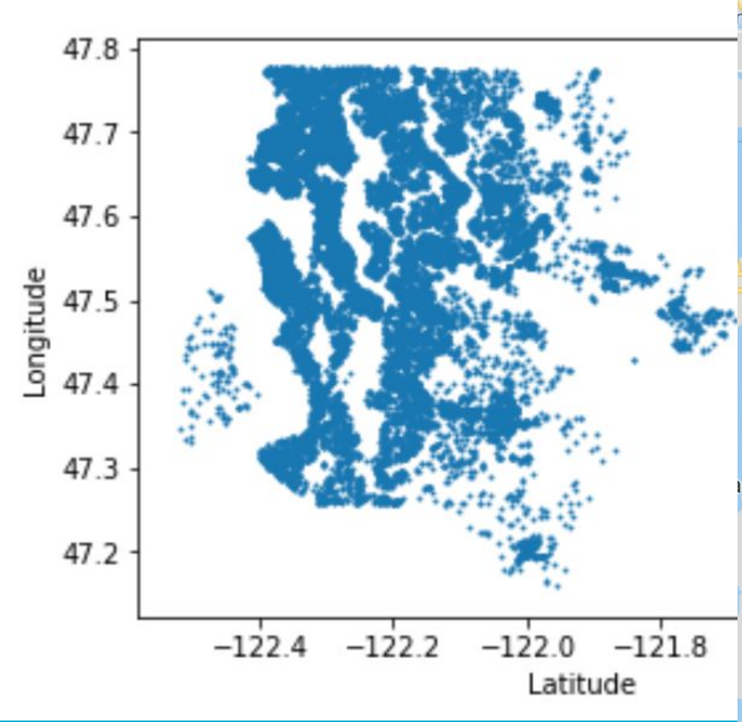
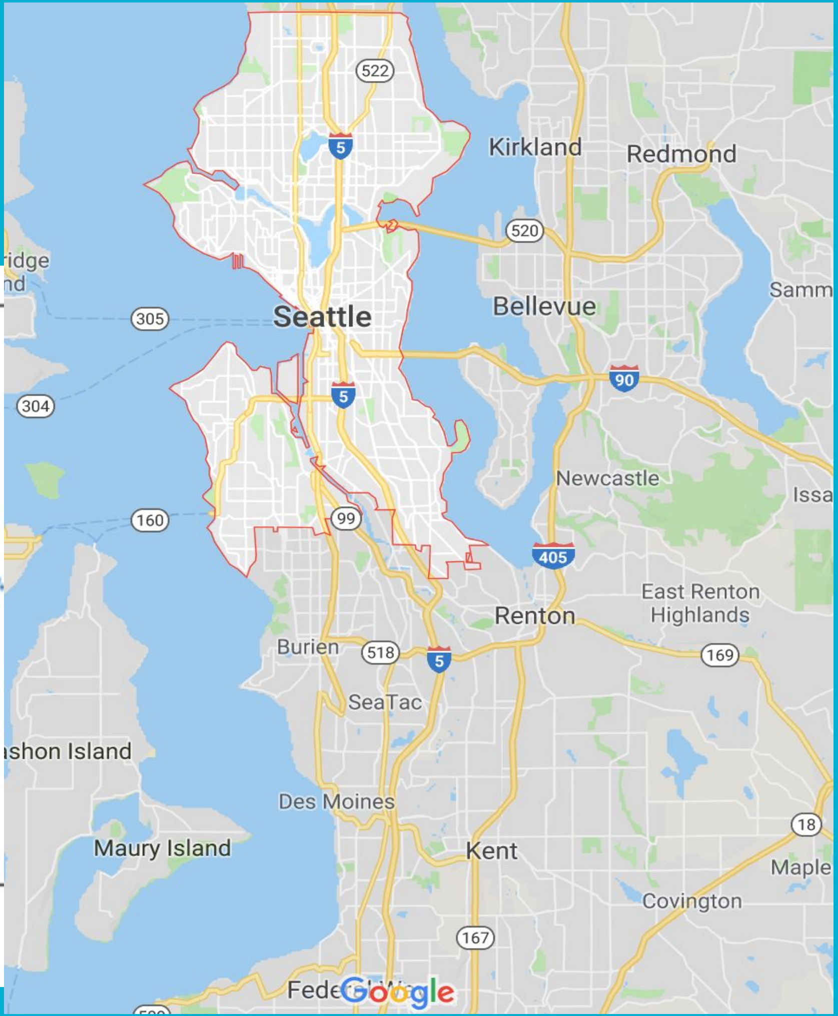
Predicting Home Sale Prices in Greater Seattle

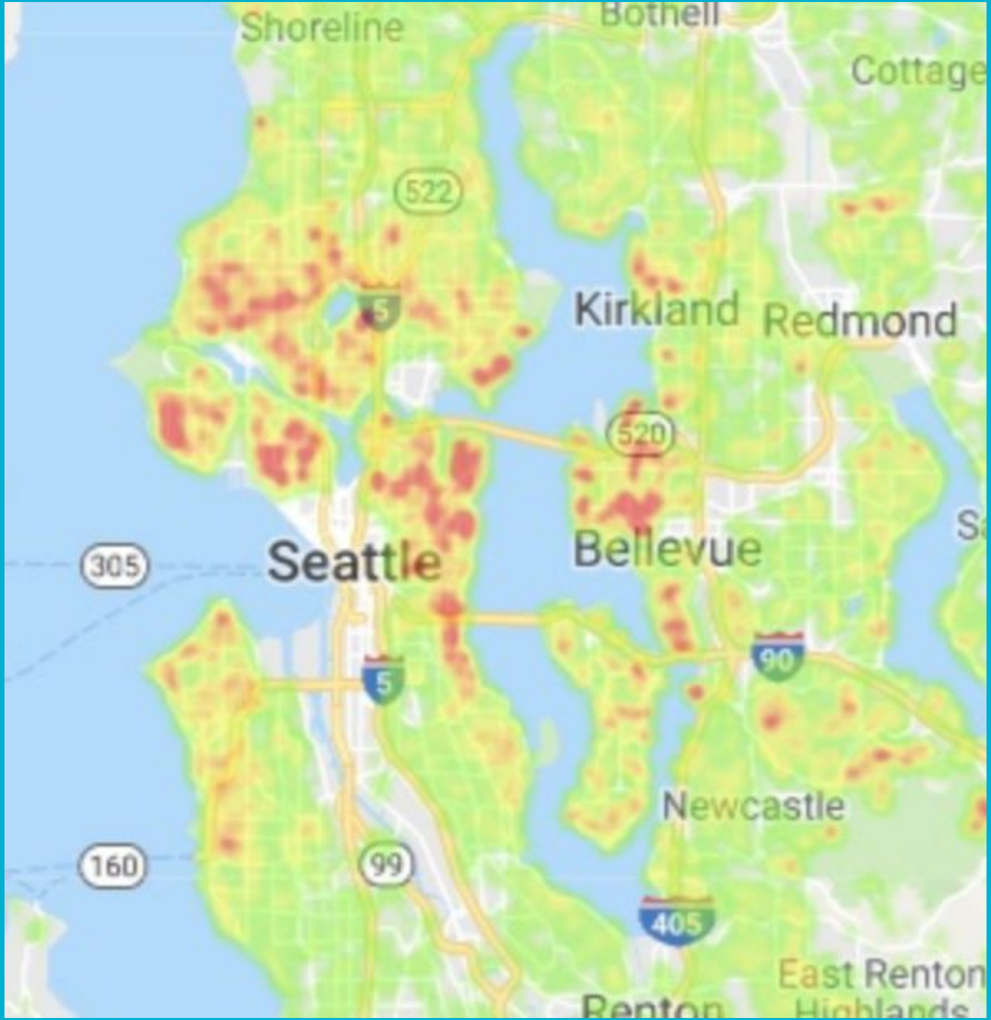
Leveraging a King County House Sales dataset for insights

Scatter Plot of Latitude & Longitude





Heat Map Highlighting Housing Density



King County House Sales Dataset

Original Dataset (Sales from 2014-2015)

- 21,597 instances of home sales (observations, rows)
- 21 columns or features

Cleaned Dataset

- 21,313 observations after removing 284 outliers (or extreme influencers)
- Maintained 98.7% of the data set (trimmed only 1.3% of the data)
- Included 10 of the original 21 features in our model as predictors of price

Model Description

Developed a model with 85% accuracy

- Important predictors...
  - Square footage of the home
  - Location
  - Lot size
  - Cumulative impact of condition, year built & renovation
- Outcome (or target) variable → log of price

Training Data

75%

```
In [45]: model = ols(formula= formula, data=train).fit()
         model.summary()
```

Out[45]:

OLS Regression Results

Dep. Variable:	log_price	**R-squared:**	0.859
Model:	OLS	**Adj. R-squared:**	0.857
Method:	Least Squares	**F-statistic:**	491.2
Date:	Thu, 20 Jun 2019	**Prob (F-statistic):**	0.00
Time:	10:37:31	**Log-Likelihood:**	3675.8
No. Observations:	15984	**AIC:**	-6960.
Df Residuals:	15788	**BIC:**	-5454.
Df Model:	195		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	41.8020	0.261	159.971	0.000	41.290	42.314

Test Data

25%

```
In [46]: model = ols(formula= formula, data=test).fit()
         model.summary()
```

Out[46]:

OLS Regression Results

Dep. Variable:	log_price	**R-squared:**	0.865
Model:	OLS	**Adj. R-squared:**	0.860
Method:	Least Squares	**F-statistic:**	168.5
Date:	Thu, 20 Jun 2019	**Prob (F-statistic):**	0.00
Time:	10:42:10	**Log-Likelihood:**	1468.4
No. Observations:	5329	**AIC:**	-2545.
Df Residuals:	5133	**BIC:**	-1255.
Df Model:	195		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	41.1340	0.440	93.442	0.000	40.271	41.997

Conclusions

- **Location, location, location**
  [Next iteration of model will leverage latitude & longitude]
- **Major on the majors**
  [# of bathrooms, neighborhood & grade all captured by the proxy, square footage]

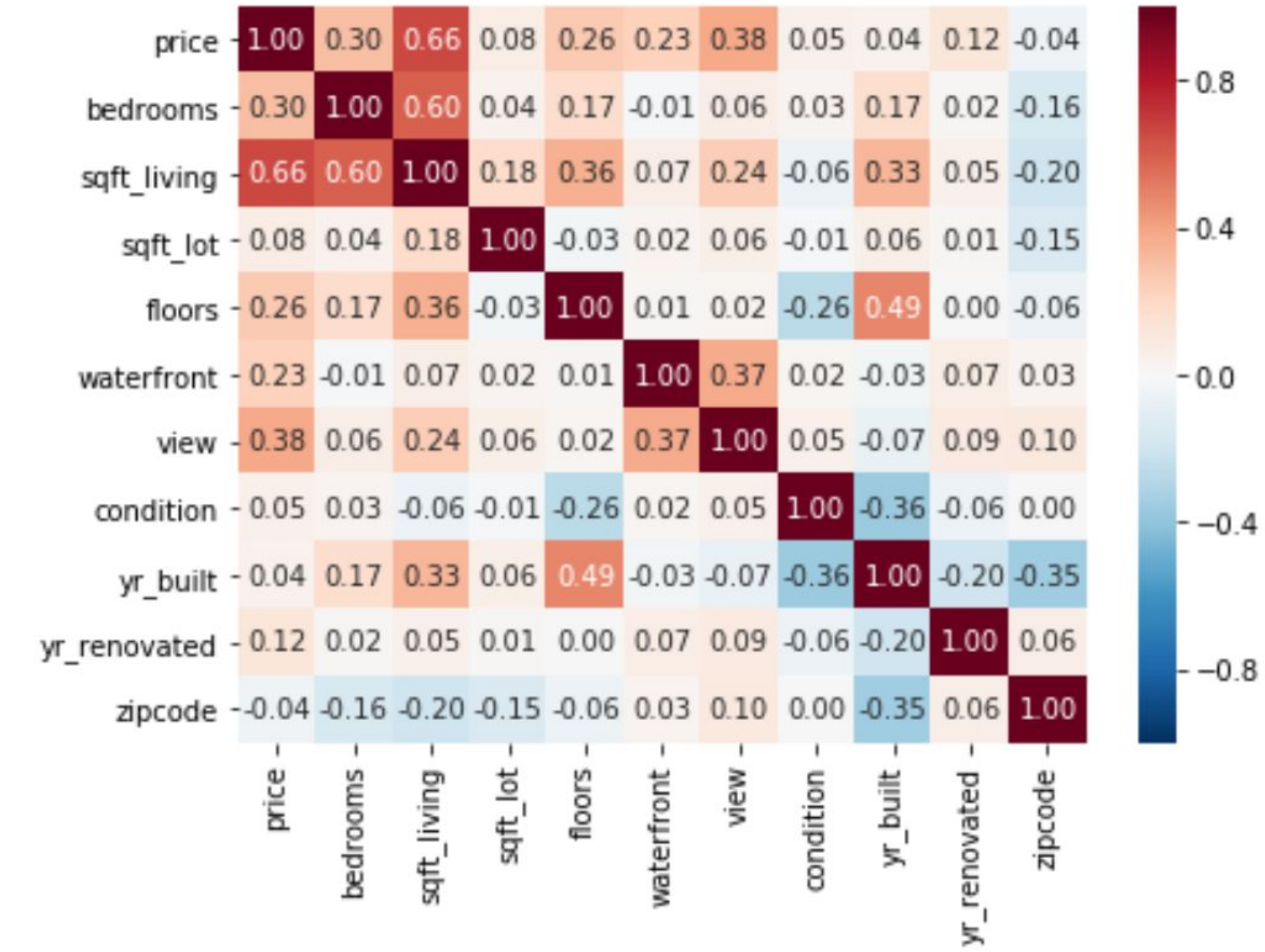
Questions?

Multicolinearity

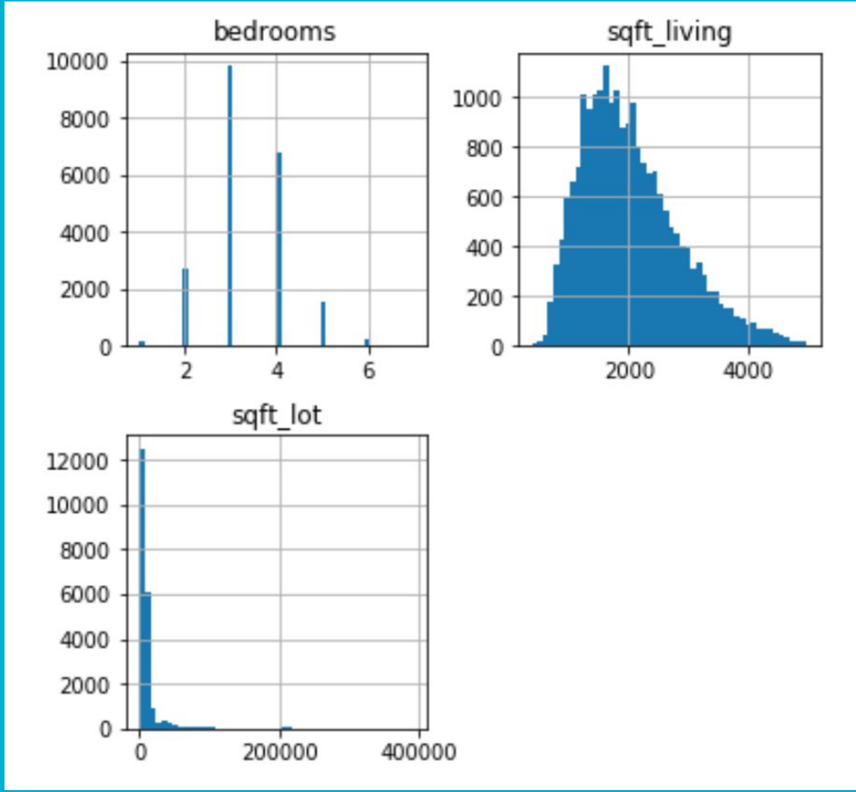
```
In [17]: df['sqft_living'].corr(df['sqft_above'])

Out[17]: 0.8764477590354981

In [18]: df['sqft_living'].corr(df['bathrooms'])

Out[18]: 0.7557576009502521

In [19]: df['sqft_living'].corr(df['grade'])

Out[19]: 0.7627790466721344

In [20]: df['sqft_living'].corr(df['sqft_living15'])

Out[20]: 0.7564015282475002
```

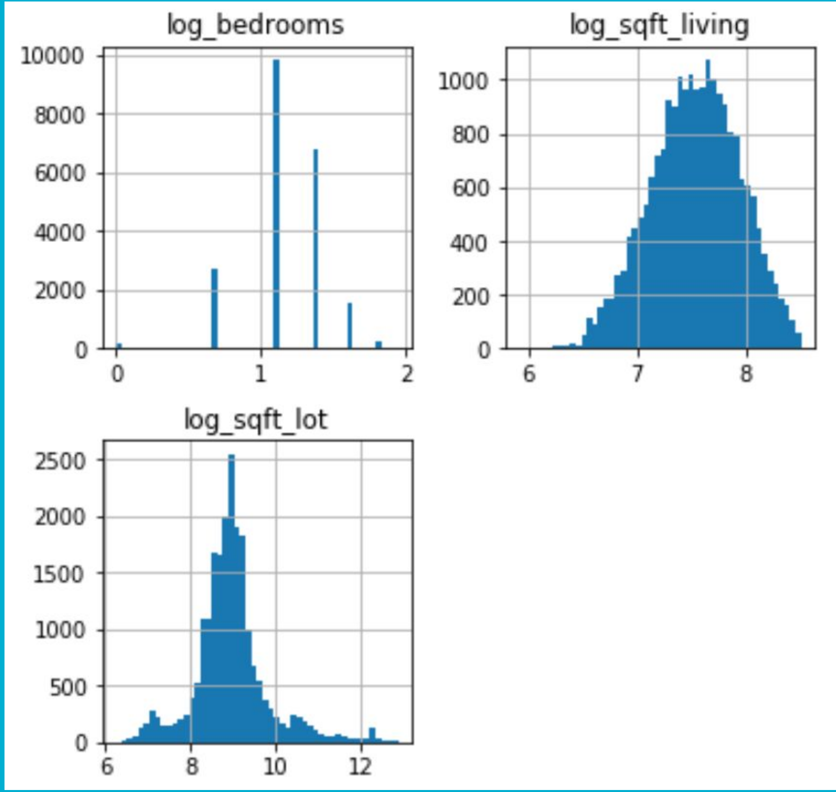
.75 Threshold

```
In [29]: df_dropped['sqft_lot'].corr(df['sqft_lot15'])

Out[29]: 0.7881861189991689
```


price 1.00 0.30 0.66 0.08 0.26 0.23 0.38 0.05 0.04 0.12 -0.04
bedrooms 0.30 1.00 0.60 0.04 0.17 -0.01 0.06 0.03 0.17 0.02 -0.16
sqft_living 0.66 0.60 1.00 0.18 0.36 0.07 0.24 -0.06 0.33 0.05 -0.20
sqft_lot 0.08 0.04 0.18 1.00 -0.03 0.02 0.06 -0.01 0.06 0.01 -0.15
floors 0.26 0.17 0.36 -0.03 1.00 0.01 0.02 -0.26 0.49 0.00 -0.06
waterfront 0.23 -0.01 0.07 0.02 0.01 1.00 0.37 0.02 -0.03 0.07 0.03
view 0.38 0.06 0.24 0.06 0.02 0.37 1.00 0.05 -0.07 0.09 0.10
condition 0.05 0.03 -0.06 -0.01 -0.26 0.02 0.05 1.00 -0.36 -0.06 0.00
yr_built 0.04 0.17 0.33 0.06 0.49 -0.03 -0.07 -0.36 1.00 -0.20 -0.35
yr_renovated 0.12 0.02 0.05 0.01 0.00 0.07 0.09 -0.06 -0.20 1.00 0.06
zipcode -0.04 -0.16 -0.20 -0.15 -0.06 0.03 0.10 0.00 -0.35 0.06 1.00
price bedrooms sqft_living sqft_lot floors waterfront view condition yr_built yr_renovated zipcode
0.8
0.4
0.0
−0.4
−0.8

Transforming a few features





Price → Log of Price