```
!pip install turicreate
```

```
from google.colab import drive
drive.mount('/content/gdrive')
```

```
Mounted at /content/gdrive
```

Launch Turi Create

```
import turicreate
```

Load house sales data

```
sales = turicreate.SFrame('/content/gdrive/My Drive/Turicreate/Week 2/home_data.sfr
```

+ 코드 + 텍스트

```
sales
```

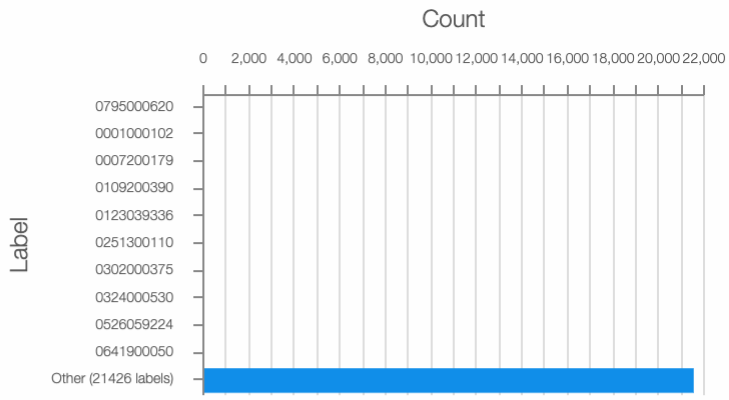
id	date	price	bedrooms	bathrooms	sqft_living	sqft
7129300520	2014-10-13 00:00:00+00:00	221900	3	1	1180	56
6414100192	2014-12-09 00:00:00+00:00	538000	3	2.25	2570	72
5631500400	2015-02-25 00:00:00+00:00	180000	2	1	770	100
2487200875	2014-12-09	604000	4	3	1960	50

Explore

```
sales.show()
```

```
Materializing SFrame
Warning: Skipping column 'date'. Unable to show columns of type 'datetime'; or
Further warnings of unsupported type will be suppressed.
```

id



Num. Rows:

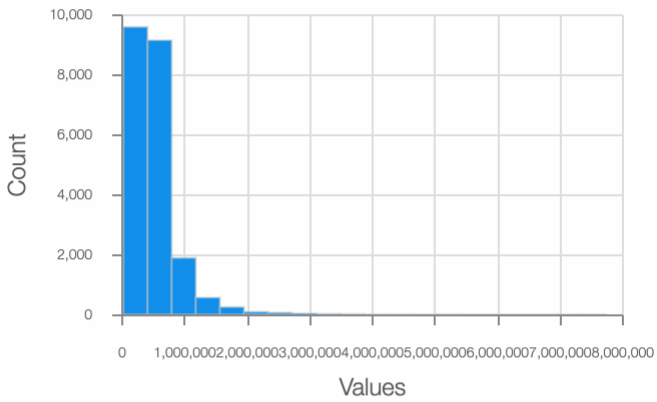
Num. Unique:

Missing:

Frequent Items

0795000620
0001000102
0007200179
0109200390
0123039336

price



Num. Rows:

Num. Unique:

Missing:

Mean: 540,0

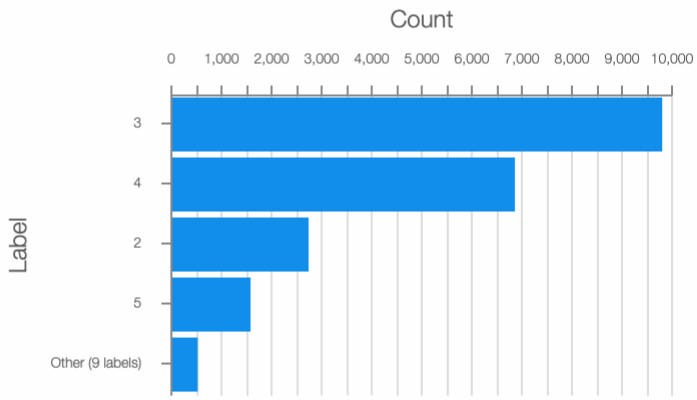
Min:

Max:

Median:

St. Dev: 367,1

bedrooms



Num. Rows:

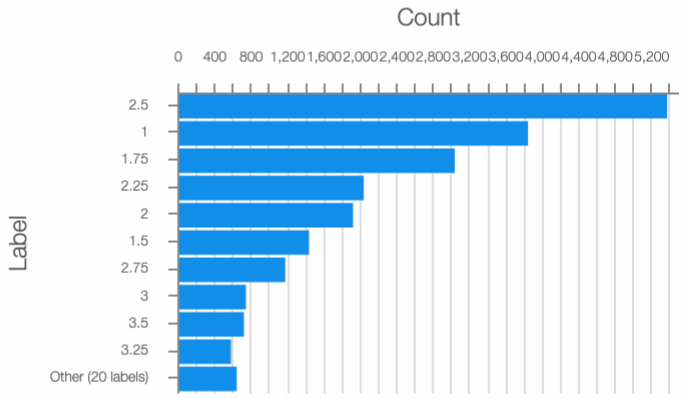
Num. Unique:

Missing:

Frequent Items

3
4
2
5
Other (9 labels)

bathrooms



Num. Rows:

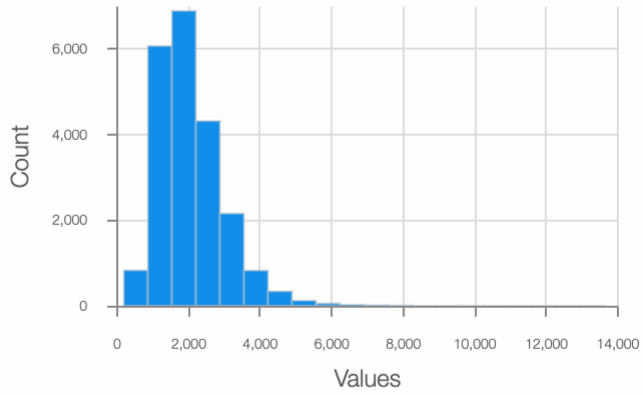
Num. Unique:

Missing:

Frequent Items

2.5
1
1.75
2.25
2

sqft_living



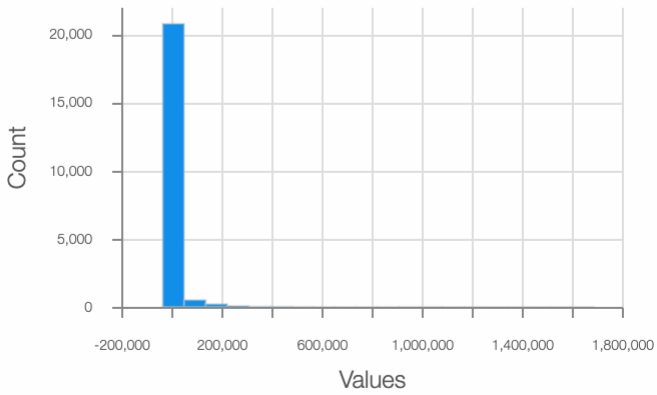
Num. Rows:

Num. Unique:

Missing:

Mean: 2,0
Min:
Max:
Median:
St. Dev: 9

sqft_lot



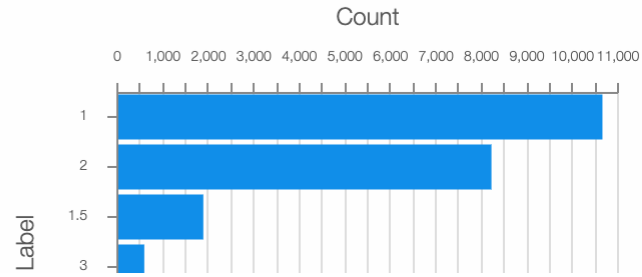
Num. Rows:

Num. Unique:

Missing:

Mean: 15,1
Min:
Max:
Median:
St. Dev: 41,4

floors



Num. Rows:

Num. Unique:

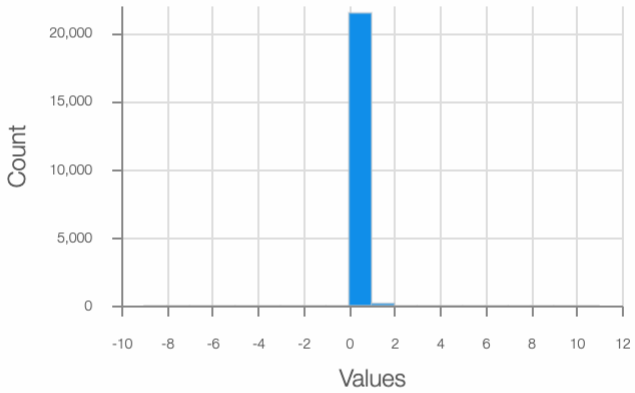
Missing:

Frequent Items

1

2
1.5
3
2.5

waterfront



Num. Rows:

Num. Unique:

Missing:

Mean:

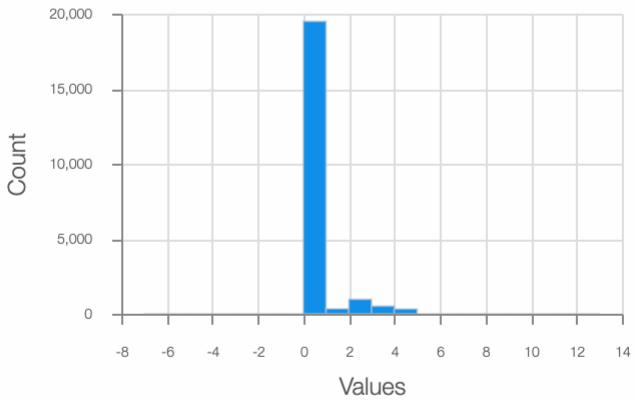
Min:

Max:

Median:

St. Dev:

view



Num. Rows:

Num. Unique:

Missing:

Mean:

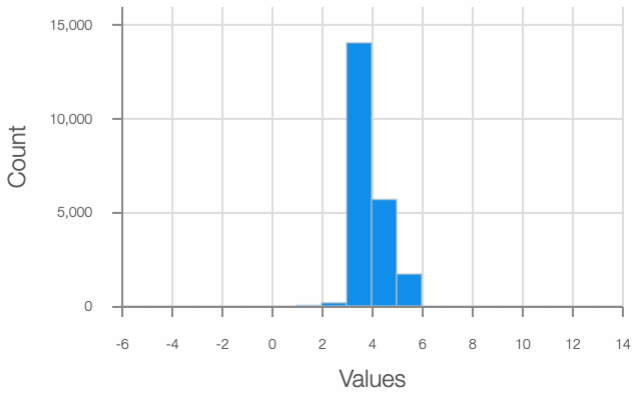
Min:

Max:

Median:

St. Dev:

condition



Num. Rows:

Num. Unique:

Missing:

Mean:

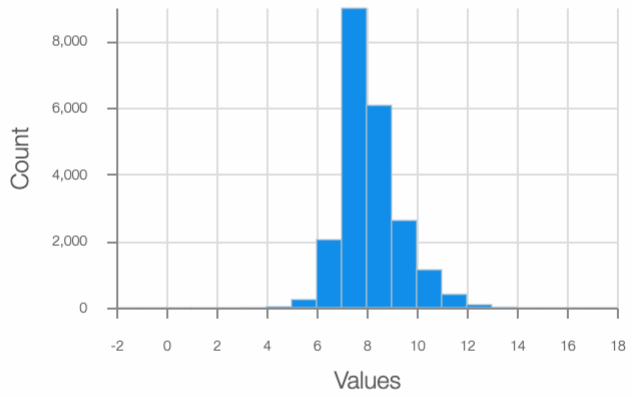
Min:

Max:

Median:

St. Dev:

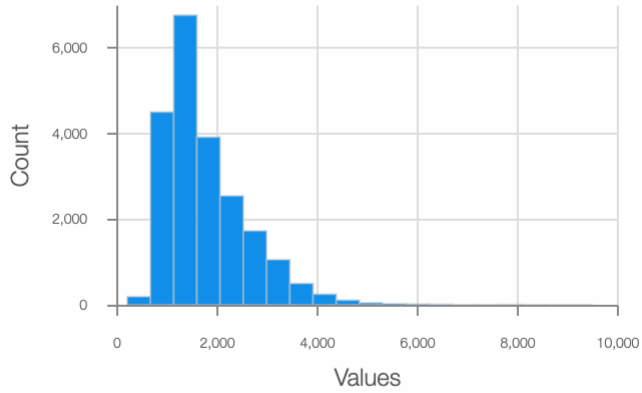
grade



Num. Rows:
Num. Unique:
Missing:

Mean:
Min:
Max:
Median:
St. Dev:

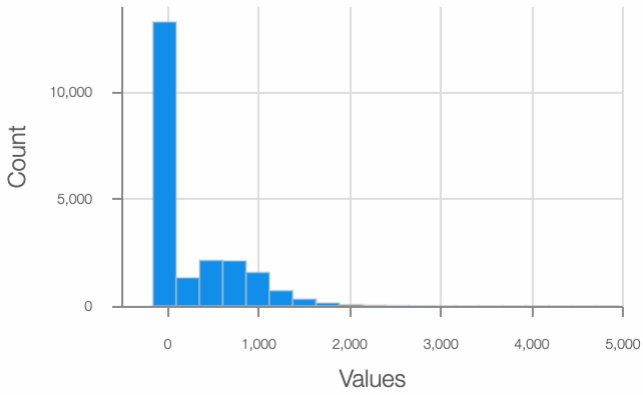
sqft_above



Num. Rows:
Num. Unique:
Missing:

Mean: 1,7
Min:
Max:
Median:
St. Dev:

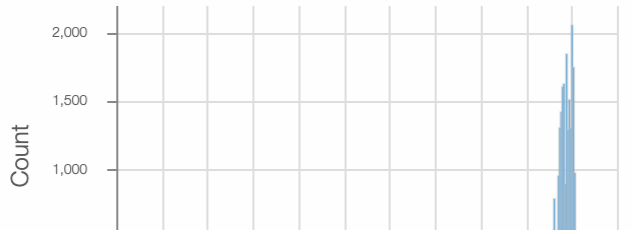
sqft_basement



Num. Rows:
Num. Unique:
Missing:

Mean: 2
Min:
Max:
Median:
St. Dev: 4

yr_built

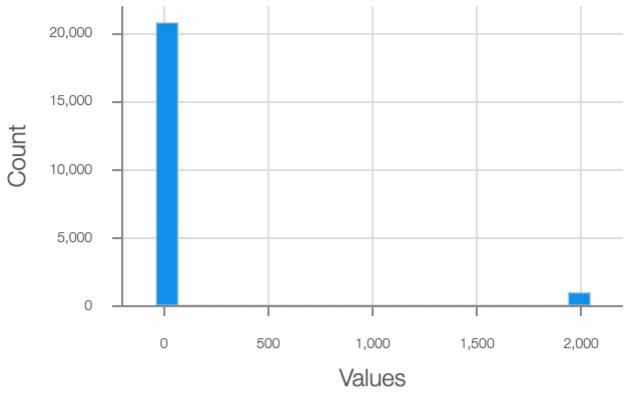


Num. Rows:
Num. Unique:
Missing:

Mean: 1,9
Min:

Min:
Max:
Median:
St. Dev:

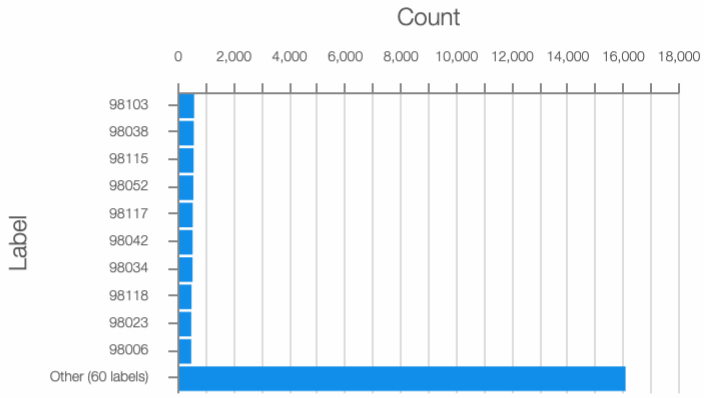
yr_renovated



Num. Rows:
Num. Unique:
Missing:

Mean:
Min:
Max:
Median:
St. Dev: 4

zipcode

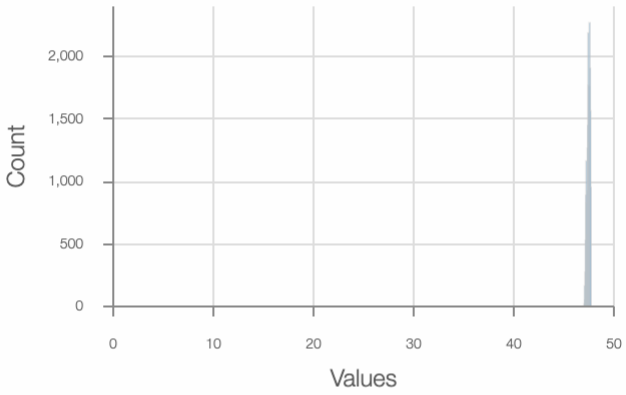


Num. Rows:
Num. Unique:
Missing:

Frequent Items

98103
98038
98115
98052
98117

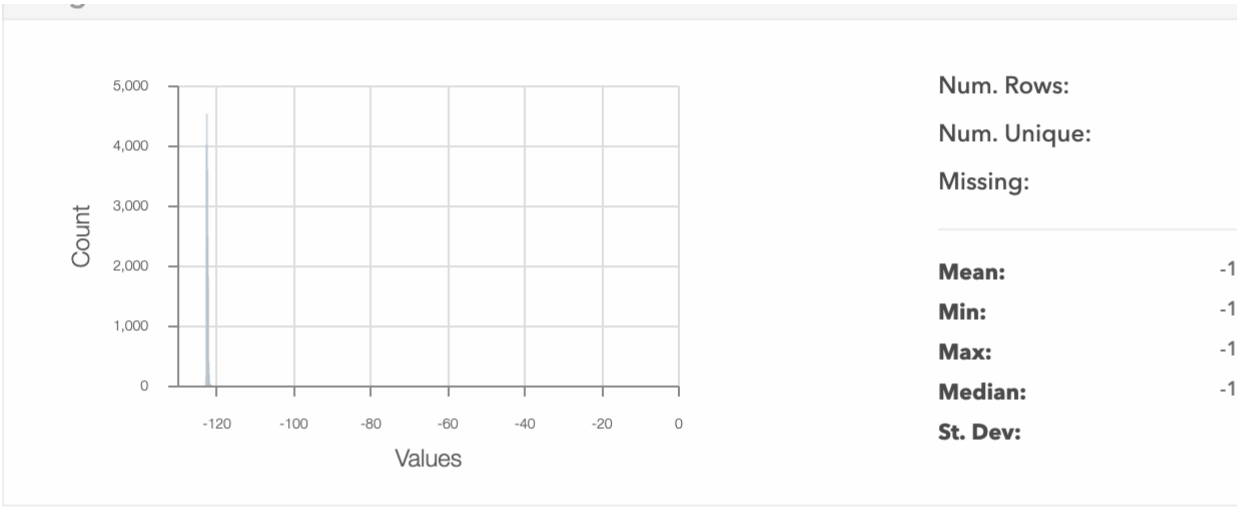
lat



Num. Rows:
Num. Unique:
Missing:

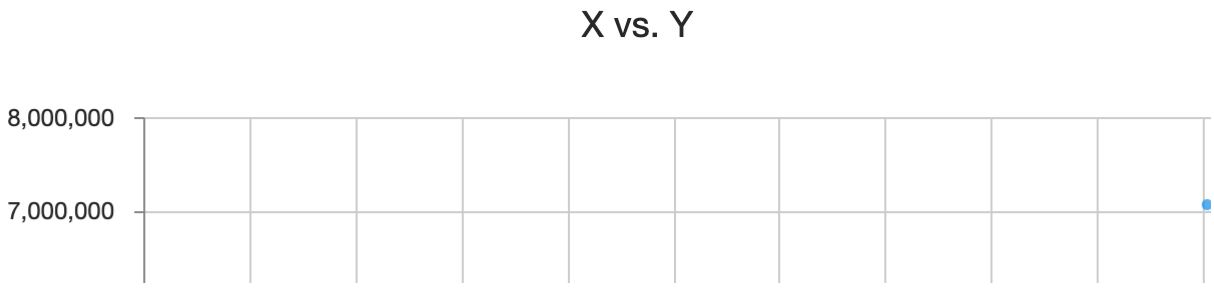
Mean:
Min:
Max:
Median:
St. Dev:

long

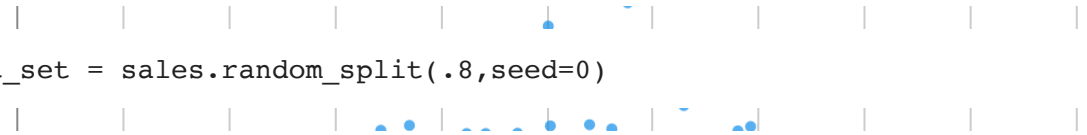



```
turicreate.show(sales[1:5000]['sqft_living'],sales[1:5000]['price'])
```

```
Materializing X axis SArray
Materializing Y axis SArray
```



Simple regression model that predicts price from square feet



```
training_set, test_set = sales.random_split(.8,seed=0)
```

train simple regression model


```
sqft_model = turicreate.linear_regression.create(training_set,target='price',featur

    PROGRESS: Creating a validation set from 5 percent of training data. This may
              You can set ``validation_set=None`` to disable validation tracking.

    Linear regression:
    --------------------------------------------------------
    Number of examples          : 16514
    Number of features          : 1
    Number of unpacked features : 1
    Number of coefficients    : 2
    Starting Newton Method
    --------------------------------------------------------
    +-----------+----------+--------------+--------------------+------------------
    | Iteration | Passes   | Elapsed Time | Training Max Error | Validation Max Er
    +-----------+----------+--------------+--------------------+------------------
    | 1         | 2        | 1.004688     | 4347310.043694     | 2673878.325606
    +-----------+----------+--------------+--------------------+------------------
    SUCCESS: Optimal solution found.
```

Evaluate the quality of our model

```
print (test_set['price'].mean())

    543054.0425632533
```

```
print (sqft_model.evaluate(test_set))

    {'max_error': 4141808.6553260004, 'rmse': 255195.34326236605}
```

Explore model a little further

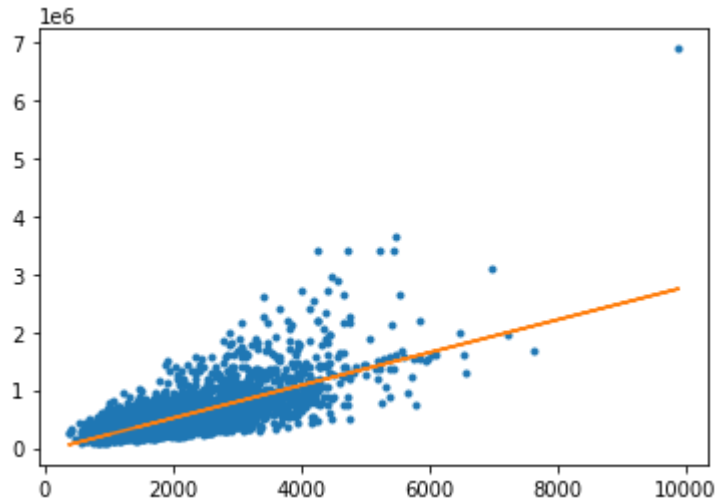
```
sqft_model.coefficients
```

name	index	value	stderr
(intercept)	None	-47522.20580990845	5064.391930298379
sqft_living	None	282.17528316318584	2.2259407752400837

[2 rows x 4 columns]

```
import matplotlib.pyplot as plt
%matplotlib inline
plt.plot(test_set['sqft_living'],test_set['price'],'.',
        test_set['sqft_living'],sqft_model.predict(test_set),'-')
```

```
[<matplotlib.lines.Line2D at 0x7f56b7e97d90>,
 <matplotlib.lines.Line2D at 0x7f56b8f57890>]
```



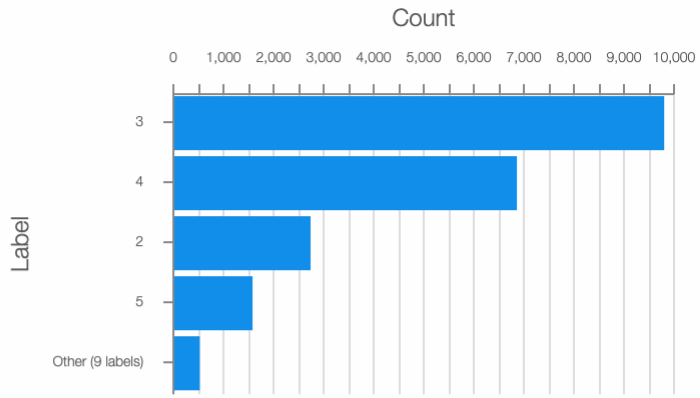
Explore other features of the data

```
my_features = ['bedrooms','bathrooms','sqft_living','sqft_lot','floors','zipcode']
```

```
sales[my_features].show()
```

```
Materializing SFrame
```

bedrooms



Num. Rows:

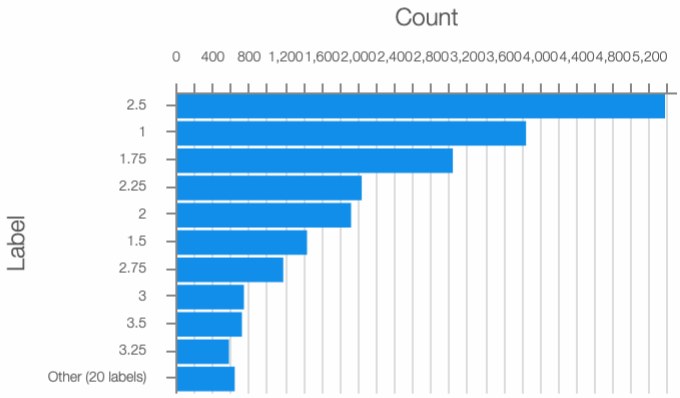
Num. Unique:

Missing:

Frequent Items

3
4
2
5
Other (9 labels)

bathrooms



Num. Rows:

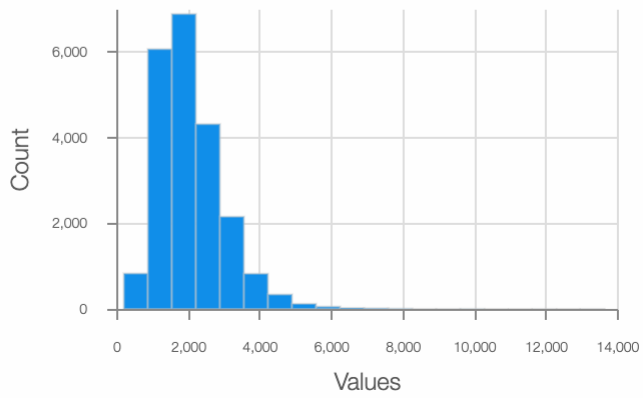
Num. Unique:

Missing:

Frequent Items

2.5
1
1.75
2.25
2

sqft_living



Num. Rows:

Num. Unique:

Missing:

Mean: 2,0

Min:

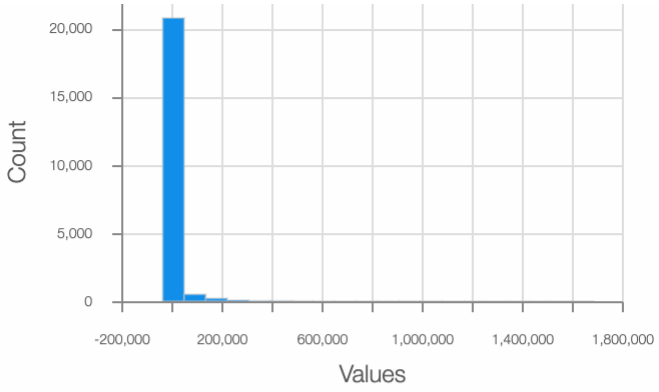
Max:

Median:

St. Dev: 9

sqft_lot

Num. Rows:



Num. Unique:

Missing:

Mean: 15,1

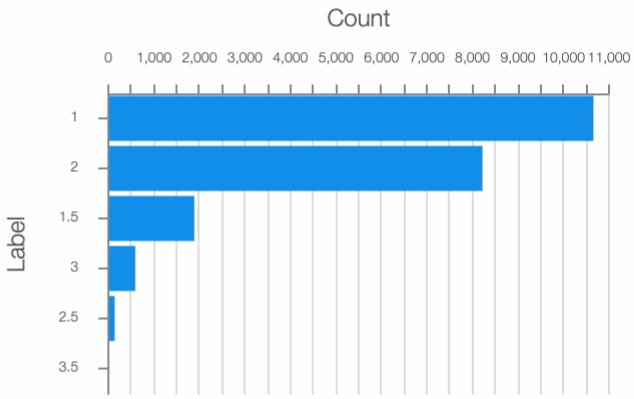
Min:

Max:

Median:

St. Dev: 41,4

floors



Num. Rows:

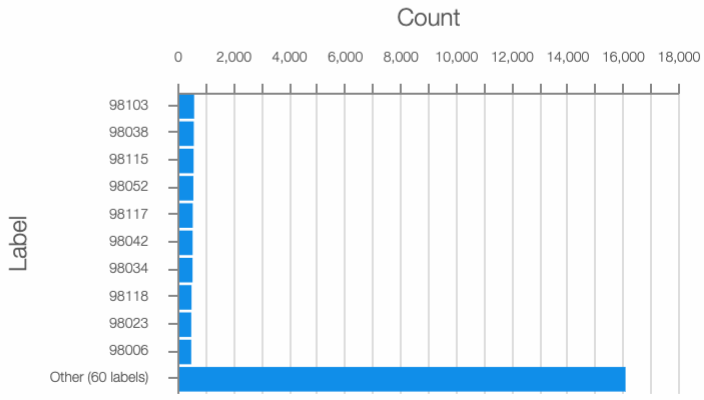
Num. Unique:

Missing:

Frequent Items

1
2
1.5
3
2.5

zipcode



Num. Rows:

Num. Unique:

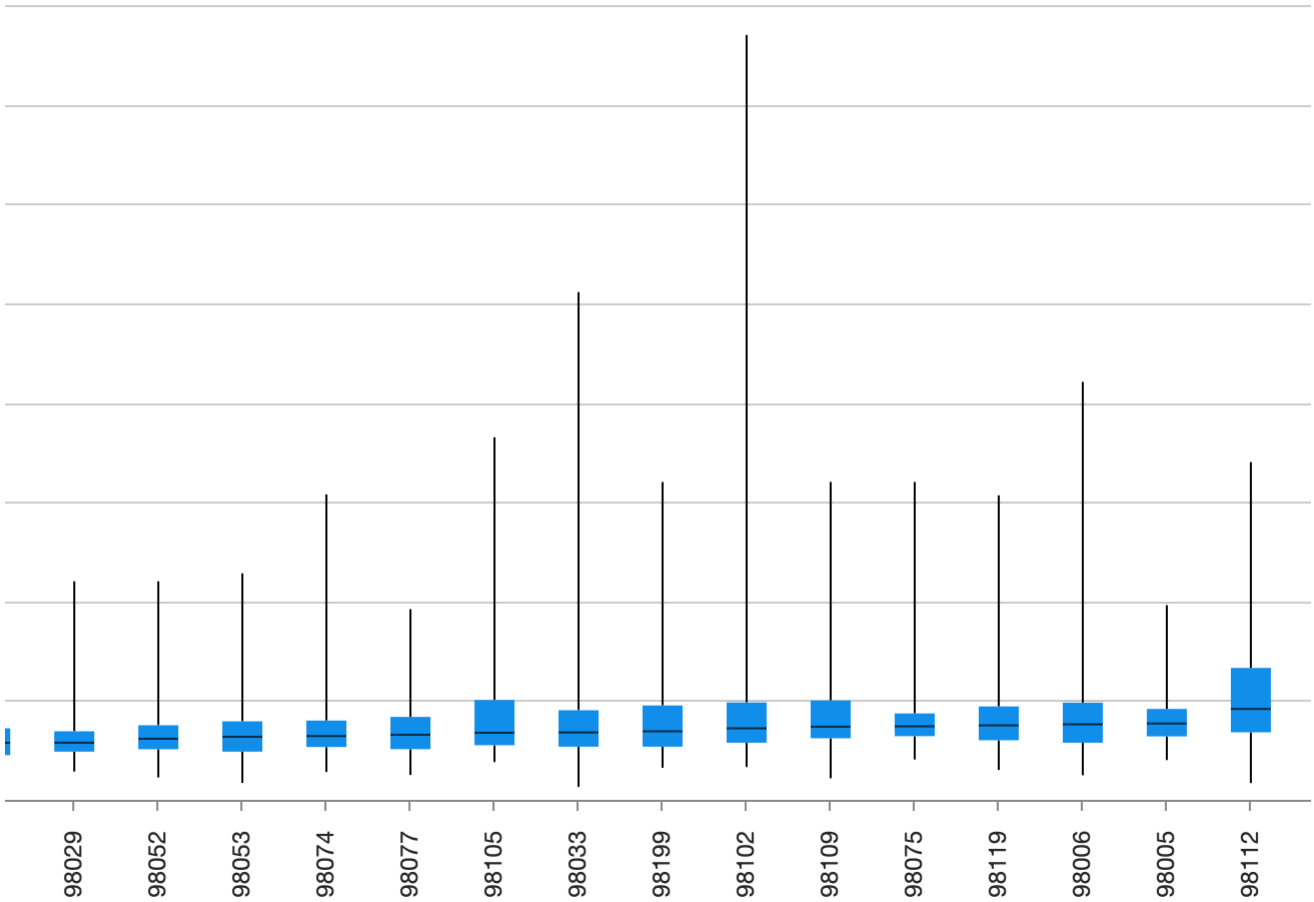
Missing:

Frequent Items

98103
98038
98115
98052
98117

```
turicreate.show(sales['zipcode'],sales['price'])
```

```
Materializing X axis SArray
Materializing Y axis SArray
```



Build a model with these additional features

```
my_features_model = turicreate.linear_regression.create(training_set,target='price'
```

```
PROGRESS: Creating a validation set from 5 percent of training data. This may
          You can set ``validation_set=None`` to disable validation tracking.

Linear regression:
--------------------------------------------------------
Number of examples          : 16514
```

Compare simple model with more complex one

```
print (my_features)
```

```
['bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot', 'floors', 'zipcode']
```

```
print (sqft_model.evaluate(test_set))
print (my_features_model.evaluate(test_set))
```

```
{'max_error': 4141808.6553260004, 'rmse': 255195.34326236605}
{'max_error': 3524061.710787955, 'rmse': 179605.63282317846}
```

Apply learned models to make predictions

```
house1 = sales[sales['id']=='5309101200']
```

```
house1
```

id	date	price	bedrooms	bathrooms	sqft_living	sc
5309101200	2014-06-05 00:00:00+00:00	620000	4	2.25	2400	5

view	condition	grade	sqft_above	sqft_basement	yr_built	yr_renovated	
0	4	7	1460	940	1929	0	

long	sqft_living15	sqft_lot15
-122.37010126	1250.0	4880.0

[? rows x 21 columns]
Note: Only the head of the SFrame is printed. This SFrame is lazily evaluated.
You can use sf.materialize() to force materialization

```
print (house1['price'])
```

```
    [620000, ... ]
```

```
print (sqft_model.predict(house1))
```

```
    [629698.4737817376]
```

```
print (my_features_model.predict(house1))
```

```
    [723966.3314462015]
```

Prediction for a second house, a fancier one

```
house2 = sales[sales['id']=='1925069082']
```

```
house2
```

id	date	price	bedrooms	bathrooms	sqft_living	
1925069082	2015-05-11 00:00:00+00:00	2200000	5	4.25	4640	

view	condition	grade	sqft_above	sqft_basement	yr_built	yr_renovated	
4	5	8	2860	1780	1952	0	

long	sqft_living15	sqft_lot15
-122.09722322	3140.0	14200.0

```
print (sqft_model.predict(house2))
```

```
[1261771.1080672739]
```

```
print (my_features_model.predict(house2))
```

```
[1477158.0083541381]
```

Prediction for a super fancy home

```
bill_gates = {'bedrooms':[8],
              'bathrooms':[25],
              'sqft_living':[50000],
              'sqft_lot':[225000],
              'floors':[4],
              'zipcode':['98039'],
              'condition':[10],
              'grade':[10],
```

```
                'waterfront':[1],
                'view':[4],
                'sqft_above':[37500],
                'sqft_basement':[12500],
                'yr_built':[1994],
                'yr_renovated':[2010],
                'lat':[47.627606],
                'long':[-122.242054],
                'sqft_living15':[5000],
                'sqft_lot15':[40000]}
```

```
print (my_features_model.predict(turicreate.SFrame(bill_gates)))
```

```
    [13583428.045934515]
```

✓ 0초 오전 1:54에 완료됨