California Property Insights

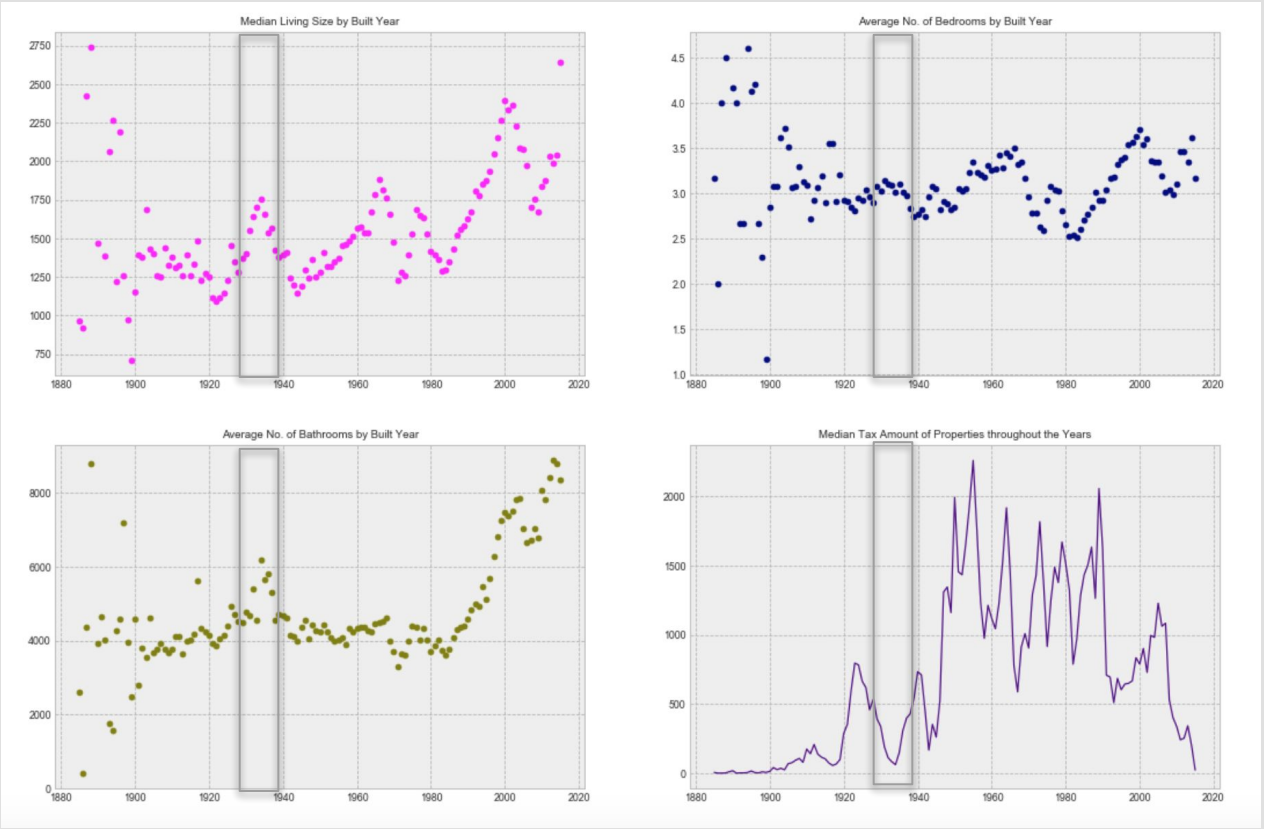
Sara and Mina

Introduction

Our client is looking for a historic property in California.

Using data from 2016 and 2017, we are predicting how historic properties behave in the market.

Isolating the Optimal Time Period



Properties built in the 1930s show promise in delivering:

Historic Value

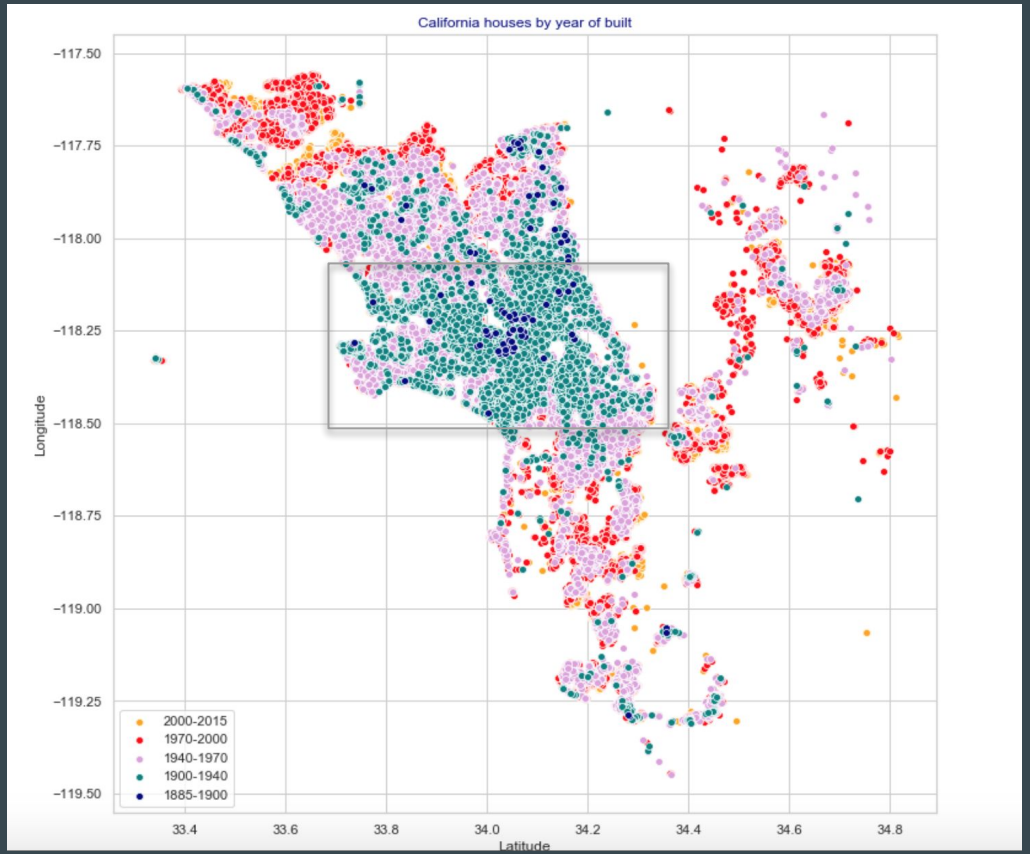
Large Living Space

Low Tax

Where to find historic houses?

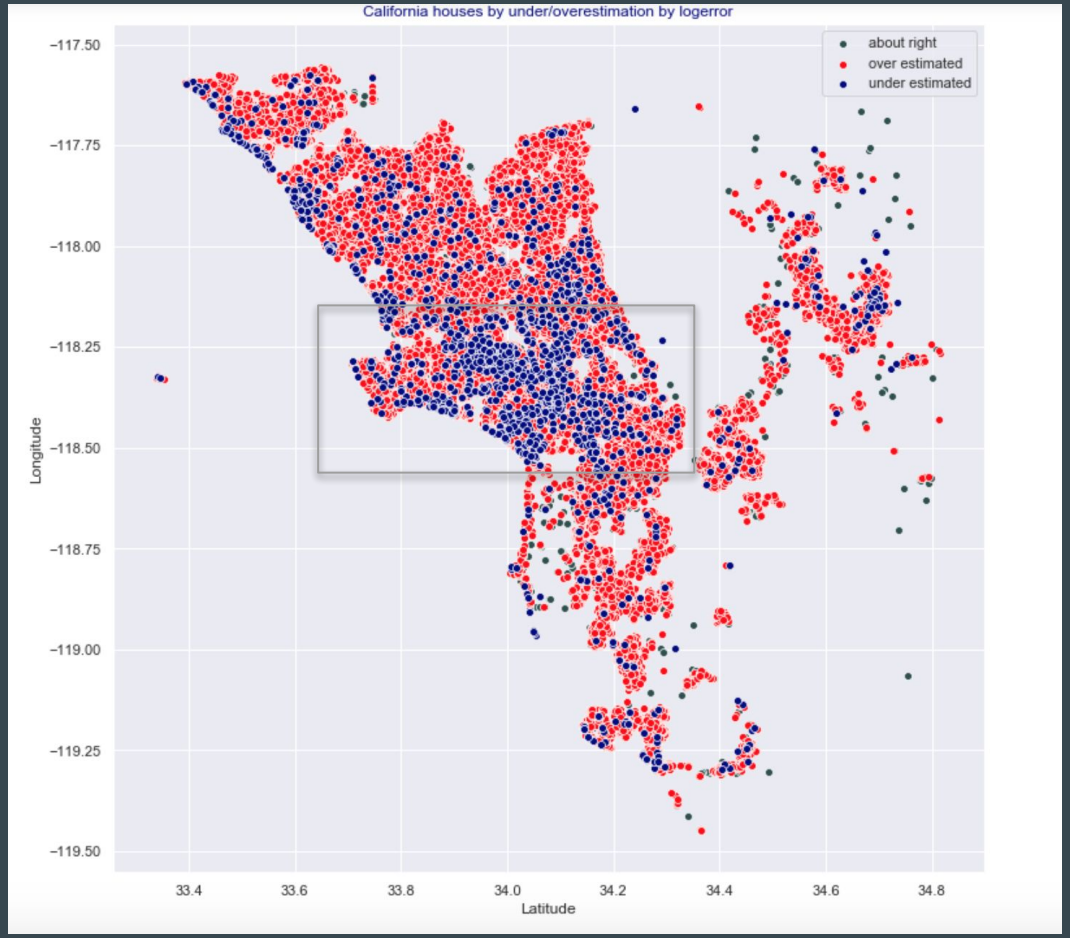
The green dots depict property built between 1900 - 1940.

Heavily concentrated in central California.



The blue dots show properties that were sold for more than they were listed.

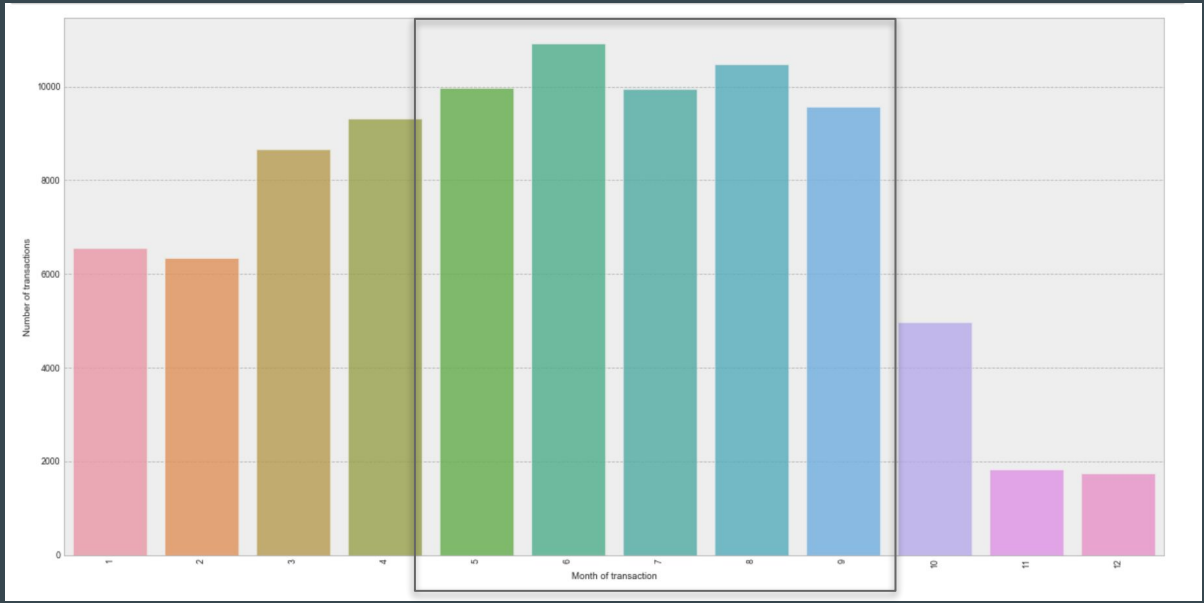
Fast bidding movements on desirable historic properties would be advised.



When is the best time to buy?

The months of May - September.

Causality is unclear however client will have largest variety of properties to consider if purchasing that time period.



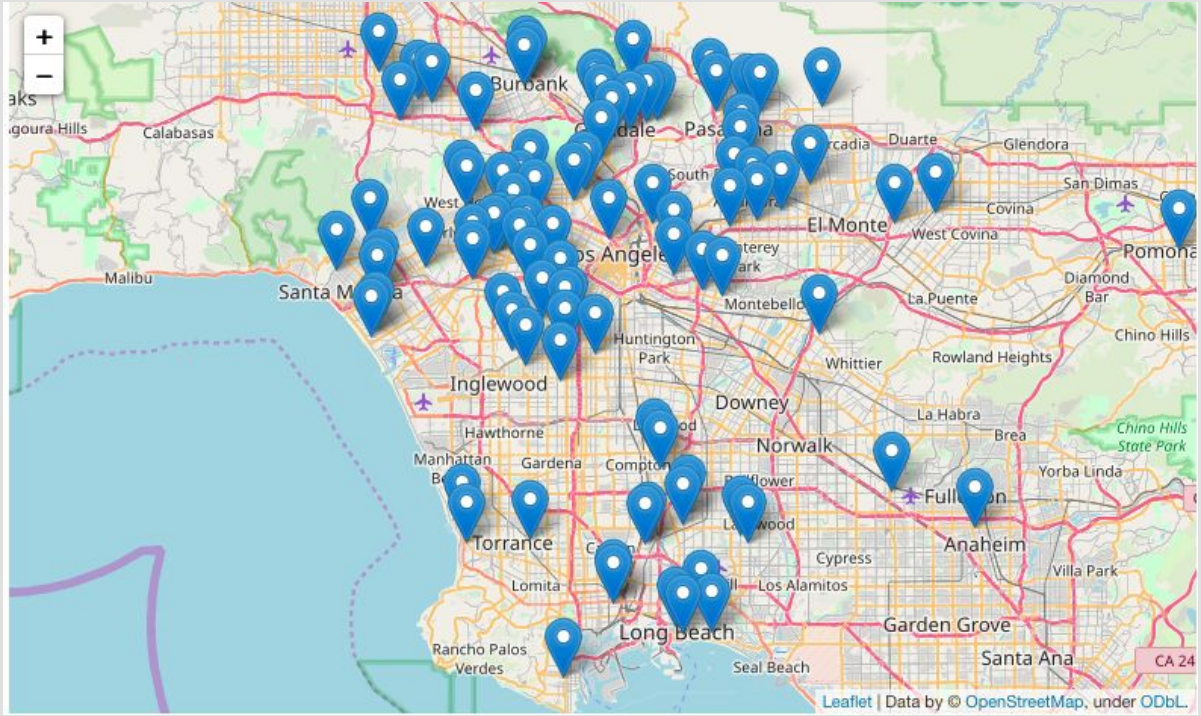
Examples of properties that fit into the client niche.

Sample of 100 properties (out of 5526) which are:

Built between 1930-1940.

Have four or more bedrooms.

Sold for less than the asking value.




MEMORIAL

In Conclusion

- We can predict that it is very likely that our client can find the exact property he desires.
- He is also able to find the property and purchase it for less than it is listed.

Further Work:

Would include analysis of the oldest properties (pre 1900s) as a subset to identify any trends in their behaviour in the market.

Thank you for watching

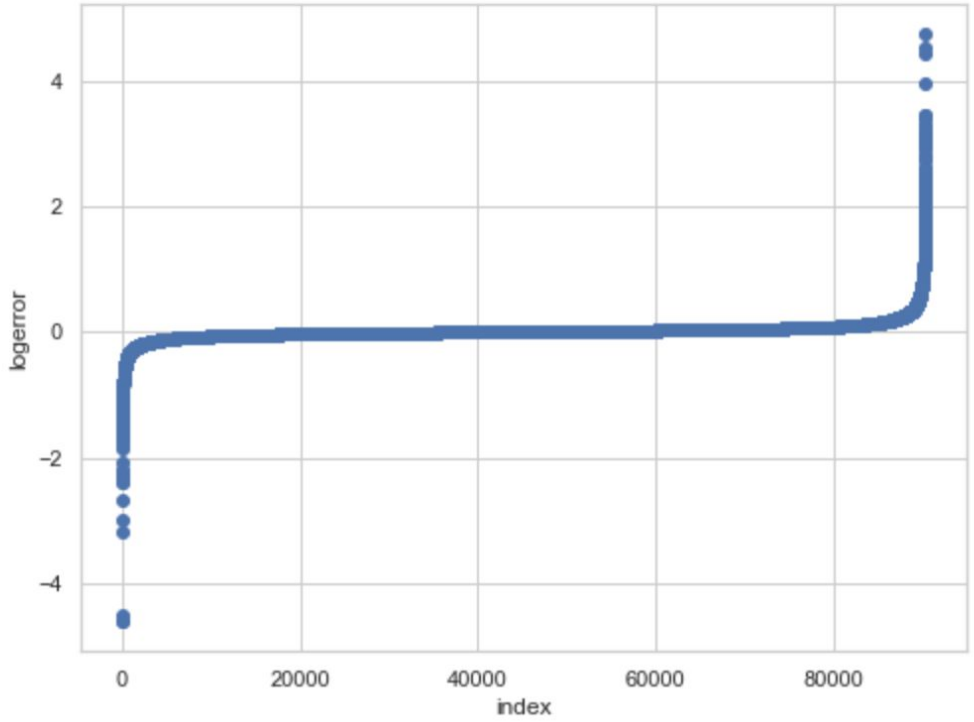
Q & A

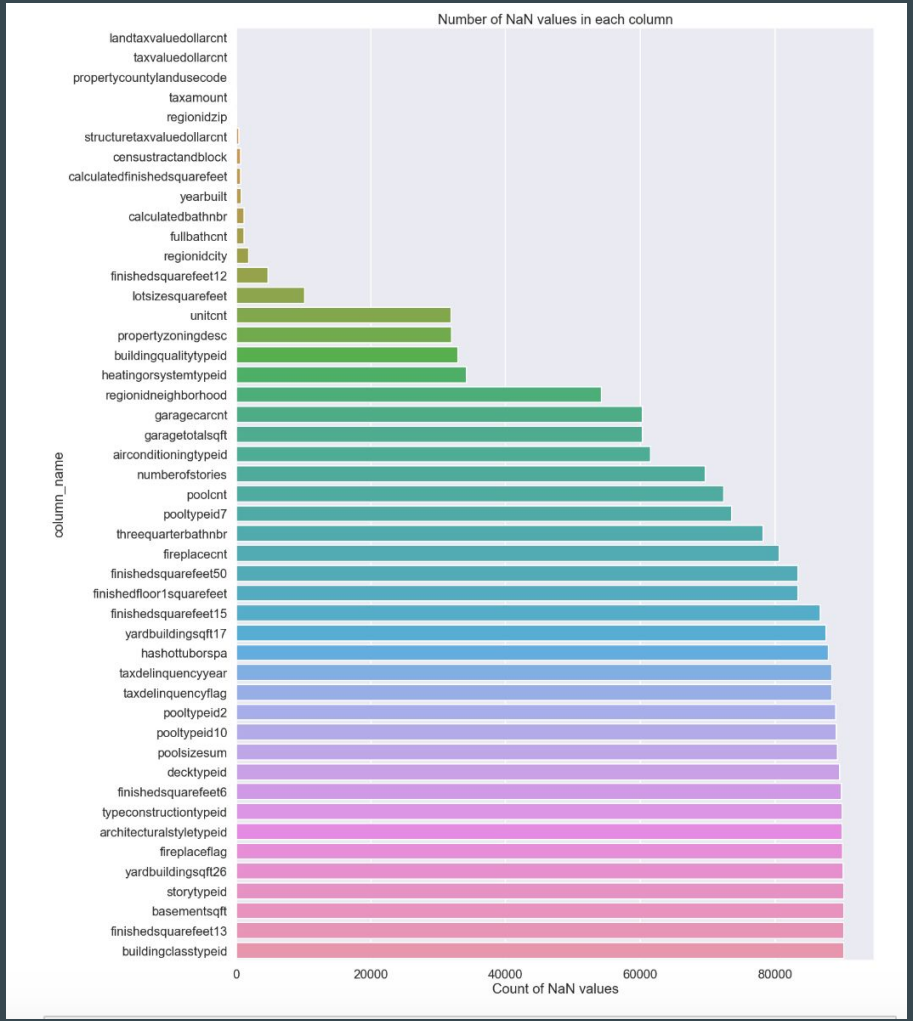
Dep. Variable:	logerror	**R-squared:**	0.006
Model:	OLS	**Adj. R-squared:**	0.006
Method:	Least Squares	**F-statistic:**	95.06
Date:	Wed, 04 Dec 2019	**Prob (F-statistic):**	3.56e-100
Time:	11:05:59	**Log-Likelihood:**	68308.
No. Observations:	75761	**AIC:**	-1.366e+05
Df Residuals:	75755	**BIC:**	-1.365e+05
Df Model:	5		
Covariance Type:	nonrobust		

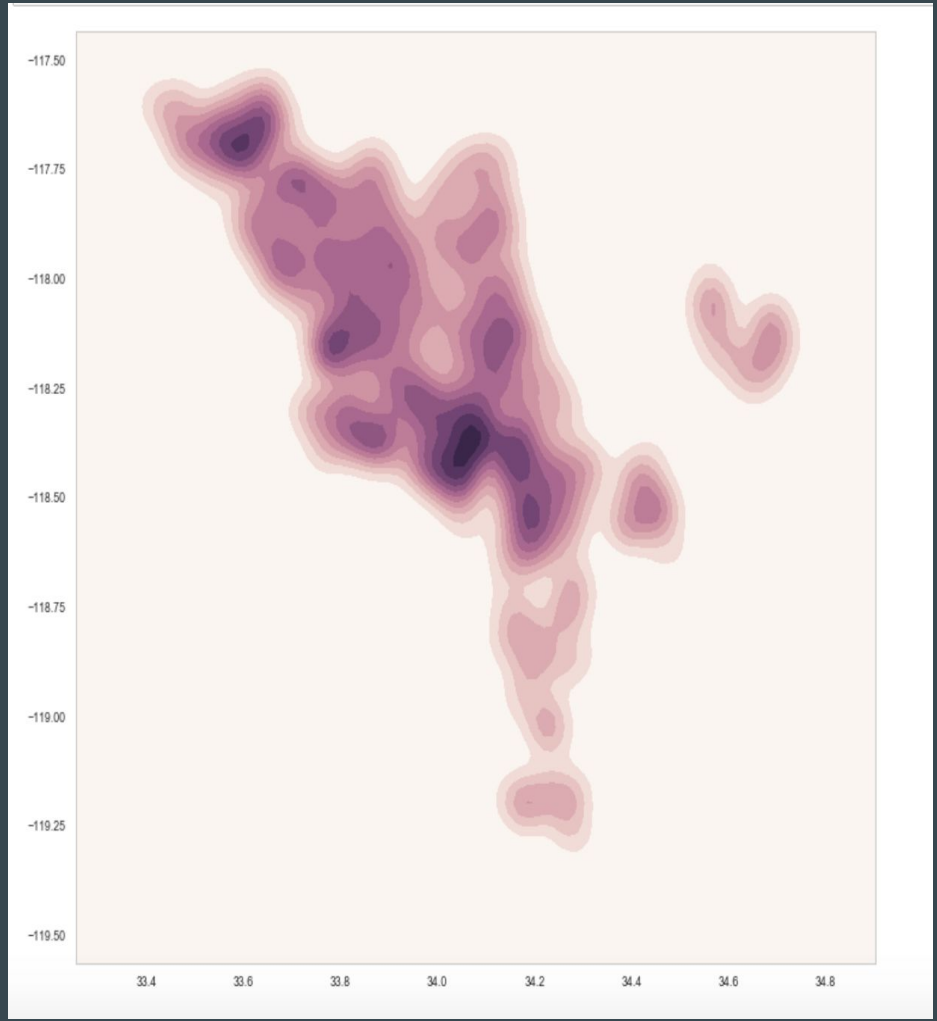
	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-0.0351	0.033	-1.065	0.287	-0.100	0.029
finishedsquarefeet12	1.067e-05	6.43e-07	16.602	0.000	9.41e-06	1.19e-05
yearbuilt	1.811e-05	1.68e-05	1.075	0.282	-1.49e-05	5.11e-05
taxamount	-1.056e-06	6.49e-08	-16.279	0.000	-1.18e-06	-9.29e-07
lotsizesquarefeet	4.751e-09	2.94e-09	1.616	0.106	-1.01e-09	1.05e-08
bedroomcnt	-0.0008	0.000	-1.619	0.105	-0.002	0.000

Omnibus:	19144.464	**Durbin-Watson:**	1.986
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	159142.040
Skew:	0.985	**Prob(JB):**	0.00
Kurtosis:	9.822	**Cond. No.**	1.17e+07

In [43]:
```
# plot logerror
plt.figure(figsize=(8,6))
plt.scatter(range(df_train.shape[0]), np.sort(df_train.logerror.values))
plt.xlabel('index')
plt.ylabel('logerror')
plt.show()
```




Number of NaN values in each column
landtaxvaluedollarcnt
taxvaluedollarcnt
propertycountylandusecode
taxamount
regionidzip
structuretaxvaluedollarcnt
censustractandblock
calculatedfinishedsquarefeet
yearbuilt
calculatedbathnbr
fullbathcnt
regionidcity
finishedsquarefeet12
lotsizesquarefeet
unitcnt
propertyzoningdesc
buildingqualitytypeid
heatingorsystemtypeid
regionidneighborhood
garagecarcnt
garagetotalsqft
airconditioningtypeid
numberofstories
poolcnt
pooltypeid7
threequarterbathnbr
fireplacecnt
finishedsquarefeet50
finishedfloor1squarefeet
finishedsquarefeet15
yardbuildingsqft17
hashottuborspa
taxdelinquencyyear
taxdelinquencyflag
pooltypeid2
pooltypeid10
poolsizesum
decktypeid
finishedsquarefeet6
typeconstructiontypeid
architecturalstyletypeid
fireplaceflag
yardbuildingsqft26
storytypeid
basementsqft
finishedsquarefeet13
buildingclasstypeid
column_name
0
20000
40000
60000
80000
Count of NaN values


−117.50
−117.75
−118.00
−118.25
−118.50
−118.75
−119.00
−119.25
−119.50
33.4
33.6
33.8
34.0
34.2
34.4
34.6
34.8