Predicting House Prices

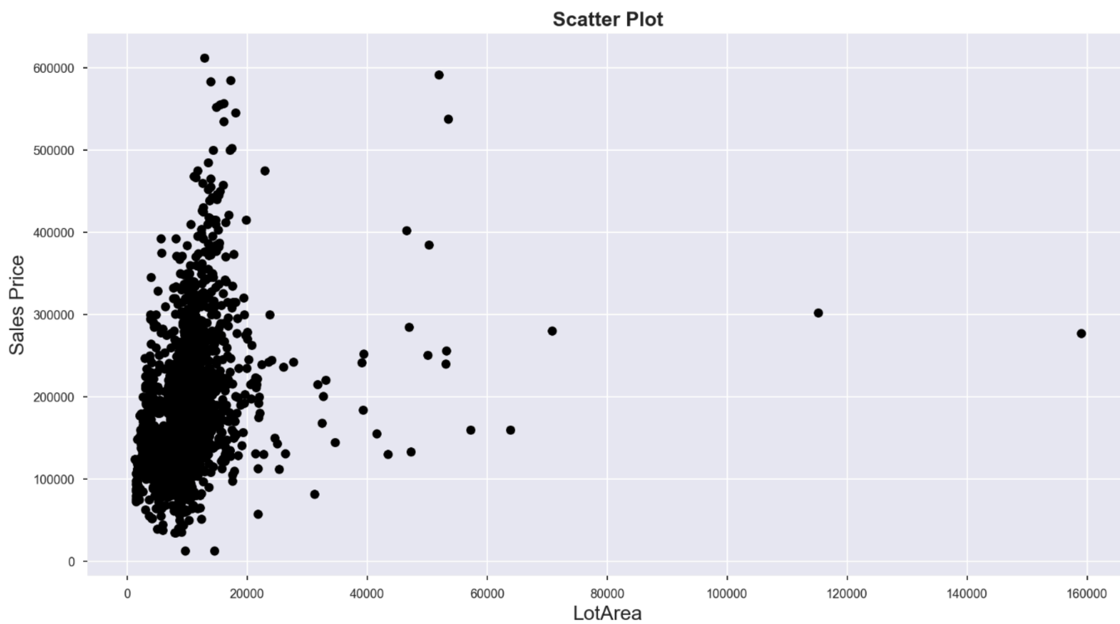
in Ames, Iowa

Keith, Clement, Lynn

Objective

Identifying factors that are statistically significant in predicting house prices in Ames

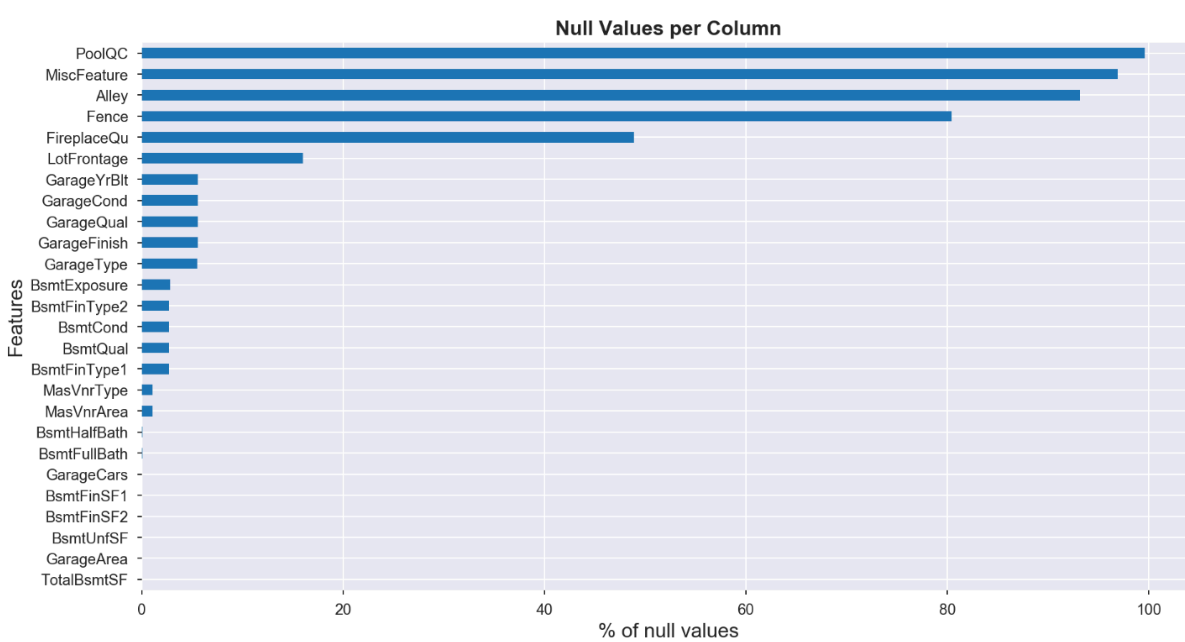
1. Removal of anomalies




```
# Visually check for presence of anomalies in Lot Area
scatter('LotArea')
```

```
#Drop anomalies where lot area >70,000 - left 2048 observations
train.drop(train[train['LotArea']>70000].index,inplace=True)
```

2. Handling of null values and categorical variables

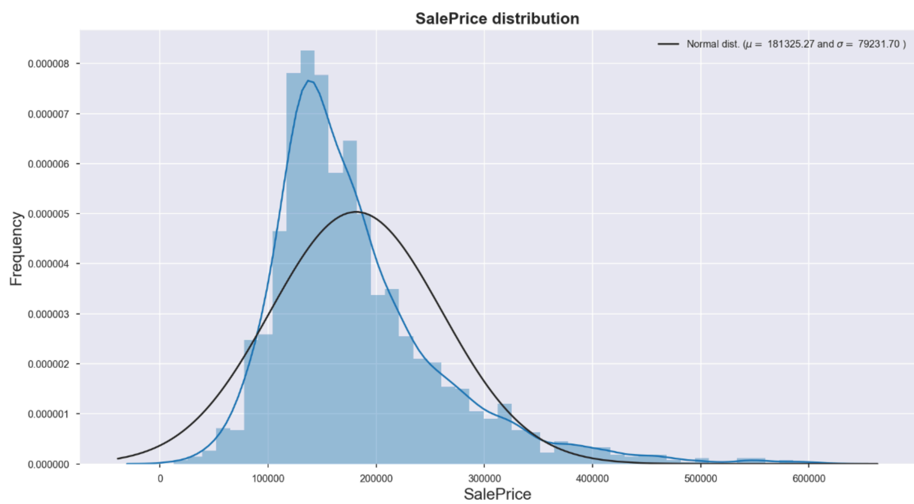



```
#fill NaNs with 'NA', where it is already an option
for i in ['MiscFeature','Alley','Fence','GarageType','BsmtFinType2','BsmtFinType1']:
    clean_check_col(train, i,'NA')
    clean_check_col(test, i,'NA')

#fill NaNs with 'None', where it is already an option
for i in ['MasVnrType']:
    clean_check_col(train, i,'None')
    clean_check_col(test, i,'None')
```

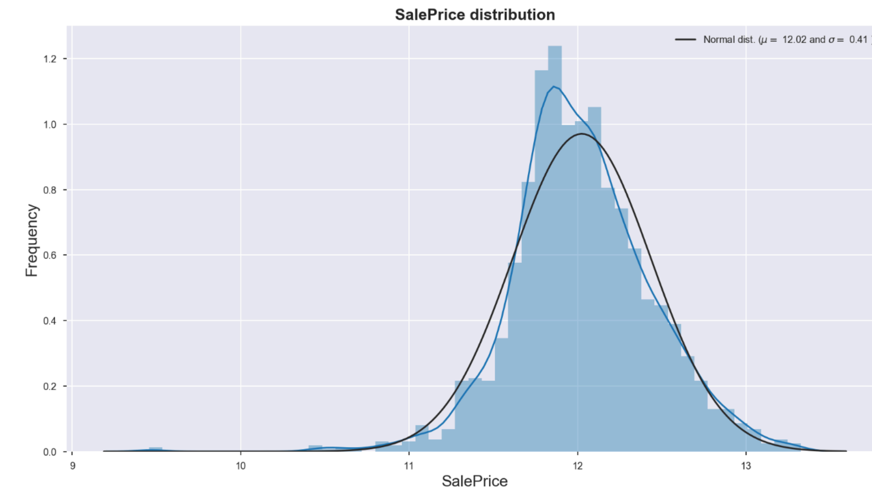
```
#fill NaNs with mode for discrete variables
for i in ['GarageYrBlt','BsmtHalfBath','BsmtFullBath','GarageCars']:
    clean_check_col(train, i,int(train[i].mode()))
    clean_check_col(test, i,int(train[i].mode()))
```

```
quality_cols = ['ExterQual','ExterCond','KitchenQual','BsmtQual','HeatingQC','FireplaceQu',
                'GarageQual','GarageCond','BsmtCond','PoolQC']
dict_values = {'Ex': 5,
               'Gd' : 4,
               'TA' : 3,
               'Fa' : 2,
               'Po' : 1,
               'NA' : 0 }
for col in quality_cols:
    clean_quality_col(train, col, dict_values, "NA")
    clean_quality_col(test, col, dict_values, "NA")
```

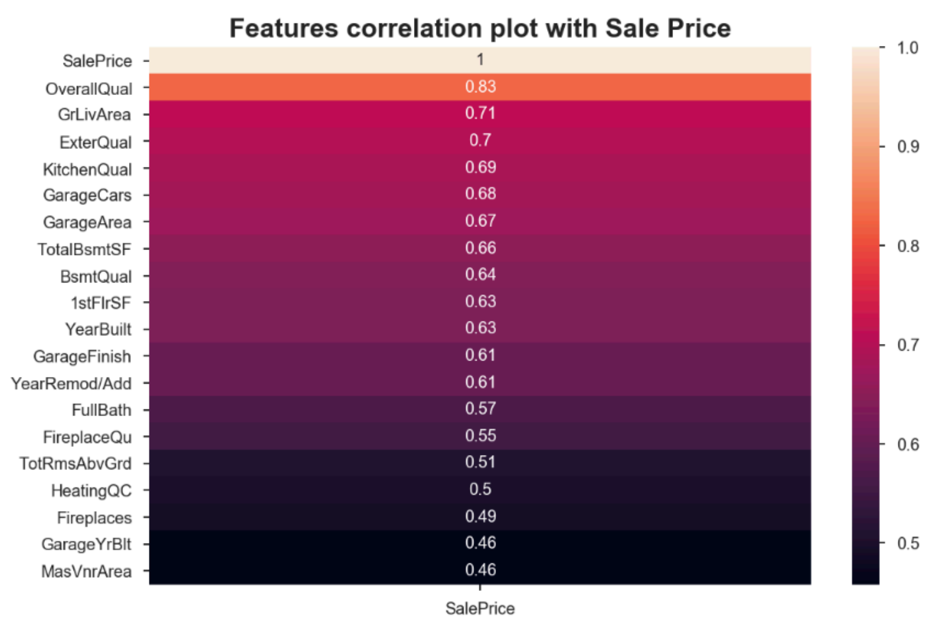
3. Normalisation of dependent variable




```
#log transform the target:
train["SalePrice"] = np.log1p(train["SalePrice"])
```



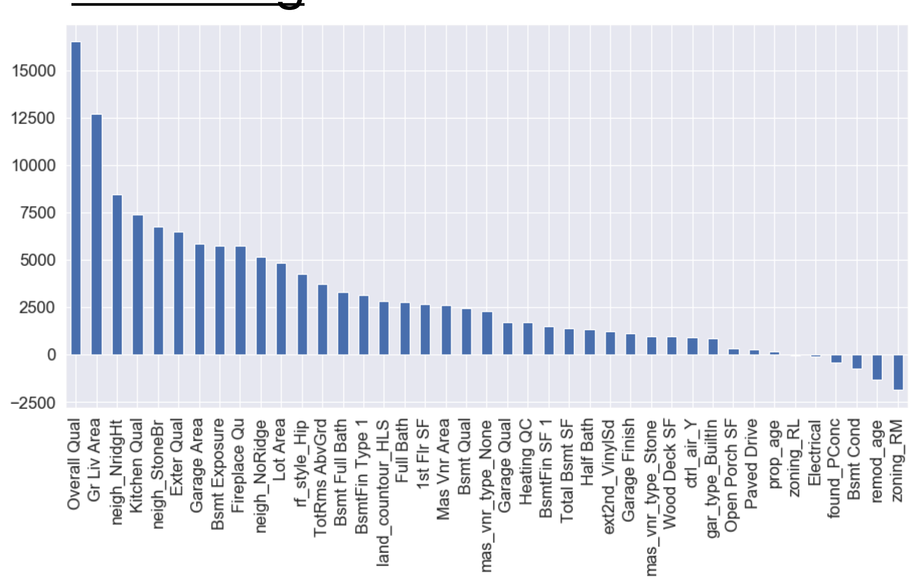
4. Initial feature selection




```
#Finding top 20 strongly correlated features with Sale Price, ordered by descending order
corrmat = train.corr()
columns = abs(corrmat['SalePrice']).sort_values(ascending=False).head(20)
columns
```

```
#Correlation heatmap of top 20 strongly correlated features with SalesPrice
plt.figure(figsize=(9,6))
plt.title('Features correlation plot with Sale Price',fontsize=16,fontweight='bold');
sns.heatmap(pd.DataFrame(columns.head(20)),annot=True);
```

5. Modelling

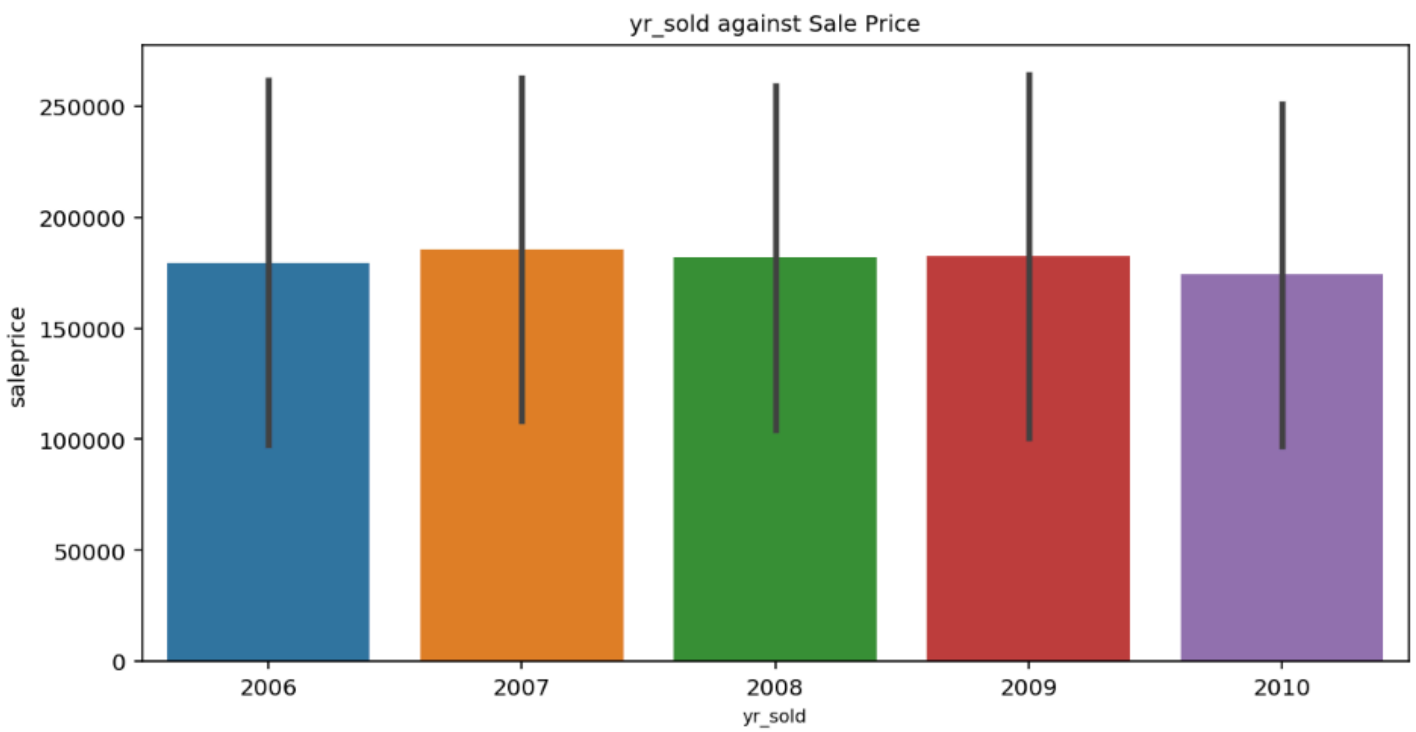



```
#Converting selected features with categorical values into dummy
train_x = pd.get_dummies(train_x)
test_x = pd.get_dummies(test_x)
```

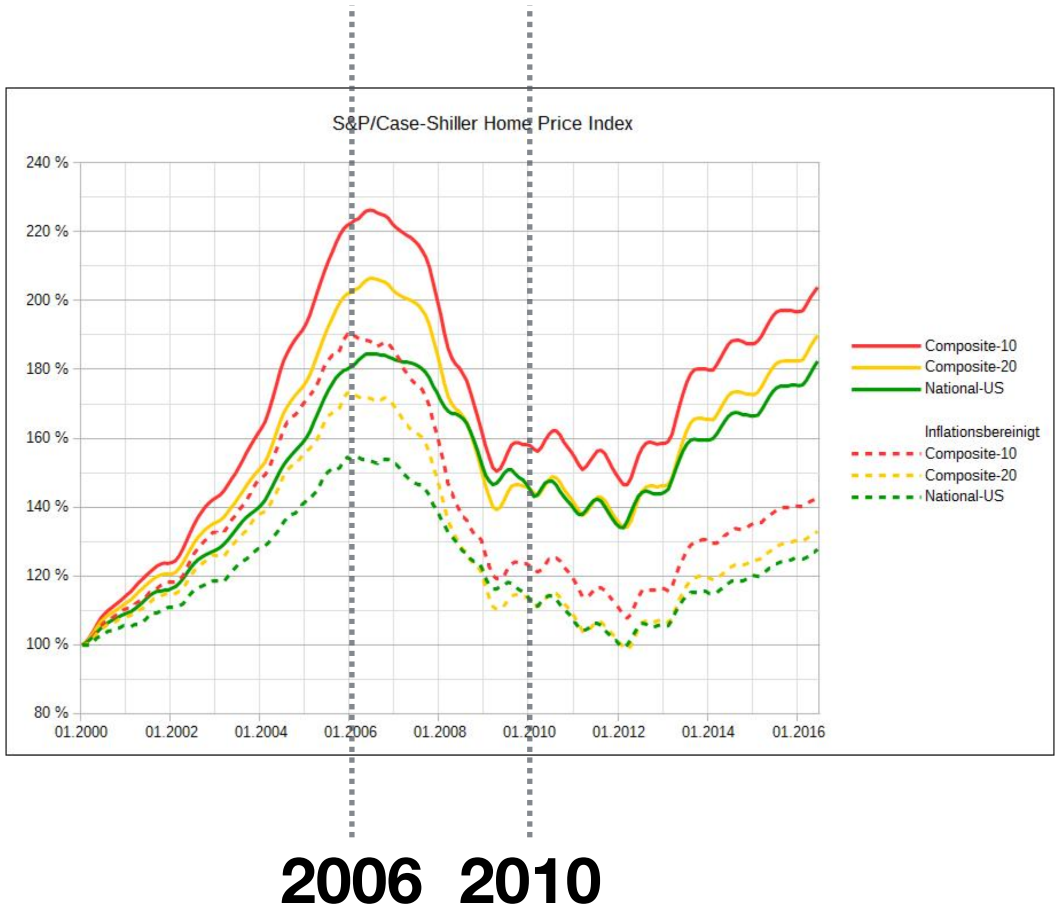
```
#initiating, fitting and transforming using polynomial features
poly = PolynomialFeatures(include_bias=False)

train_x = poly.fit_transform(train_x)
test_x = poly.fit_transform(test_x)
```

```
#initiating linear regerssion, lasso and ridge, and cross-validated to find model with best fit.
lr = LinearRegression()
lasso = LassoCV(n_alphas=20)
ridge = RidgeCV(alphas=np.linspace(.1, 10, 200))

lr_scores = cross_val_score(lr, train_x, train_y, cv=3)
print('Linear Regression Rsquared score is {}'.format(lr_scores.mean()))

lasso_scores = cross_val_score(lasso, train_x, train_y, cv=3)
print('Lasso Regression Rsquared score is {}'.format(lasso_scores.mean()))

ridge_scores = cross_val_score(ridge, train_x, train_y, cv=3)
print('Ridge Regression Rsquared score is {}'.format(ridge_scores.mean()))
```

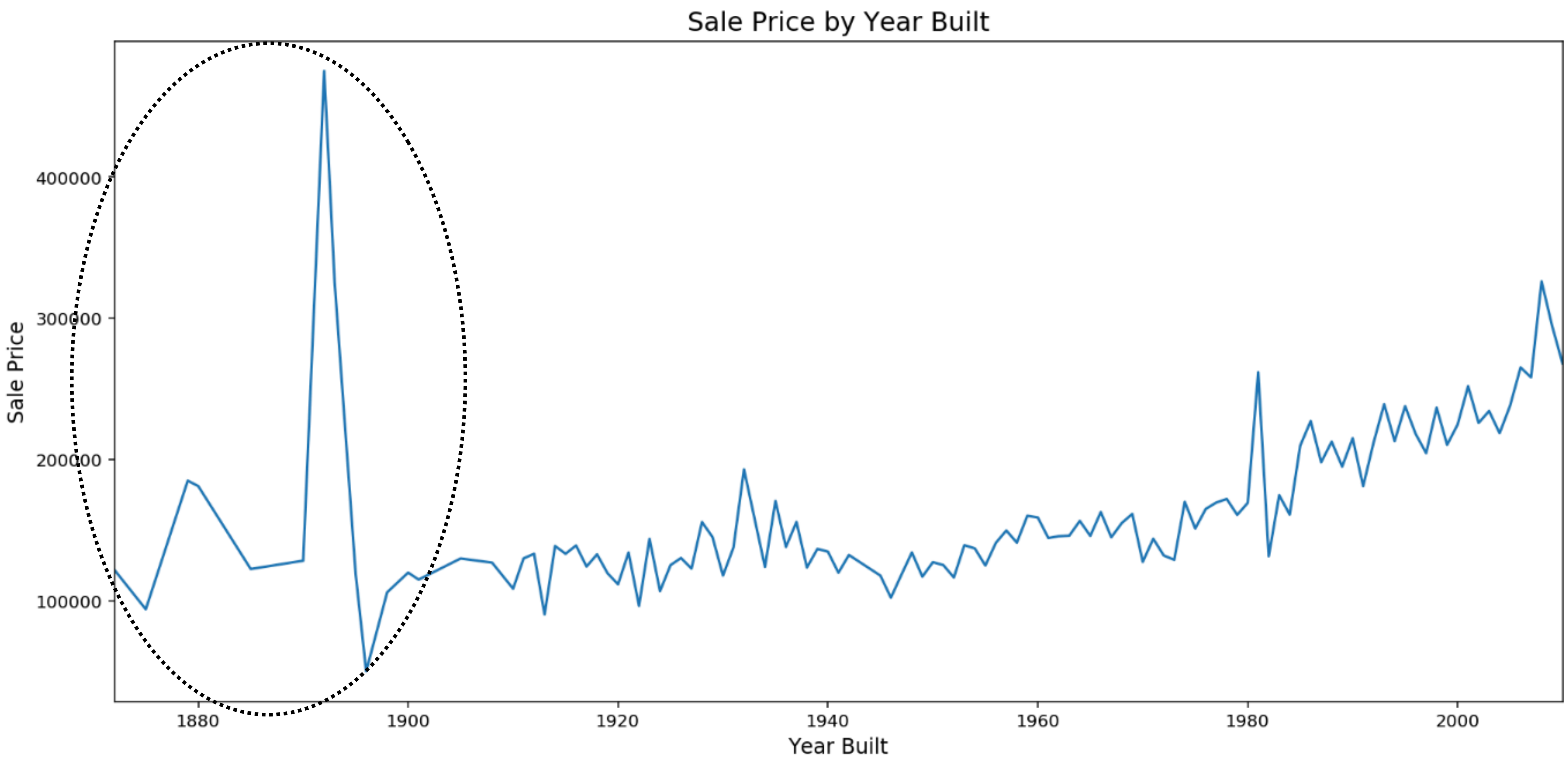
Trend in house prices



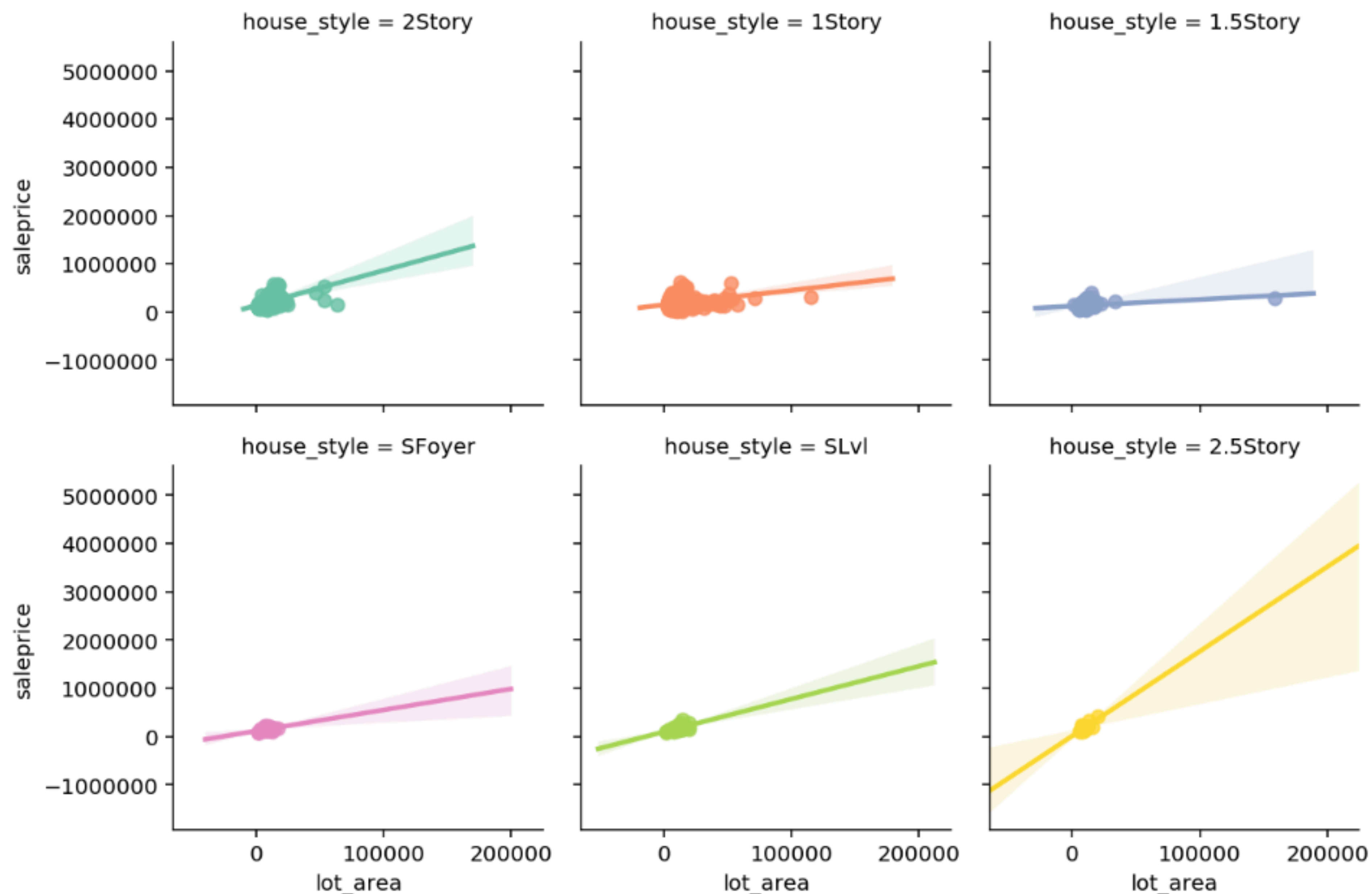
US Great Recession



Price changes based on year built



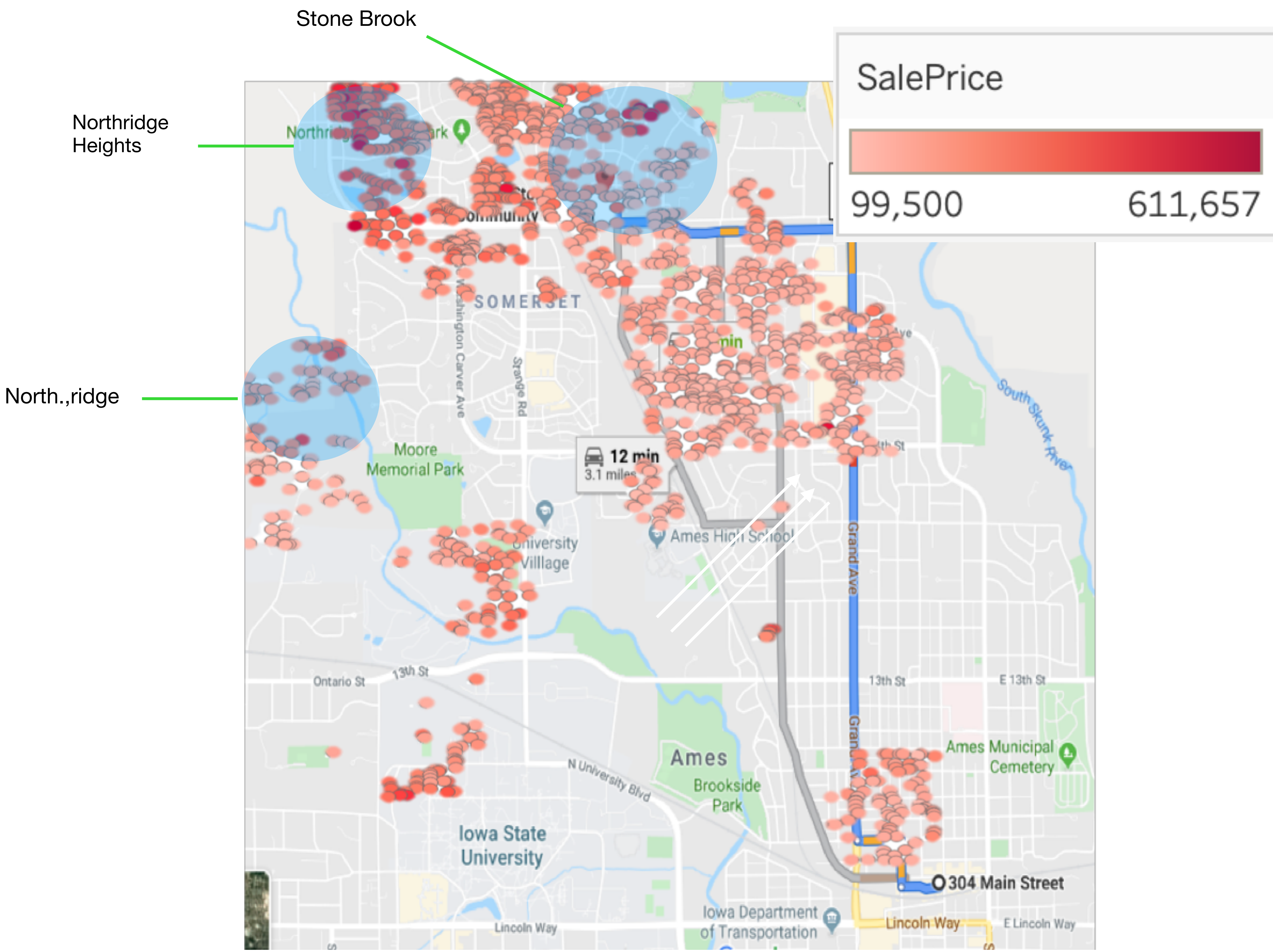
Price changes based on house styles



Conclusions

- Ridge Regression the best regression technique
- Top features
    - Overall Quality
    - Some neighbourhoods. Eg. Stone Brooks
        - Proximity to schools, university, downtown

Neighbourhood/sale price visualisation



Limitations

- Model limitations
  - Cannot use same model to predict other areas
- Data limitations
  - Some data was transacted during the mortgage crisis
- Does not capture external factors - negotiation, personal preferences, developer discounts