California Apartments Renting Price Analysis

-- Linear Regression & ANOVA Analysis

Sicheng Zhu

Content

Introduction

- Craigslist Posts
- Data Types

Data Preprocessing

- Data Collecting and Cleaning
- Data Exploration
- Transformations

Model Building

- Main Effect Model
- Variable Selection
- Lasso & Ridge
- Model Comparison
- Outlier Detecting

Model Diagnostic

- Residuals
- Cross Validation
- Test Prediction

Introduction

- House Renting Information from Craigslist.com
- Web Scraping (*lxml.html, requests, requests_cache, re*)
- 21947 Observations
- Response Variable: price
- Dependent Variables:
    - Numerical: sqft, bedrooms, bathrooms
    - Categorical: pets, laundry, parking, county

```
title           21947 non-null object
text            21947 non-null object
latitude        21864 non-null float64
longitude       21864 non-null float64
city_text       20287 non-null object
date_posted     21947 non-null object
date_updated    8809 non-null object
price           21845 non-null float64
deleted         21948 non-null bool
sqft            16357 non-null float64
bedrooms        20900 non-null float64
bathrooms       20900 non-null float64
pets            21655 non-null object
laundry         21732 non-null object
parking         21649 non-null object
craigslist      21948 non-null object
place           21247 non-null object
city            20092 non-null object
state           21853 non-null object
county          21853 non-null object
```

Data Preprocessing (lxml.html, requests, requests_cache, re)

```
requests_cache.install_cache("../craigslist")
start_url = "https://sacramento.craigslist.org/d/apts-housing-for-rent/search/apa"

def scrape_front_page(url):
    response = requests.get(url)
    response.raise_for_status()
    html = lx.fromstring(response.text)
    html.make_links_absolute(url)
    # Get all <a> tags with class "result-title"
    links = html.xpath("//a[contains(@class, 'result-title')]/@href")
    next_page = html.xpath("//a[contains(@class, 'next')]/@href")[0]

    return next_page, links

next_page, links = scrape_front_page(start_url)
```

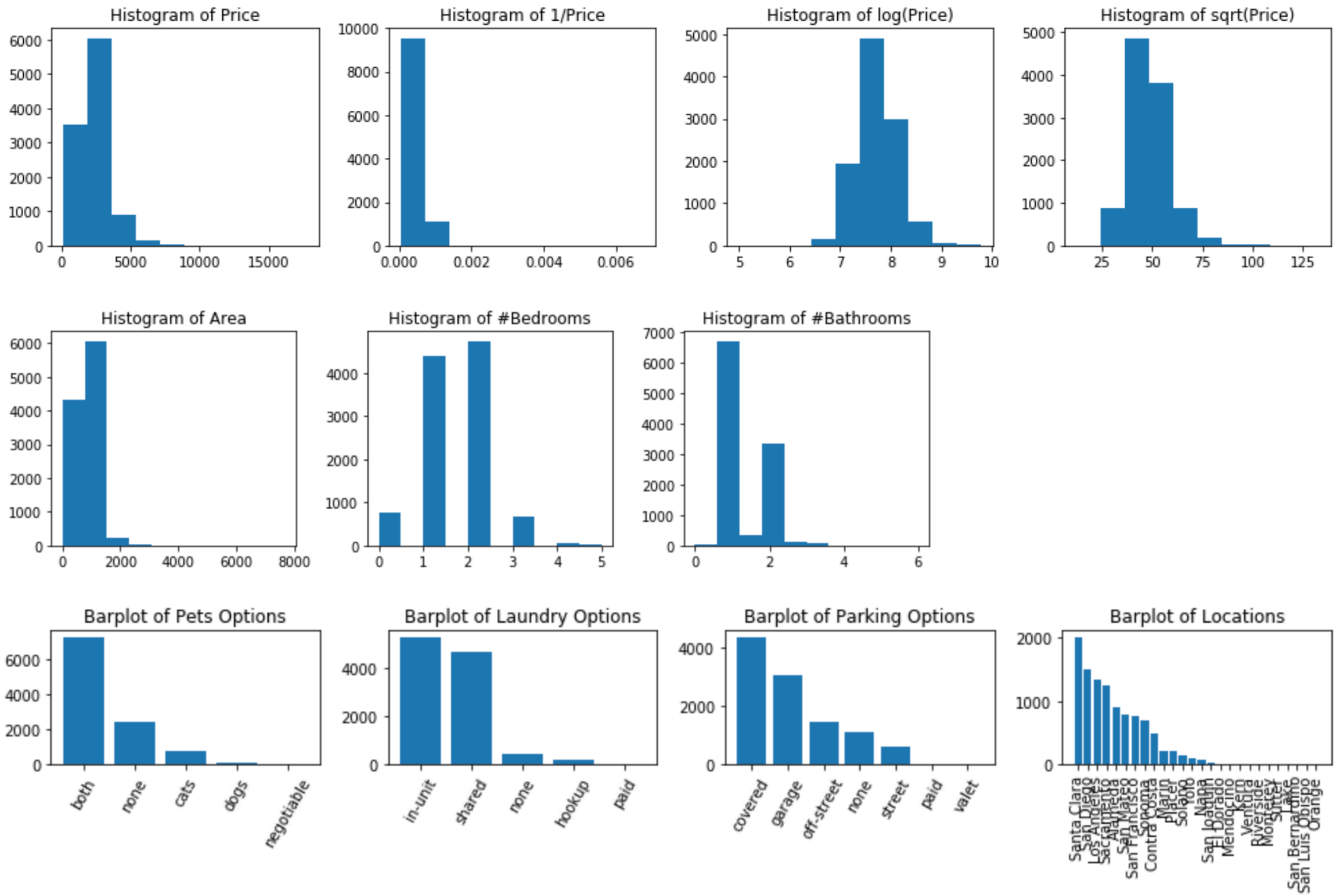
```
price = html.xpath("//*[contains(@class, 'price')]")[0]
title = html.cssselect("#titletextonly")[0].text_content()
attribs = [x.text_content() for x in html.xpath("//p[contains(@class, 'attrgroup')]/span")]
coords = html.cssselect("#map")[0]
lon = coords.attrib.get("data-longitude")
lat = coords.attrib.get("data-latitude")
text = html.cssselect("#postingbody")[0].text_content()
```

Data Cleaning

- Subset the data frame for modeling
- Missing Response Variables
- Wrong Response Variables
- Missing values in numerical variables
- Missing values in categorical variables
- **Train Test Split**

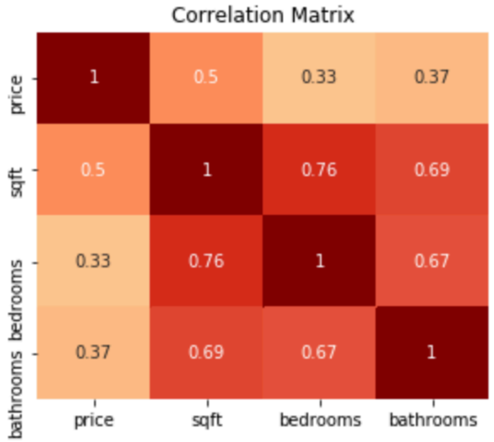
```
RangeIndex: 15876 entries, 0 to 15875
Data columns (total 9 columns):
Unnamed: 0     15876 non-null int64
price          15876 non-null int64
sqft           15876 non-null int64
bedrooms       15876 non-null int64
bathrooms      15876 non-null float64
pets           15876 non-null object
laundry        15876 non-null object
parking        15876 non-null object
county         15876 non-null object
```

Data Exploration

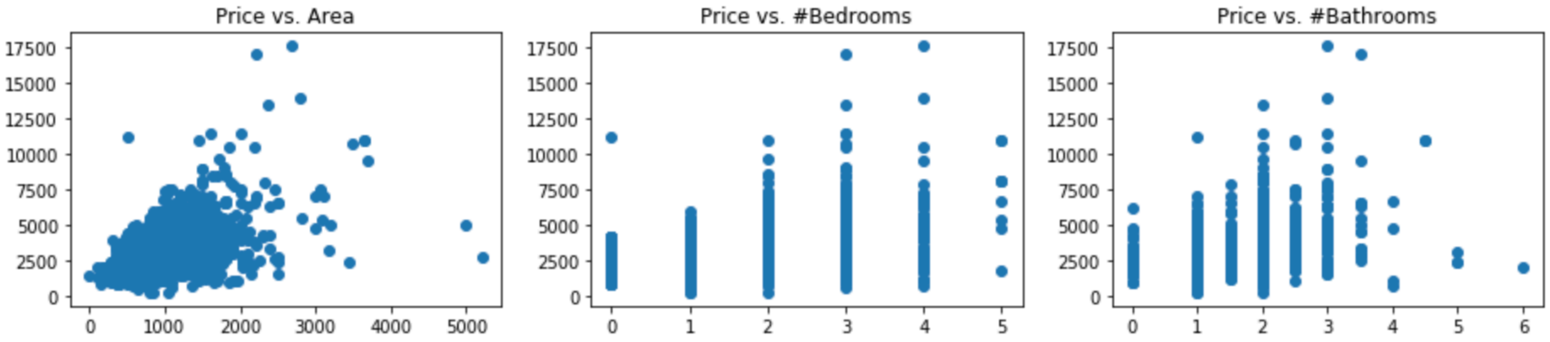


Data Exploration

- Pairwise Correlation
  - $r_{XY} = \dfrac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sum(X_i - \bar{X})^2 \sum(Y_i - \bar{Y})^2}$

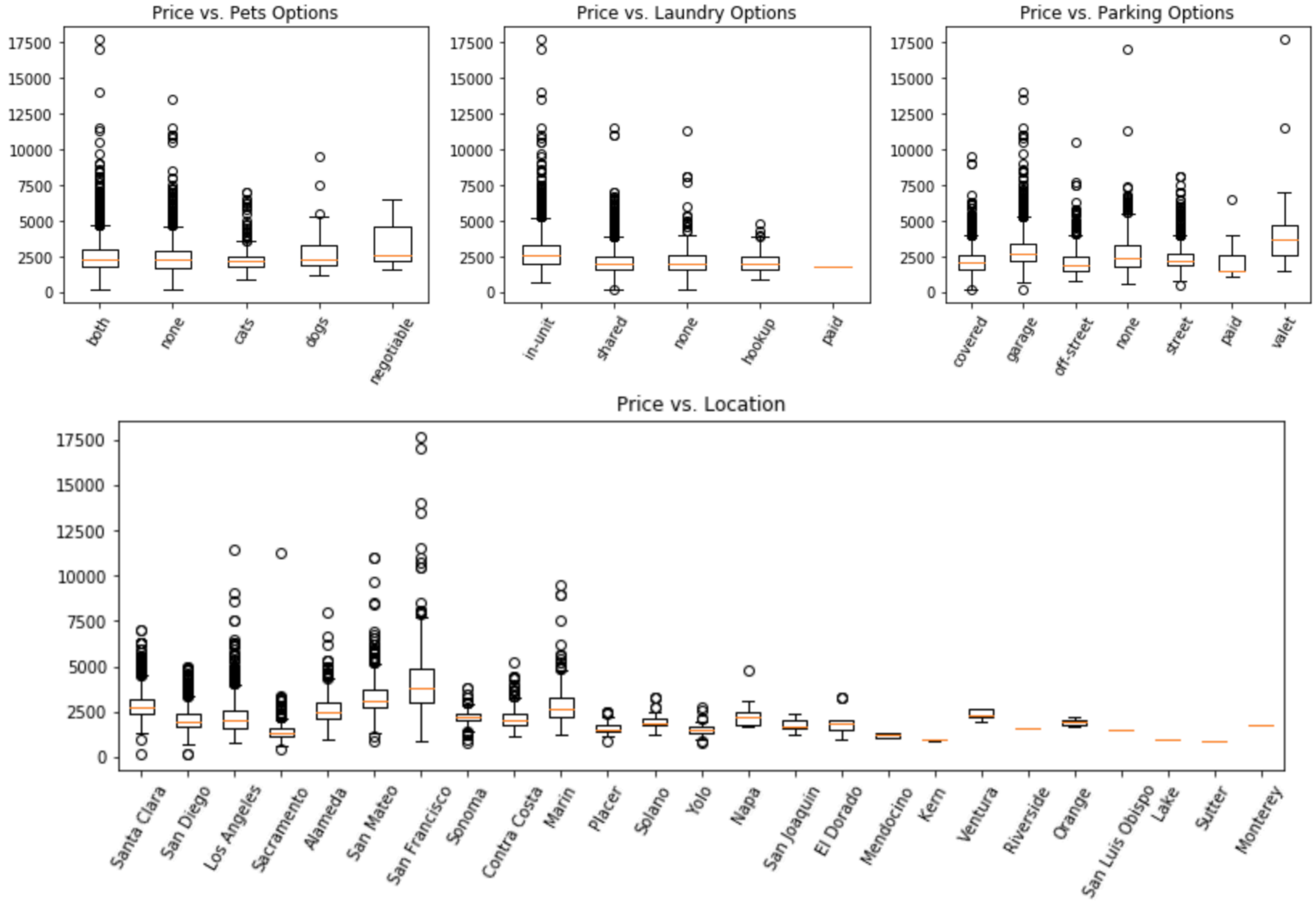


- Pairwise Scatter Plots



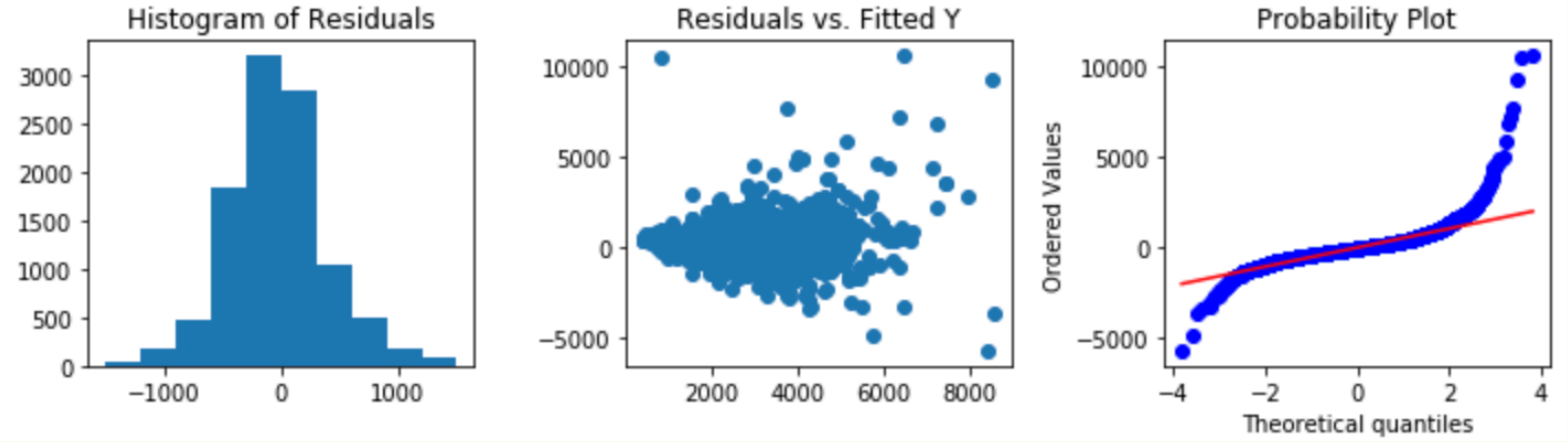
Data Exploration

- Price Distribution of Different Groups



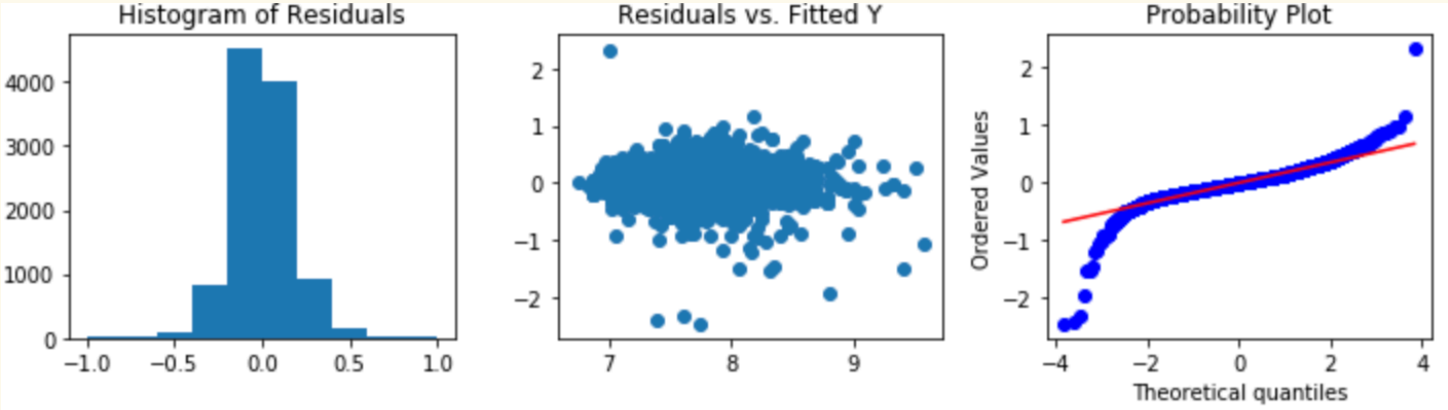
Assumptions Checking

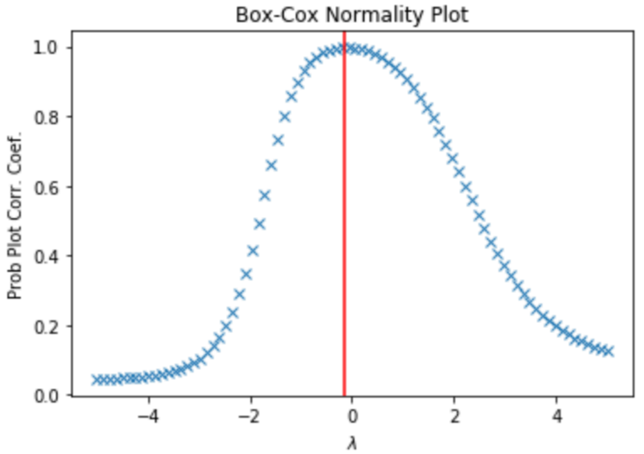
- Initial Model: Price = Sqft + Bedrooms + Bathrooms + factor(Pets)
  + factor(Laundry) + factor(Parking) + factor(County)



Data Preprocessing — Box-Cox Transformation

- $Y_i^{(\lambda)} = \begin{cases} \frac{Y_i^{\lambda} - 1}{\lambda}, if\ \lambda \neq 0 \\ \log Y_i, if\ \lambda = 0 \end{cases}$
- Apply log transformation to Price





Data Preprocessing — Standardization

- $z_i = \frac{x_i - \bar{x}}{S_x}, z_i \sim (0, 1)$
- **Reduce bias** caused by different scales
- **Reduce collinearity** by reducing VIF
  - $VIF_k = \frac{1}{1 - R_k^2}$
  - R_k^2 is the correlation of determination of the linear model $X_k = \beta_0 + \beta_1 X_1 + \cdots + \beta_{k-1} X_{k-1} + \beta_{k+1} X_{k+1} + \beta_p X_p$

```
[vif(X_train.iloc[:, 0:3].values, i) for i in range(3)]
```

```
[12.30064180690518, 12.830346973628702, 1.1550296427952673]
```

```
[vif(X_train_S.iloc[:, 0:3].values, i) for i in range(3)]
```

```
[2.761267792510718, 2.5871404283815598, 2.1156008482348003]
```

```
X_train.describe()
```

	sqft	bedrooms	bathrooms
count	10636.000000	10636.000000	10636.000000
mean	869.235709	1.518052	1.367055
std	293.564447	0.755829	0.522823
min	3.000000	0.000000	0.000000
25%	680.000000	1.000000	1.000000
50%	835.000000	2.000000	1.000000
75%	1020.000000	2.000000	2.000000
max	5210.000000	5.000000	6.000000

Model Building :: Initial Model

- **Main Effect Model**
- Log(Price) = Sqft + Bedrooms + Bathrooms + factor(Pets)

  + factor(Laundry) + factor(Parking) + factor(County)

R^2	0.770	# Observation	10636
R^2_{adjust}	0.769	Df Residuals	10593
F-statistic	845.4	Df Model	42
P-value	0.000		

- **Strong Collinearity**
  - model.eigenvals

```
3.10899947e+00, 2.14595627e+00, 1.99626086e+00, 1.14793303e+00,
1.06003835e+00, 9.94822228e-01, 7.43261506e-27, 1.04813194e-27,
5.27318888e-28, 1.34333452e-28, 8.85244332e-29])
```

One-Way ANOVA Test for Categorical Variables

- H_0: Mean renting prices of different pets / laundry / location / parking options are the same.
- H_A: Means are NOT same.

	P-value
`F_pets, p_pets = stats.f_oneway(*[train_or[train_or.pets==i].price for i in train_or.pets.value_counts().index])`	0.0000
`F_laundry, p_laundry = stats.f_oneway(*[train_or[train_or.laundry==i].price for i in train_or.laundry.value_counts().index])`	0.0000
`F_county, p_county = stats.f_oneway(*[train_or[train_or.county==i].price for i in train_or.county.value_counts().index])`	0.0000
`F_parking, p_parking = stats.f_oneway(*[train_or[train_or.parking==i].price for i in train_or.parking.value_counts().index])`	0.0000

- Result: Reject H_0 for all categorical variable at significance level 0.01

Model Building :: Variable Selection Including Interactions

- **Bias-Variance Tradeoff**
  - Bias $= E(\hat{Y}) - E(Y)$
  - Variance $= \sum var(\hat{Y}_i) = Tr(\sigma^2 H) = \sigma^2 p$, $H = X(X^T X)^{-1} X^T$
  - MSEE(M) = $var(M) + bias^2(M)$
  - E(SSE) = $\sigma^2(n\text{-}p) + ||bias^2(M)||_2^2$
- $AIC = n\log\frac{SSE_p}{n} + 2p$
- $BIC = n\log\frac{SSE_p}{n} + \log(n)\,p$
  - n: # of observations
  - p: # of variables in model
- Adding variable to model → SSE decrease, p increase

  → bias decrease, variance increase

Model Building :: Variable Selection Including Interactions

- **Stepwise Selection** using AIC and BIC criteria
- Step 0: Start from Log(Price) = constant + factor(Laundry) + factor(Parking) + factor(County)
  - AIC = 708.3728 , BIC = 311.8810

…

- AIC Model:

  Log(Price) = constant + sqft + bedrooms + bathrooms +
  bedrooms * bathrooms + sqft * bathrooms +
  factor(Laundry) + factor(Parking) + factor(County)

- **BIC Model:**

  Log(Price) = constant + sqft + bedrooms + bathrooms +
  sqft * bedrooms +
  factor(Laundry) + factor(Parking) + factor(County)

Model Comparison

- Model 1: Main Effect Model
  - Log(Price) = Sqft + Bedrooms + Bathrooms + factor(Pets) + factor(Laundry) + factor(Parking) + factor(County)

R^2	0.770	BIC	-4919
F-statistic	845.4	MSE	0.03566

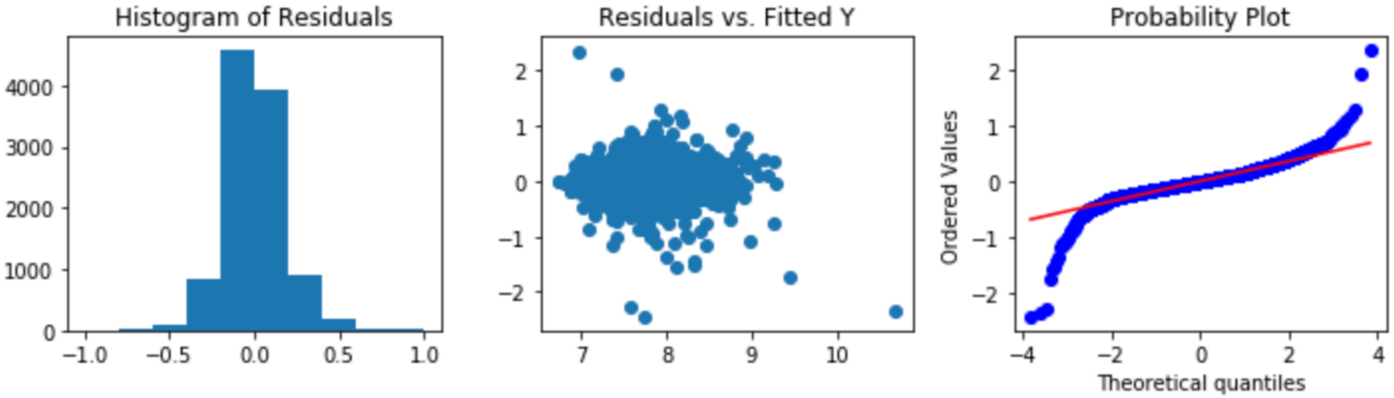
- Model 2:
  - Log(Price) = Sqft + Bedrooms + Bathrooms + Sqft * Bedrooms + factor(Pets) + factor(Laundry) + factor(Parking) + factor(County)

R^2	0.770	BIC	-4923
F-statistic	825.7	MSE	0.03565

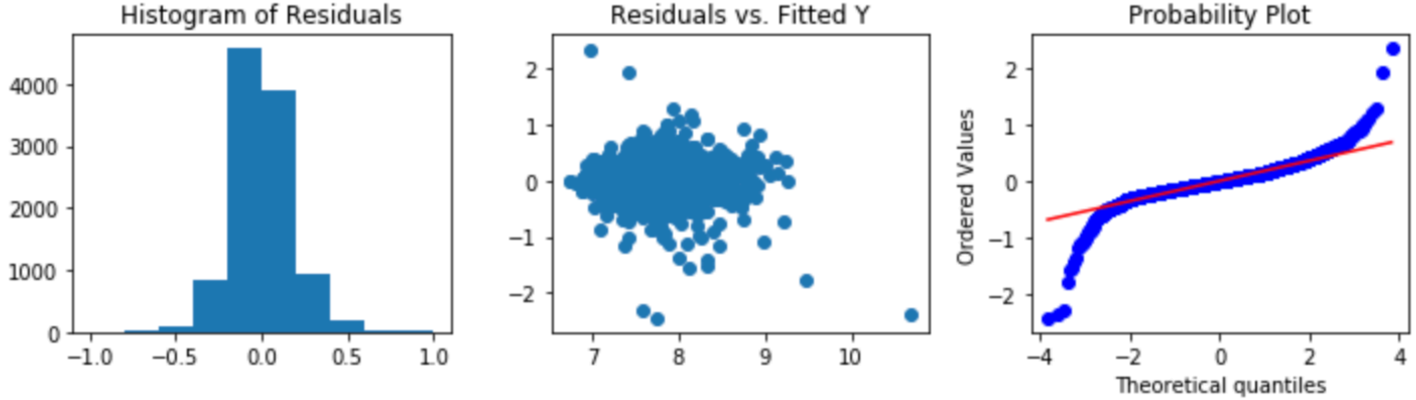
- …

Model Diagnostics

- Model 1:



- Model 2:



Model Building :: Reduce Collinearity

- Least Square: $\min_{\beta}(Y - X\beta)^T(Y - X\beta)$
- Solution: $\hat{\beta}_{OLS} = (X^TX)^{-1}X^TY$
- $(X^TX)^{-1}$ may not exist

- Add regularization term to LS optimization:
  - Ridge: $\min_{\beta}(Y - X\beta)^T(Y - X\beta) + \lambda\,||\beta||_2^2$
  - Lasso: $\min_{\beta}(Y - X\beta)^T(Y - X\beta) + \lambda\,|\beta|$
  - Elastic-Net: : $\min_{\beta}(Y - X\beta)^T(Y - X\beta) + \lambda_1\,||\beta||_2^2 + \lambda_2|\beta|$

Ridge & Lasso

```python
mr = RidgeCV(alphas=[1e-5, 1e-4, 1e-3, 1e-2, 1e-1], fit_intercept=False,
                    scoring = "neg_mean_squared_error", cv = 5).fit(X_train_dummy_S, log_y_train)
```

- Ridge
  - λ = 0.0001
  - R^2 = 0.770

```python
ml = LassoCV(alphas=[1e-8, 1e-7, 1e-6, 1e-5, 1e-4], fit_intercept=False, cv = 5,
                    max_iter=100000).fit(X_train_dummy_S, log_y_train)
```

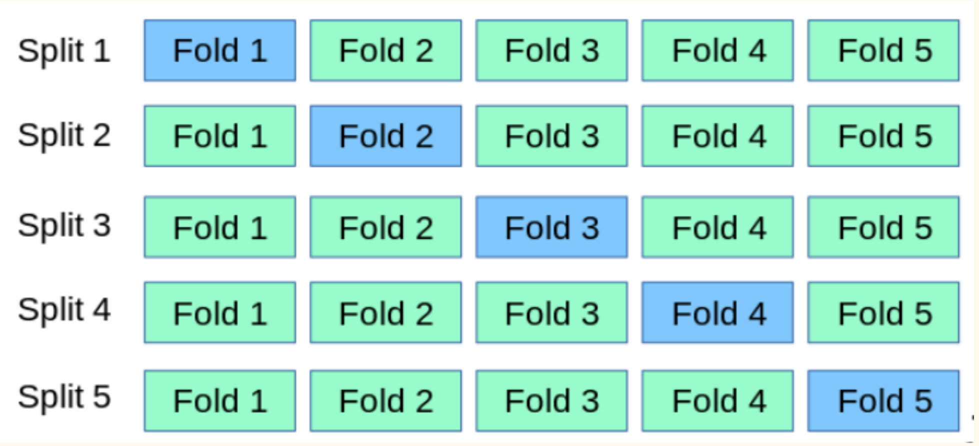
- Lasso
  - λ = 0.005
  - R^2 = 0.721
  - Modify categorical variables (43 → 15 dummy variables)

```python
new_X = new_X.replace(["shared", "hookup", "paid", "none"], "no")
new_X = new_X.replace(["off-street", "none", "street", "paid", "valet"], "no")
new_X = new_X.replace([['Kern', 'Los Angeles', 'Orange', 'Ventura', 'San Bernardino','San Luis Obispo',
                        'Solano','El Dorado', 'San Joaquin', 'Sonoma', 'Sutter','Riverside', 'Contra Costa',
                        'Napa', 'Lake','Mendocino', 'Monterey']], "no")
```

Model Comparison — Cross Validation

- Break the training dataset into 5 folders
- $MSPE = \frac{\sum(Y_i - \hat{Y}_i)^2}{m}$,

  where m is the size of the validation folder
- Cross validation score = $\sum MSPE_k/5$



Model Comparison

- Model 1: Main Effect Model

R^2	0.763		
F-statistic	832.0	MSPE	0.03632

- Model 2: Model with Interaction

R^2	0.763		
F-statistic	812.7	MSPE	1504.374

- Model 3: Lasso with $\lambda = 0.005$
  - Log(Price) = Sqft + Bedrooms + Bathrooms + factor’(Laundry) + factor’(Parking) + factor’(County)

R^2	0.722	MSPE	0.04345

- Model 4: Ridge with $\lambda = 0.0001$

R^2	0.769	MSPE	0.03632

Outlier Detecting :: Using Model 1 (-765)

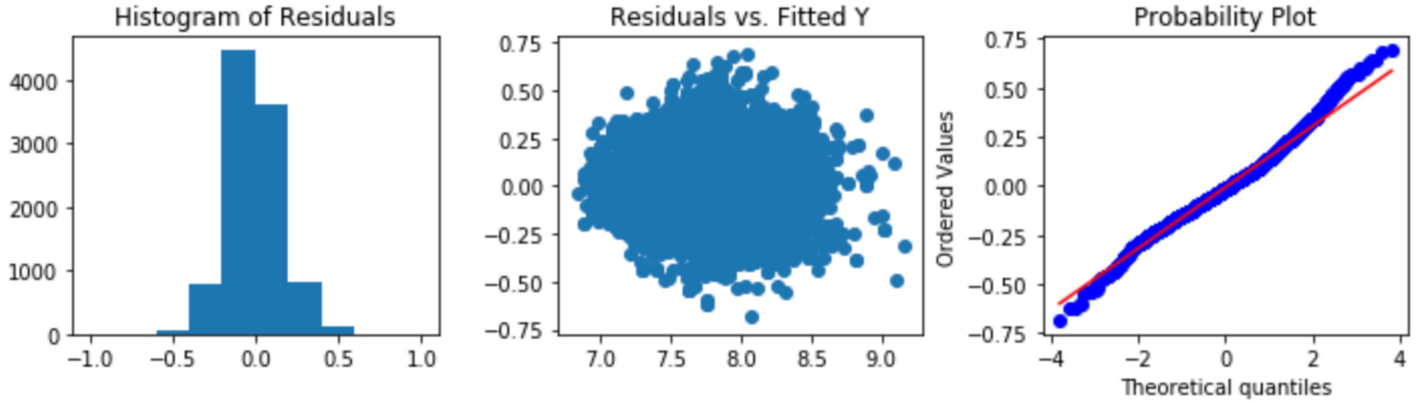
- Studentized Deleted Residuals: (35)
  - $t_i = \frac{d_i}{s\{d_i\}} = \frac{d_i}{\sqrt{MSE_{(i)}/(1-h_{ii})}} = \sqrt{\frac{n-p-1}{SSE(1-h_{ii})-e_i^2}}$,

    where $MSE_{(i)}$ is the MSE of the regression fit excluding case i,

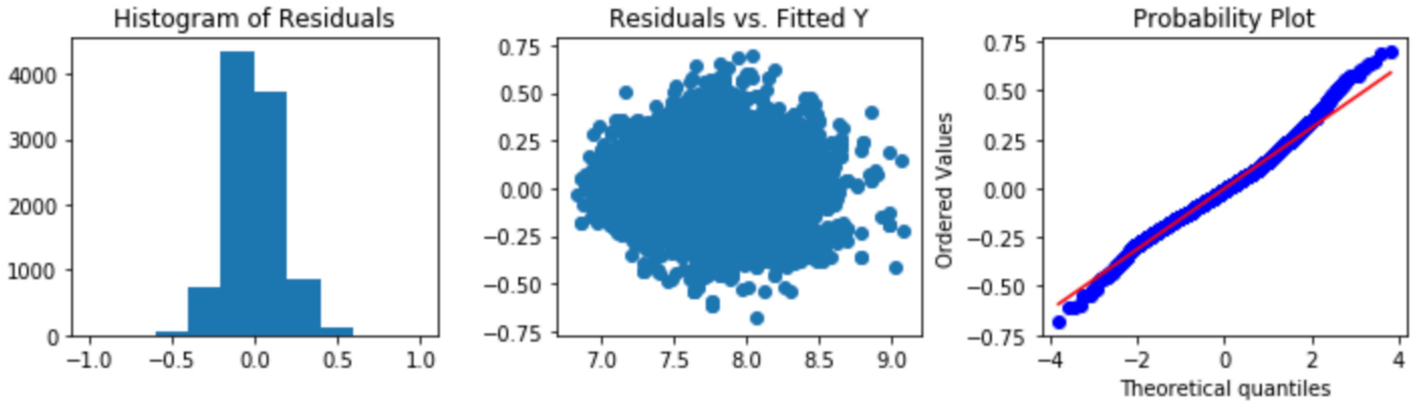
    h_{ii} is the entry (i, i) of the hat matrix $X(X^TX)^{-1}X^T$
  - Under H_0: The model is correct, and $t_i \sim t_{n\text{-}p\text{-}1}$
  - Outlying Y at α: $|t_i| > t_{n-p-1}(1-\alpha/2n)$
- Leverage Value (h_{ii}): (587)
  - $\bar{h} = \frac{1}{n}\sum h_{ii} = \frac{p}{n}$
  - Outlying X: $h_{ii} > \frac{2p}{n}$
- Cook's Distance: (307)
  - $D_i = \frac{\sum(\hat{Y}_j - \hat{Y}_{j(i)})^2}{p * MSE} = \frac{r_i^2 h_{ii}}{p(1-h_{ii})}$, where $r_j = \frac{e_i}{\sqrt{MSE(1-h_{ii})}}$
  - Influential: $D_i > \frac{4}{n-p}$

Model Diagnostics Without Outliers

- Model 1:



- Model 2:



Model Comparison Without Outliers

- Model 1: Main Effect Model

R^2	0.824		
F-statistic	1923	MSPE	0.02510

- Model 2: Model with Interaction

R^2	0.825		
F-statistic	1930	MSPE	0.02757

- **Model 3: Lasso with λ = 0.005**
  - Log(Price) = Sqft + Bedrooms + Bathrooms + factor'(Laundry) + factor'(Parking) + factor'(County)

R^2	0.795	MSPE	0.02862

- Model 4: Ridge with λ = 0.0001

R^2	0.769	MSPE	0.02457

Refitting Model & Prediction

- Model 3:
- $$\log(Price) = 0.1508 * \frac{sqft - 871.2389}{312.9308} + 0.0021 * \frac{bedrooms - 1.5208}{0.7507} + 0.0196 * bathrooms$$
$$+ 0.1203 * \mathrm{C(in\ unit\ laundry)}$$
$$- 0.0424 * \mathrm{C(covered\ parking)} + 0.0462 * \mathrm{C(garage\ parking)}$$
$$- 0.2077 * \mathrm{C(Placer)} - 0.5381 * \mathrm{C(Sacramento)} - 0.2183 * \mathrm{C(San\ Diego)}$$
$$+ 0.3119 * \mathrm{C(San\ Francisco)} + 0.1497 * \mathrm{C(San\ Mateo)}$$
$$+ 0.0908 * \mathrm{C(Santa\ Clara)} - 0.1508 * \mathrm{C(elsewhere)}$$
- Test Set Prediction:

MSPE	With outliers	Without outliers
Model 1	0.03511	> 10000
Model 2	0.03509	0.06399
Model 3	**0.04326**	**0.04370**
Model 4	0.03511	0.06389

Conclusion

- Lasso Regression has simpler model, and acceptable predicting ability.
- Collinearity will not hurt predicting ability.
- Sqft, #Bedrooms, #Bathrooms are all positive related to renting price.
- Levels of categorical data can be simplified to reduce dimensions.
- Large cities tend to have more expensive renting price.
- Standardization may hurt the predictivity, but can reduce effects caused by scale differences.
- Some coefficients do not make sense, but predicting results seem fine.

Thank you for listening!