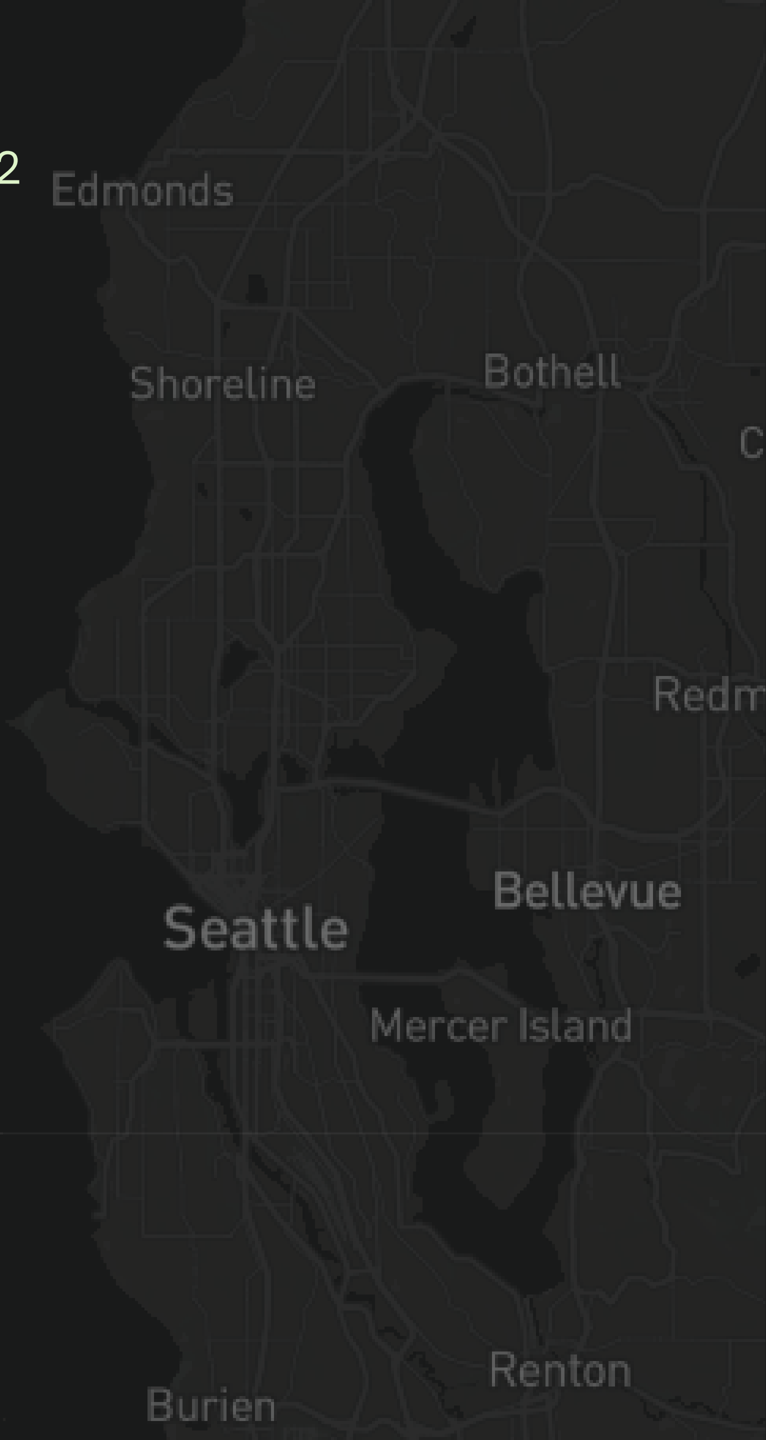
APPLICATION DEVELOPMENT COURSE PROJECT - IBA KARACHI - DEC. '22

Airbnb in Seattle

Rental Rates Analysis 2015-17



Bilal Naseem - 13216
M. Salman Malik - 27256

Overview

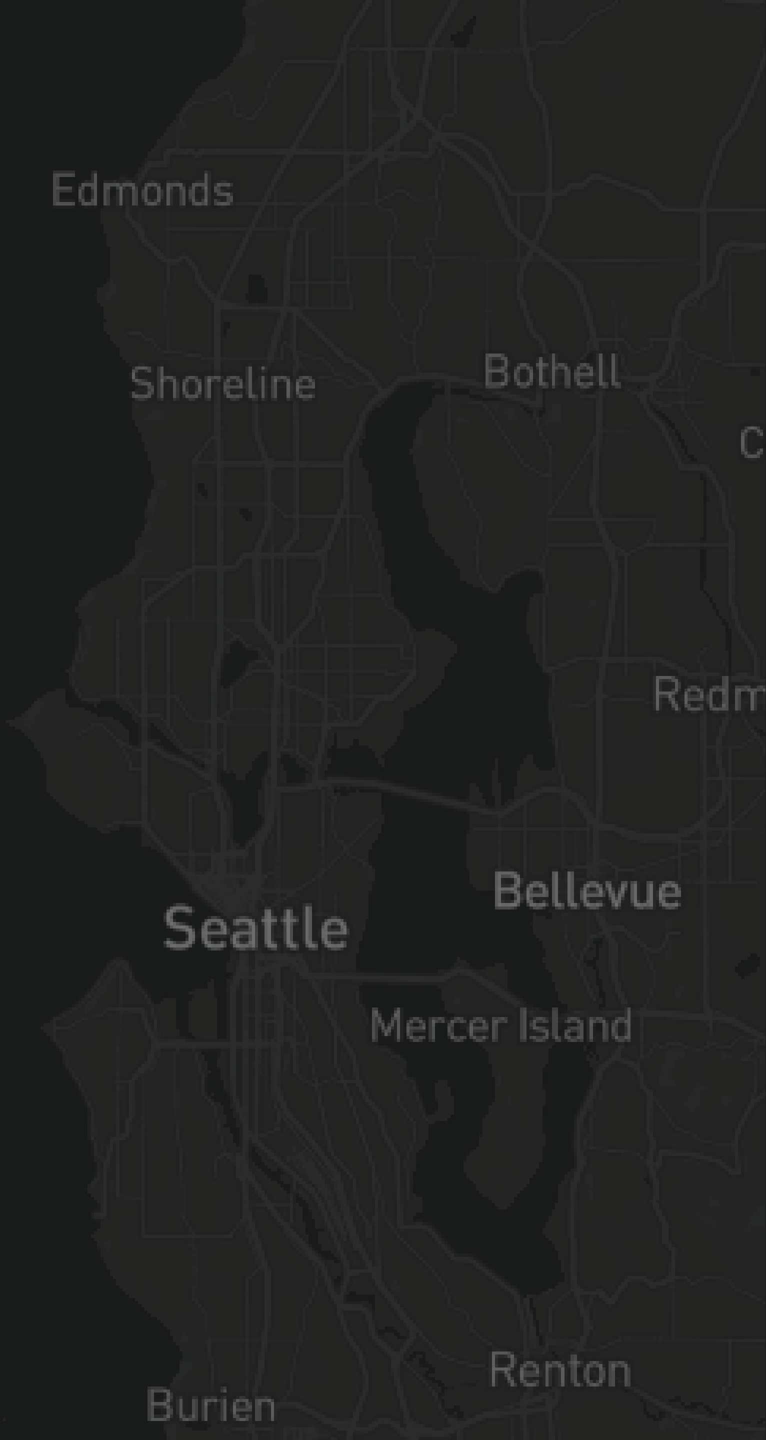
01 About the Data

02 Cleaning

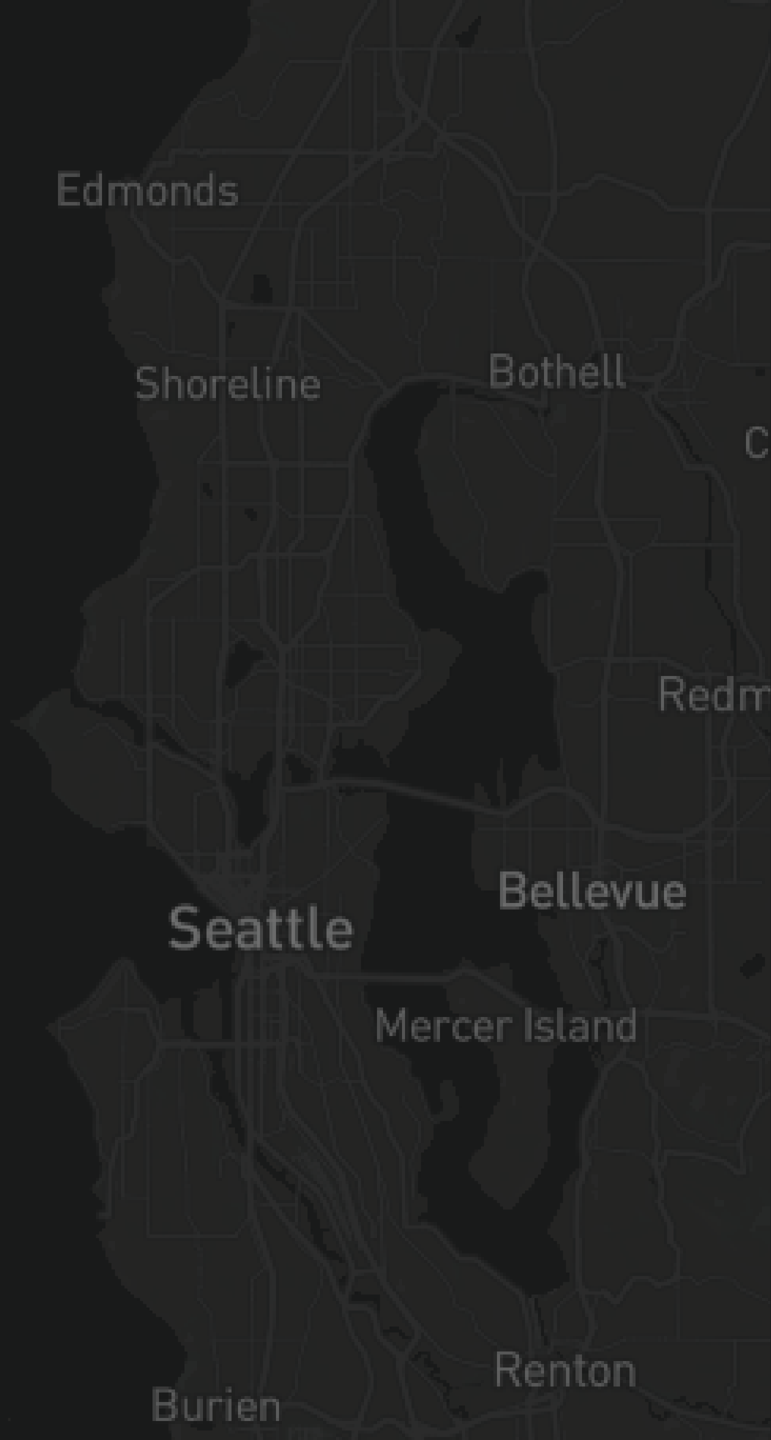
03 Exploratory Data Analysis (EDA)

04 Pre-Processing

05 Dashboard



Data Cleaning

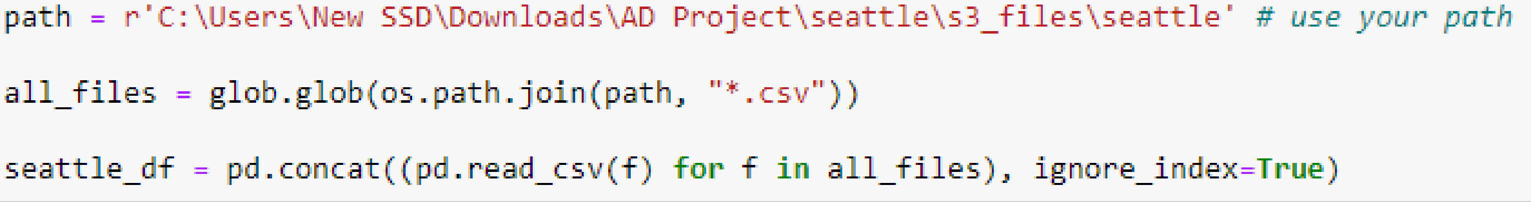


The DataSet

DataSet link:

http://tomslee.net/airbnb-data-collection-get-the-data

- The DataSet represents rates of properties put up on Airbnb in Seattle from Sept. 2015 to July 2017.
- 113,676 Rows and 19 columns.



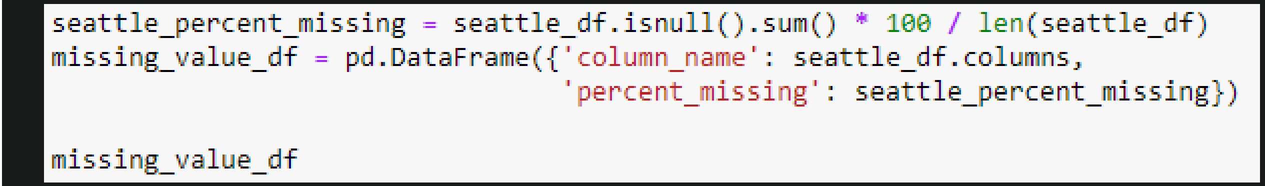
```
path = r'C:\Users\New SSD\Downloads\AD Project\seattle\s3_files\seattle' # use your path

all_files = glob.glob(os.path.join(path, "*.csv"))

seattle_df = pd.concat((pd.read_csv(f) for f in all_files), ignore_index=True)
```

room_id	host_id	room_type	neighborhood	reviews	overall_satisfaction	accommodates	bedrooms	price	minstay	latitude	longitude	last_modified	country	cat_price	cat_reviews
4597013	23827679	Private room	Alki	0	0.9	2	1	225	1	47.561296	-122.400262	11:20.0	USA	Very High	very less reviews
7048843	36964583	Private room	Atlantic	10	4.5	4	1	60	1	47.590832	-122.299813	12:21.0	USA	low	low reviews
3998922	20732089	Private room	Atlantic	13	5	2	1	68	2	47.596227	-122.302923	12:21.0	USA	low	sufficient reviews
6411986	7431966	Private room	Atlantic	7	4	2	1	90	2	47.591966	-122.308393	12:21.0	USA	Normal	low reviews
7619060	12194562	Private room	Atlantic	1	5	2	1	79	1	47.595159	-122.309061	12:21.0	USA	Normal	very less reviews
7095802	36964583	Private room	Atlantic	11	4.5	2	1	50	1	47.592136	-122.30008	12:21.0	USA	low	sufficient reviews
879181	287172	Private room	Atlantic	26	4.5	2	1	60	2	47.60051	-122.301994	12:21.0	USA	low	sufficient reviews
877203	287172	Private room	Atlantic	24	4.5	2	1	60	2	47.600266	-122.299867	12:21.0	USA	low	sufficient reviews
1898774	1274285	Private room	Atlantic	2	5	2	1	75	2	47.597899	-122.300974	12:21.1	USA	Normal	very less reviews

Initial Cleaning

- The DataSet represents rates of properties put up on Airbnb in seattle from Sept. 2015 to July 2017
- 113,676 Rows and 19 columns

column_name	percent_missing
room_id	0.00%
host_id	0.01%
room_type	0.01%
borough	100.00%
neighborhood	0.00%
reviews	0.00%
overall_satisfaction	13.51%
accommodates	3.62%
bedrooms	5.03%
price	0.00%
minstay	46.53%
latitude	0.00%
longitude	0.00%
last_modified	0.00%
survey_id	78.35%
country	100.00%
city	78.35%
bathrooms	100.00%
location	78.35%



```python
seattle_percent_missing = seattle_df.isnull().sum() * 100 / len(seattle_df)
missing_value_df = pd.DataFrame({'column_name': seattle_df.columns,
                                 'percent_missing': seattle_percent_missing})

missing_value_df
```

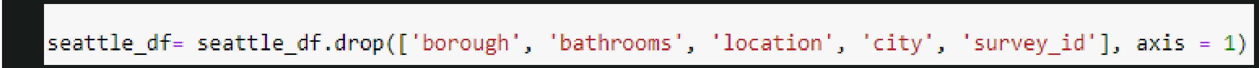
Initial Cleaning

- The DataSet represents rates of properties put up on Airbnb in seattle from Sept. 2015 to July 2017
- 113,676 Rows and 19 columns

column_name	percent_missing
room_id	0.00%
host_id	0.01%
room_type	0.01%
borough	100.00%
neighborhood	0.00%
reviews	0.00%
overall_satisfaction	13.51%
accommodates	3.62%
bedrooms	5.03%
price	0.00%
minstay	46.53%
latitude	0.00%
longitude	0.00%
last_modified	0.00%
survey_id	78.35%
country	100.00%
city	78.35%
bathrooms	100.00%
location	78.35%

Dropped columns: (>75% nulls)

- Borough - 100% nulls
- Bathrooms - 100% nulls
- City - 78.35% nulls
- Location - 78.35% nulls



```
seattle_df= seattle_df.drop(['borough', 'bathrooms', 'location', 'city', 'survey_id'], axis = 1)
```

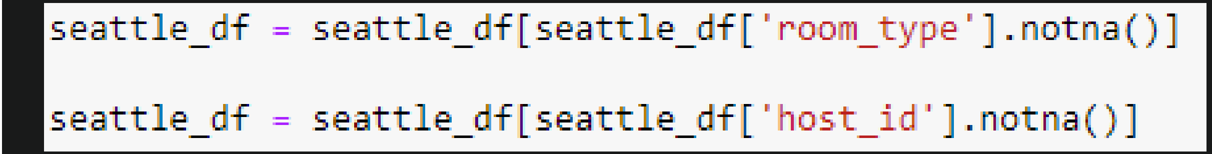
Initial Cleaning

- The DataSet represents rates of properties put up on Airbnb in seattle from Sept. 2015 to July 2017
- 113,676 Rows and 19 columns

column_name	percent_missing
room_id	0.00%
host_id	0.01%
room_type	0.01%
borough	100.00%
neighborhood	0.00%
reviews	0.00%
overall_satisfaction	13.51%
accommodates	3.62%
bedrooms	5.03%
price	0.00%
minstay	46.53%
latitude	0.00%
longitude	0.00%
last_modified	0.00%
survey_id	78.35%
country	100.00%
city	78.35%
bathrooms	100.00%
location	78.35%

Removed Rows: (<1% nulls)

- Host id
- Room type



```
seattle_df = seattle_df[seattle_df['room_type'].notna()]

seattle_df = seattle_df[seattle_df['host_id'].notna()]
```

Initial Cleaning

- The DataSet represents rates of properties put up on Airbnb in seattle from Sept. 2015 to July 2017
- 113,676 Rows and 19 columns

column_name	percent_missing
room_id	0.00%
host_id	0.01%
room_type	0.01%
borough	100.00%
neighborhood	0.00%
reviews	0.00%
overall_satisfaction	13.51%
accommodates	3.62%
bedrooms	5.03%
price	0.00%
minstay	46.53%
latitude	0.00%
longitude	0.00%
last_modified	0.00%
survey_id	78.35%
country	100.00%
city	78.35%
bathrooms	100.00%
location	78.35%

Imputed columns: (<50% nulls)

- Country
- Minstay
- Overall Satisfaction
- Accommodates
- Bedrooms

Initial Cleaning

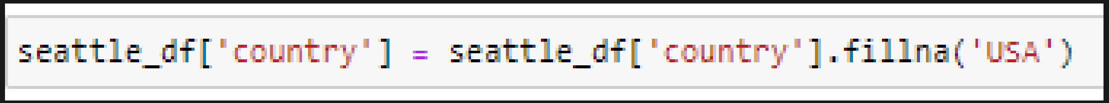
- The DataSet represents rates of properties put up on Airbnb in seattle from Sept. 2015 to July 2017
- 113,676 Rows and 19 columns

column_name	percent_missing
room_id	0.00%
host_id	0.01%
room_type	0.01%
borough	100.00%
neighborhood	0.00%
reviews	0.00%
overall_satisfaction	13.51%
accommodates	3.62%
bedrooms	5.03%
price	0.00%
minstay	46.53%
latitude	0.00%
longitude	0.00%
last_modified	0.00%
survey_id	78.35%
country	100.00%
city	78.35%
bathrooms	100.00%
location	78.35%

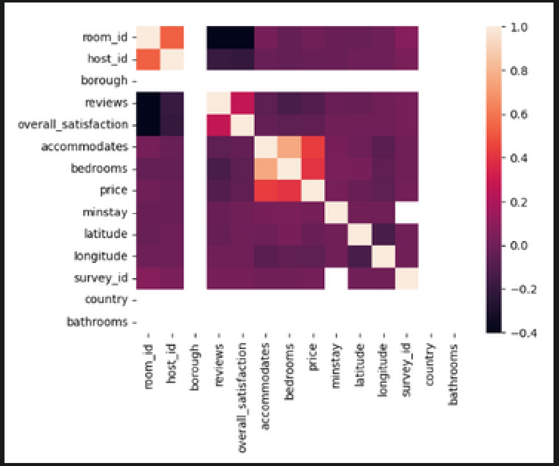
Imputed columns: (<50% nulls)

- Country
- Minstay
- Overall Satisfaction
- Accommodates
- Bedrooms

The entire Country Column was imputed with 'USA'.



```
seattle_df['country'] = seattle_df['country'].fillna('USA')
```



Initial Cleaning

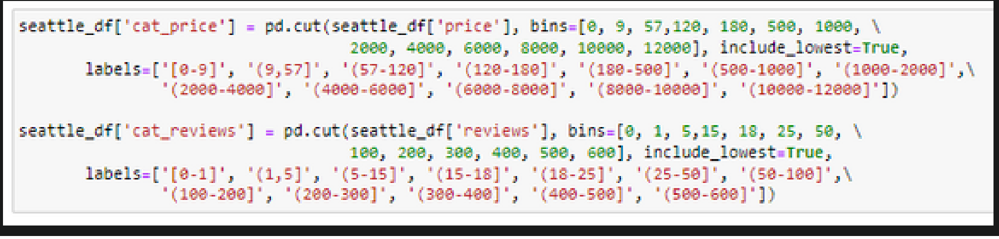
- The DataSet represents rates of properties put up on Airbnb in seattle from Sept. 2015 to July 2017
- 113,676 Rows and 19 columns

column_name	percent_missing
room_id	0.00%
host_id	0.01%
room_type	0.01%
borough	100.00%
neighborhood	0.00%
reviews	0.00%
overall_satisfaction	13.51%
accommodates	3.62%
bedrooms	5.03%
price	0.00%
minstay	46.53%
latitude	0.00%
longitude	0.00%
last_modified	0.00%
survey_id	78.35%
country	100.00%
city	78.35%
bathrooms	100.00%
location	78.35%

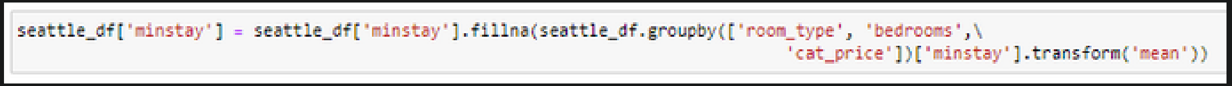
Imputed columns: (<50% nulls)

- Country
- Minstay
- Overall Satisfaction
- Accommodates
- Bedrooms

- It was assumed that Minimum Stay depends on room type, bedrooms, and price.
- Price and Reviews and continuous so they were categorized.
- The mean value of the features were imputed for nulls.



```
seattle_df['cat_price'] = pd.cut(seattle_df['price'], bins=[0, 9, 57,120, 180, 500, 1000, \
                                    2000, 4000, 6000, 8000, 10000, 12000], include_lowest=True,
        labels=['[0-9]', '(9,57]', '(57-120]', '(120-180]', '(180-500]', '(500-1000]', '(1000-2000]',\
              '(2000-4000]', '(4000-6000]', '(6000-8000]', '(8000-10000]', '(10000-12000]'])

seattle_df['cat_reviews'] = pd.cut(seattle_df['reviews'], bins=[0, 1, 5,15, 18, 25, 50, \
                                    100, 200, 300, 400, 500, 600], include_lowest=True,
        labels=['[0-1]', '(1,5]', '(5-15]', '(15-18]', '(18-25]', '(25-50]', '(50-100]',\
              '(100-200]', '(200-300]', '(300-400]', '(400-500]', '(500-600]'])
```



```
seattle_df['minstay'] = seattle_df['minstay'].fillna(seattle_df.groupby(['room_type', 'bedrooms',\
                                                                         'cat_price'])['minstay'].transform('mean'))
```

Initial Cleaning

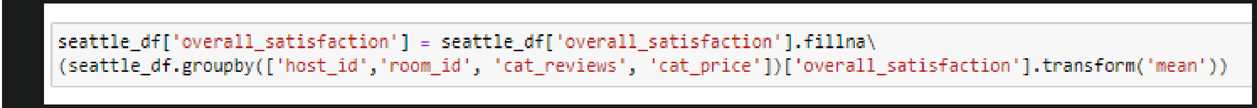
- The DataSet represents rates of properties put up on Airbnb in seattle from Sept. 2015 to July 2017
- 113,676 Rows and 19 columns

column_name	percent_missing
room_id	0.00%
host_id	0.01%
room_type	0.01%
borough	100.00%
neighborhood	0.00%
reviews	0.00%
overall_satisfaction	13.51%
accommodates	3.62%
bedrooms	5.03%
price	0.00%
minstay	46.53%
latitude	0.00%
longitude	0.00%
last_modified	0.00%
survey_id	78.35%
country	100.00%
city	78.35%
bathrooms	100.00%
location	78.35%

Imputed columns: (<50% nulls)

- Country
- Minstay
- Overall Satisfaction
- Accommodates
- Bedrooms

- It was assumed that Overall Satisfaction depends on host, room id, number of reviews and price.
- The mean value of these features were imputed.



```
seattle_df['overall_satisfaction'] = seattle_df['overall_satisfaction'].fillna\
(seattle_df.groupby(['host_id','room_id', 'cat_reviews', 'cat_price'])['overall_satisfaction'].transform('mean'))
```

Initial Cleaning

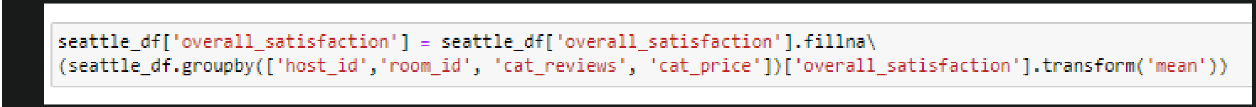
- The DataSet represents rates of properties put up on Airbnb in seattle from Sept. 2015 to July 2017
- 113,676 Rows and 19 columns

column_name	percent_missing
room_id	0.00%
host_id	0.01%
room_type	0.01%
borough	100.00%
neighborhood	0.00%
reviews	0.00%
overall_satisfaction	13.51%
accommodates	3.62%
bedrooms	5.03%
price	0.00%
minstay	46.53%
latitude	0.00%
longitude	0.00%
last_modified	0.00%
survey_id	78.35%
country	100.00%
city	78.35%
bathrooms	100.00%
location	78.35%

Imputed columns: (<50% nulls)

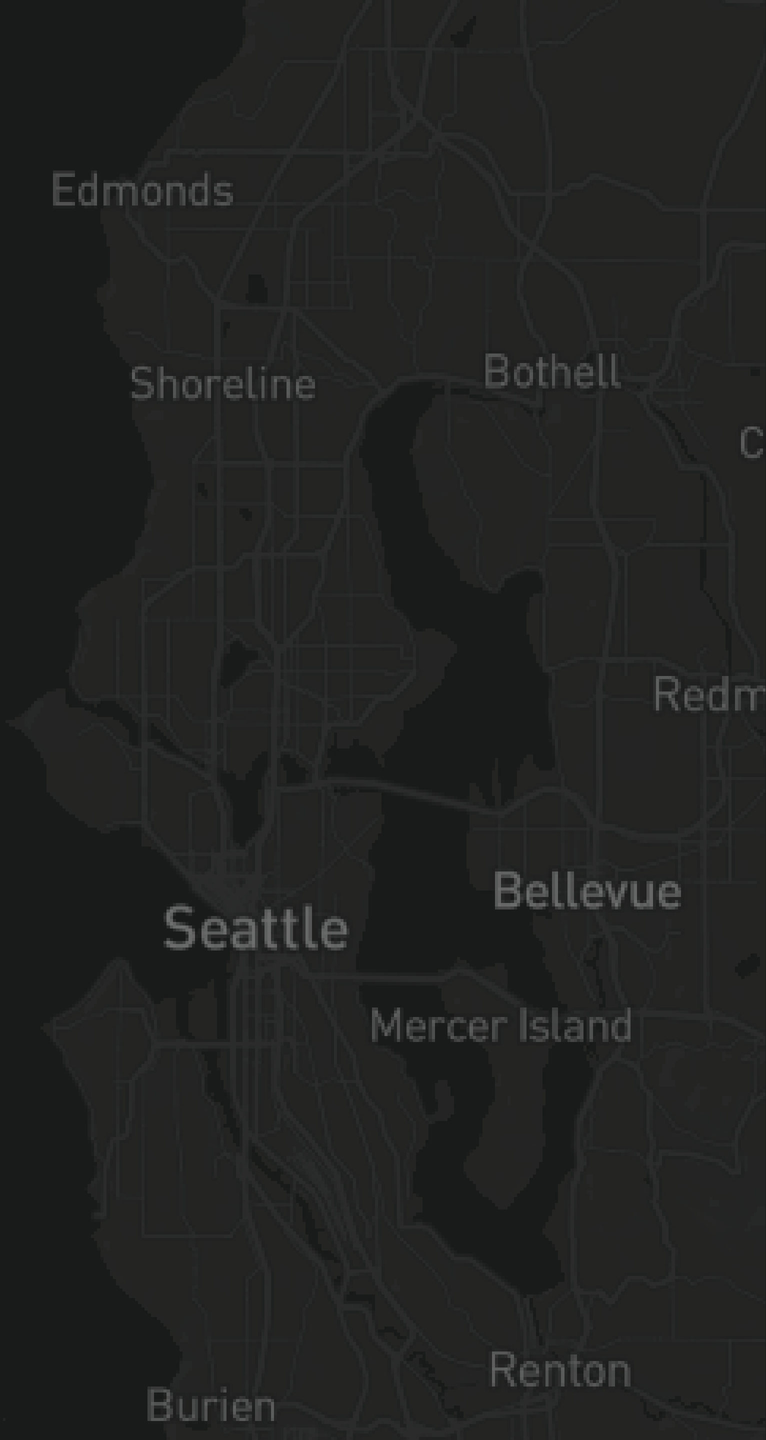
- Country
- Minstay
- Overall Satisfaction
- Accommodates
- Bedrooms

- It was assumed that Accomodates depend on room id, room type, neighborhood, and bedrooms.
- The mean value of these features were imputed.



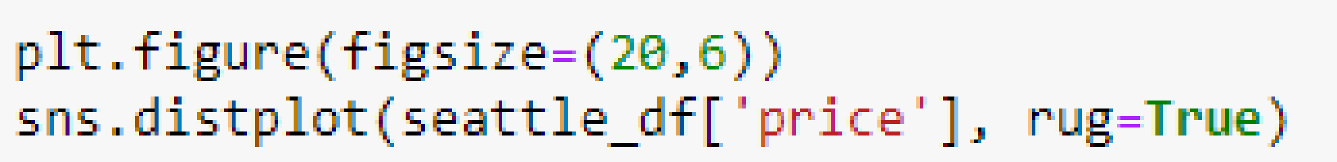
```
seattle_df['overall_satisfaction'] = seattle_df['overall_satisfaction'].fillna\
(seattle_df.groupby(['host_id','room_id', 'cat_reviews', 'cat_price'])['overall_satisfaction'].transform('mean'))
```

Exploratory Data Analysis (EDA)

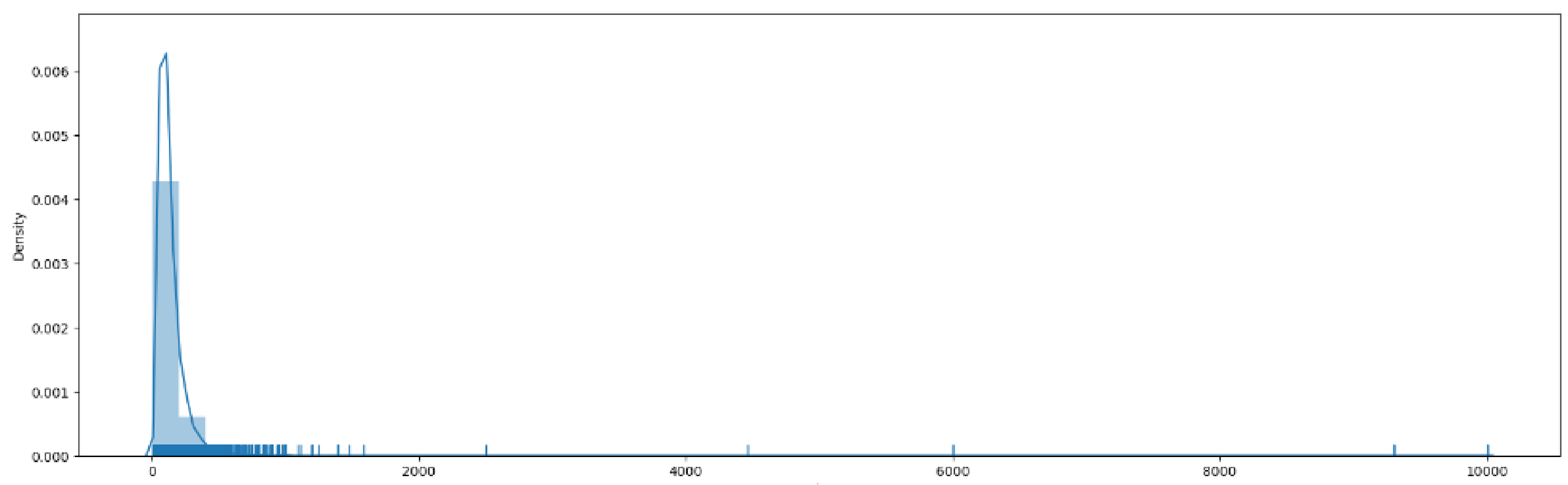


Exploratory Data Analysis (EDA)

KDE Plot of the price column

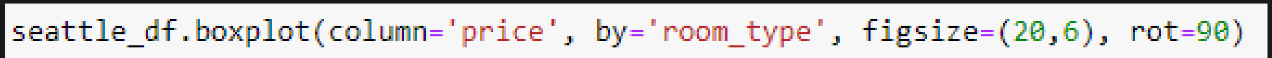


```
plt.figure(figsize=(20,6))
sns.distplot(seattle_df['price'], rug=True)
```

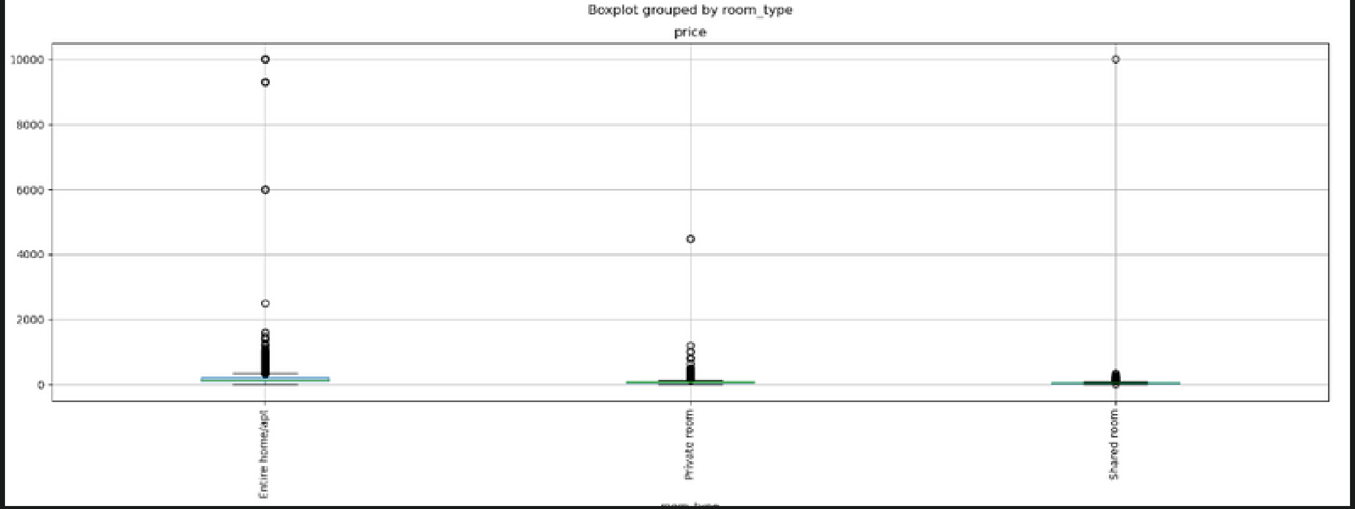


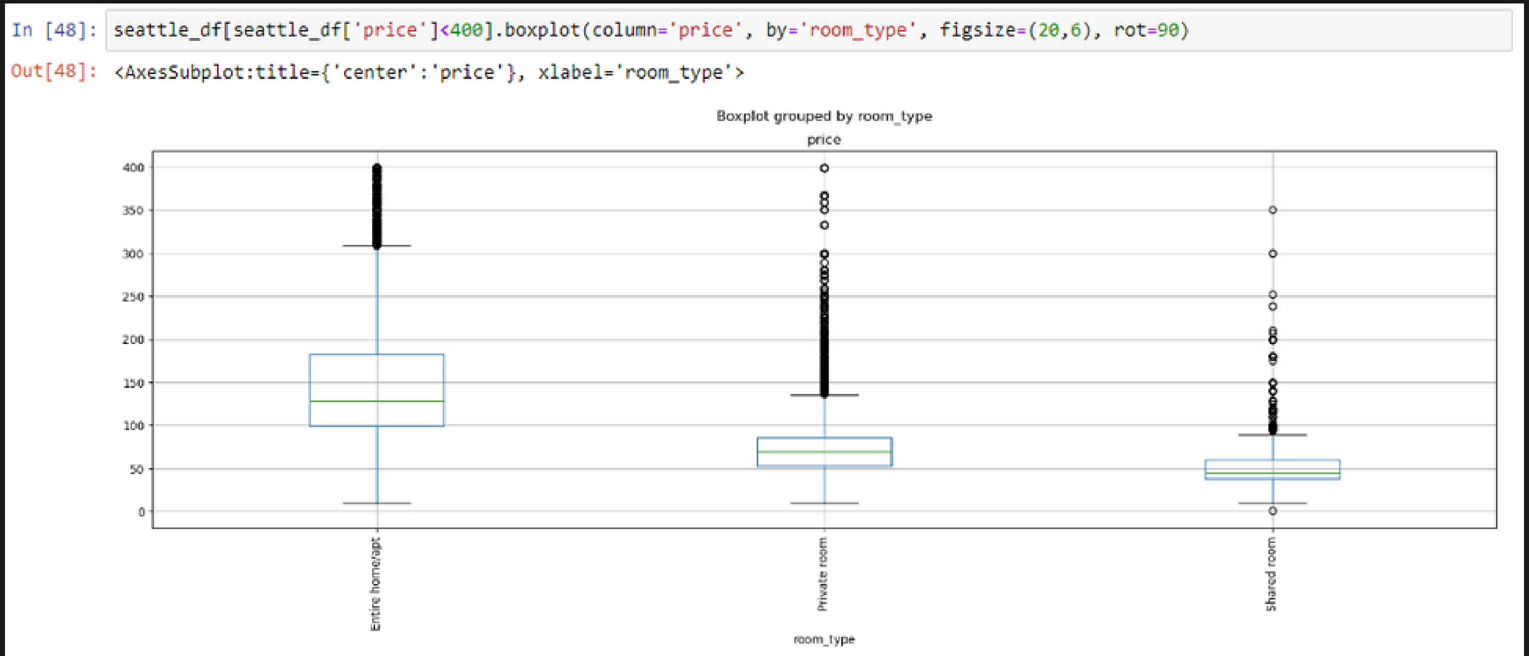
Exploratory Data Analysis (EDA)

Room Type vs Price



```
seattle_df.boxplot(column='price', by='room_type', figsize=(20,6), rot=90)
```

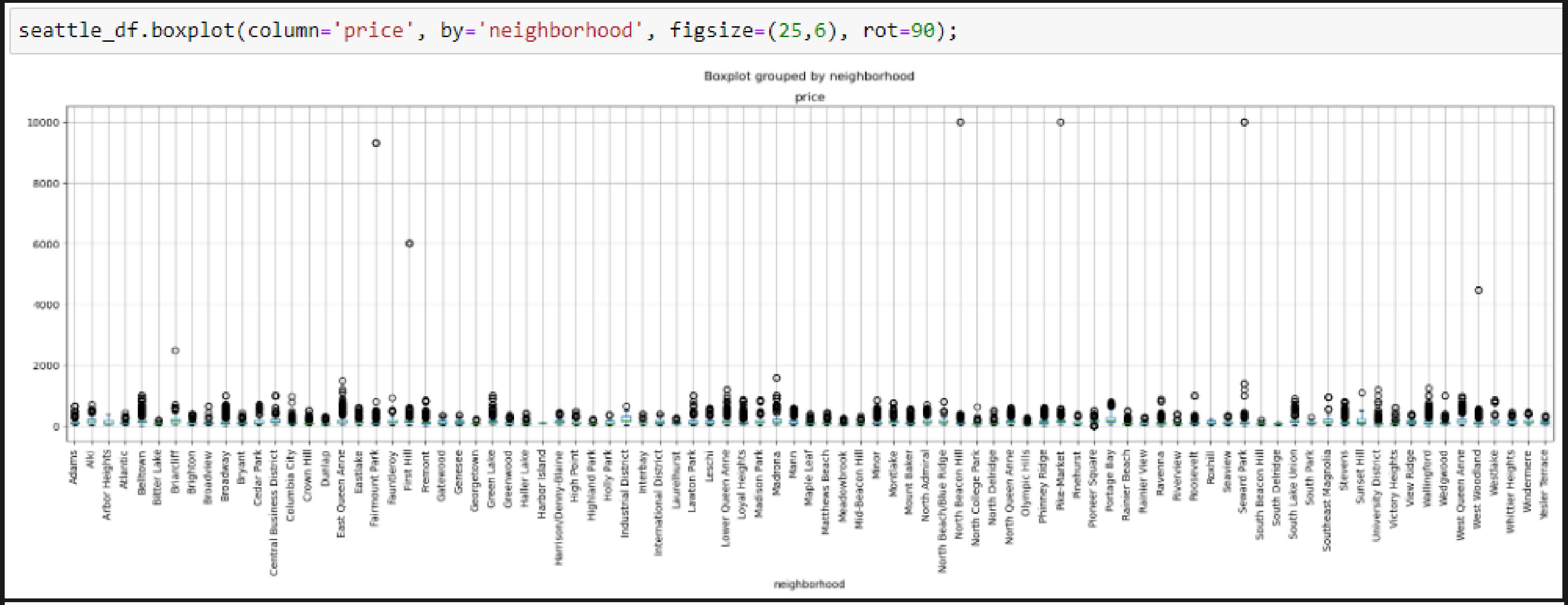




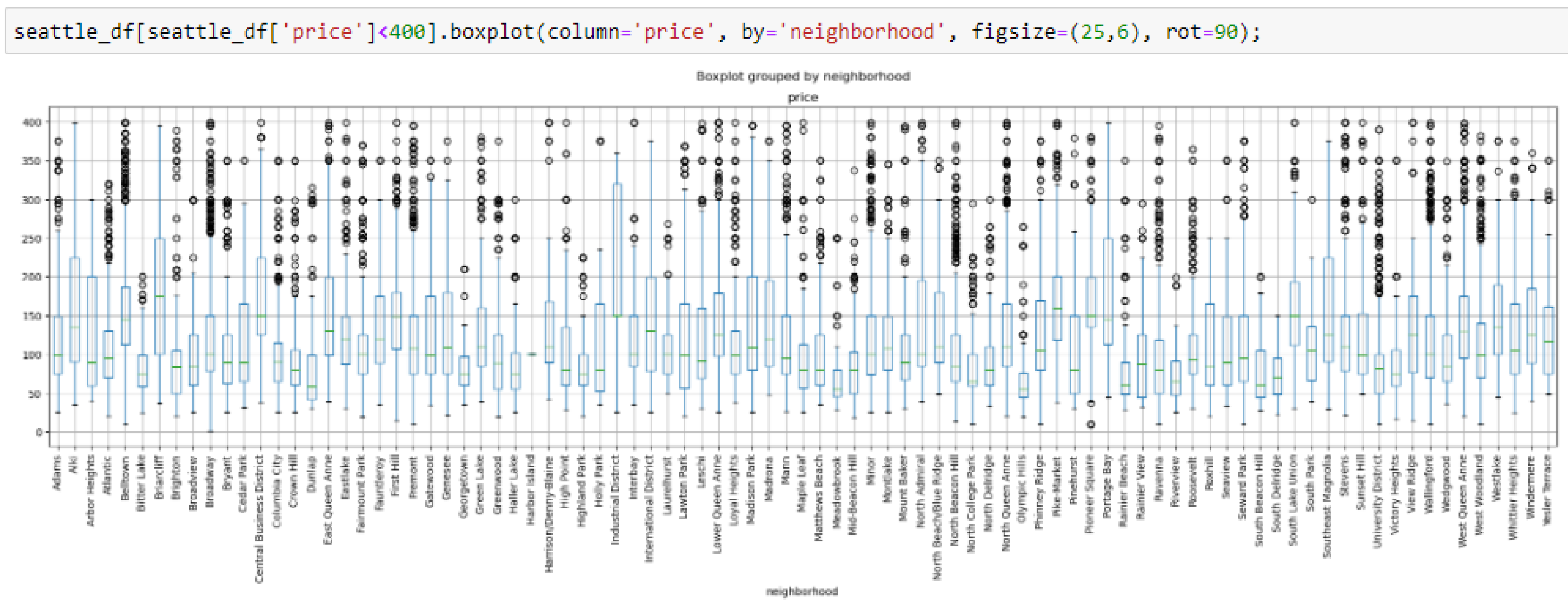
```
In [48]: seattle_df[seattle_df['price']<400].boxplot(column='price', by='room_type', figsize=(20,6), rot=90)
Out[48]: <AxesSubplot:title={'center':'price'}, xlabel='room_type'>
```

Exploratory Data Analysis (EDA)

Neighborhood vs Price



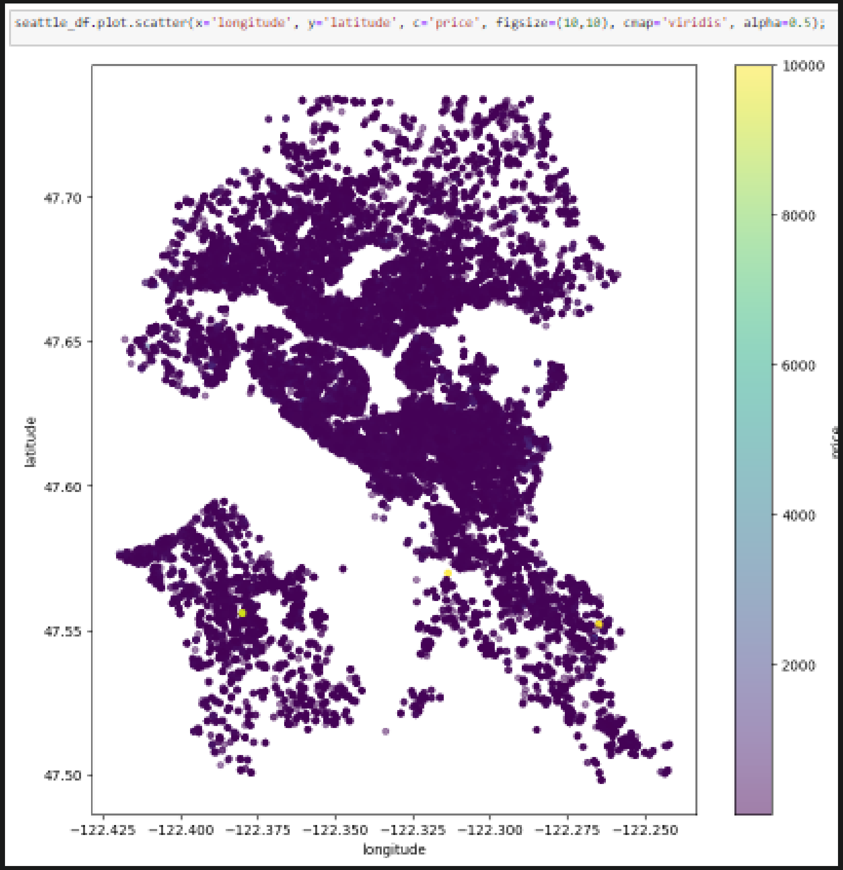
```
seattle_df.boxplot(column='price', by='neighborhood', figsize=(25,6), rot=90);
```



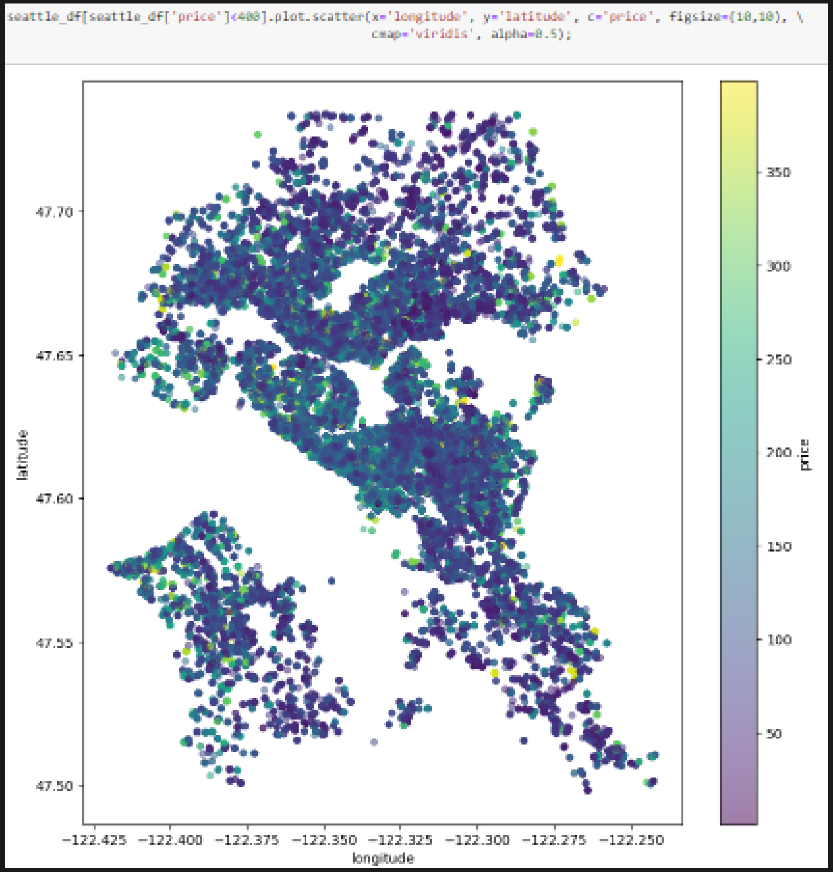
```
seattle_df[seattle_df['price']<400].boxplot(column='price', by='neighborhood', figsize=(25,6), rot=90);
```

Exploratory Data Analysis (EDA)

Latitude & Longitude vs Price



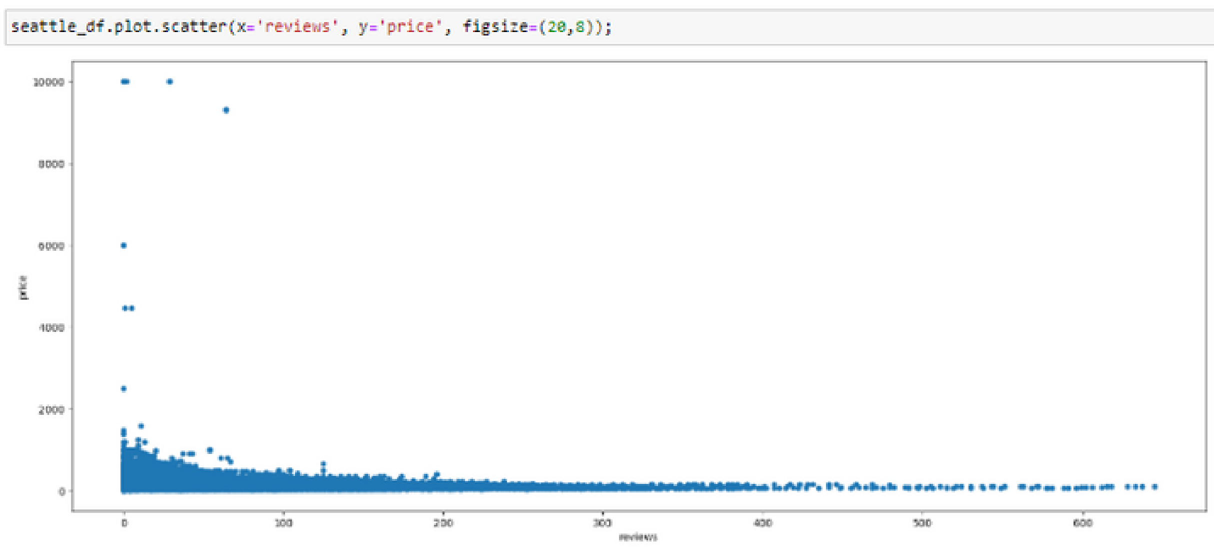
```
seattle_df.plot.scatter(x='longitude', y='latitude', c='price', figsize=(10,10), cmap='viridis', alpha=0.5);
```



```
seattle_df[seattle_df['price']<400].plot.scatter(x='longitude', y='latitude', c='price', figsize=(10,10), \
                                              cmap='viridis', alpha=0.5);
```

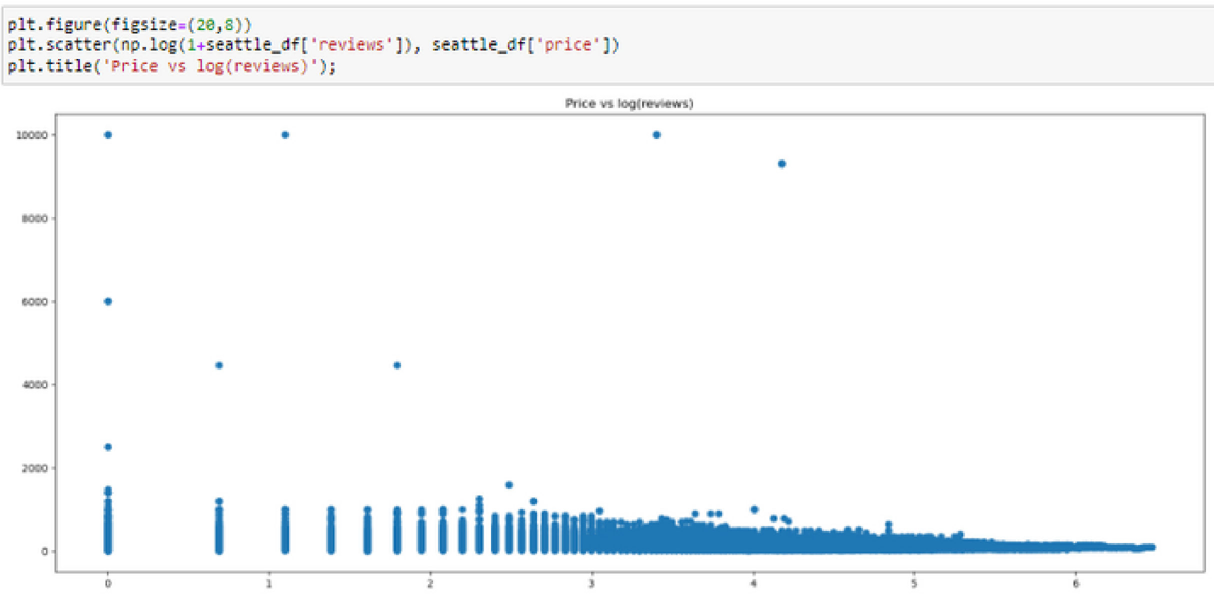
Exploratory Data Analysis (EDA)

Number of Reviews



```
seattle_df.plot.scatter(x='reviews', y='price', figsize=(20,8));
```

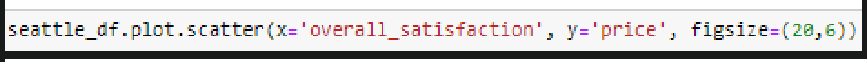
Price vs log(Reviews)



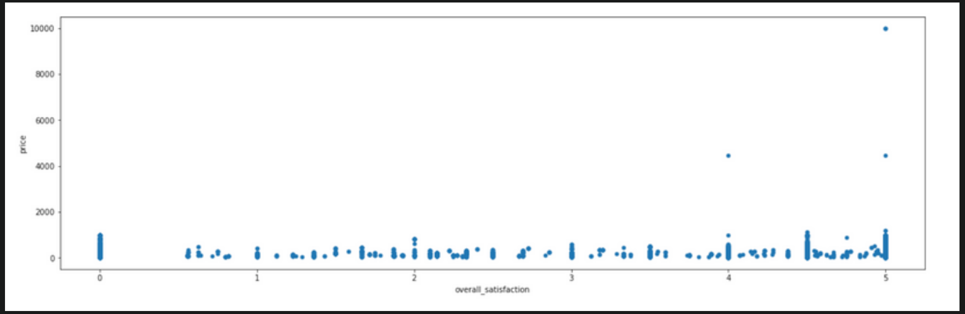
```
plt.figure(figsize=(20,8))
plt.scatter(np.log(1+seattle_df['reviews']), seattle_df['price'])
plt.title('Price vs log(reviews)');
```

Exploratory Data Analysis (EDA)

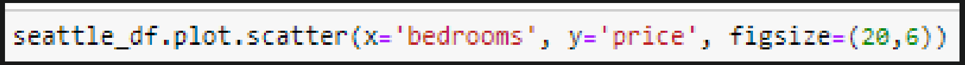
Price vs Overall Satisfaction



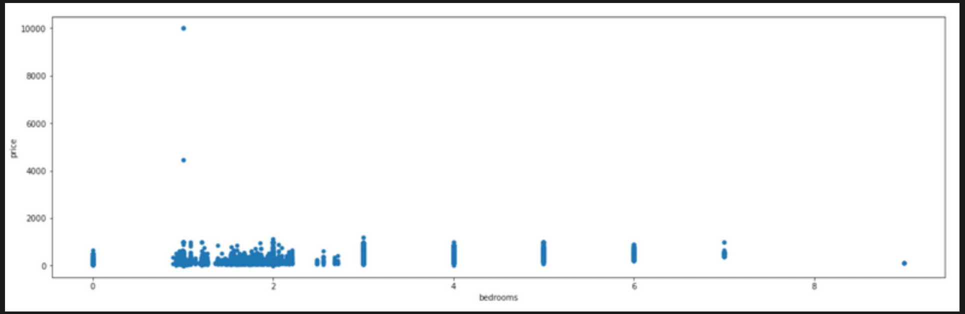
```
seattle_df.plot.scatter(x='overall_satisfaction', y='price', figsize=(20,6))
```



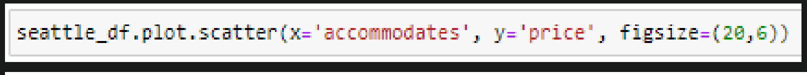
Price vs Bedrooms



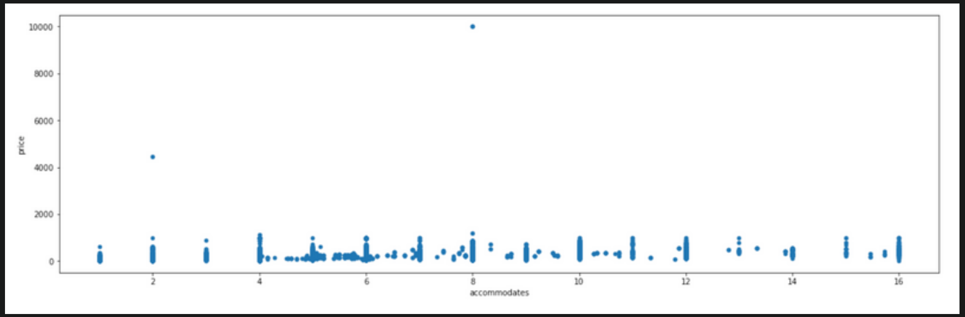
```
seattle_df.plot.scatter(x='bedrooms', y='price', figsize=(20,6))
```



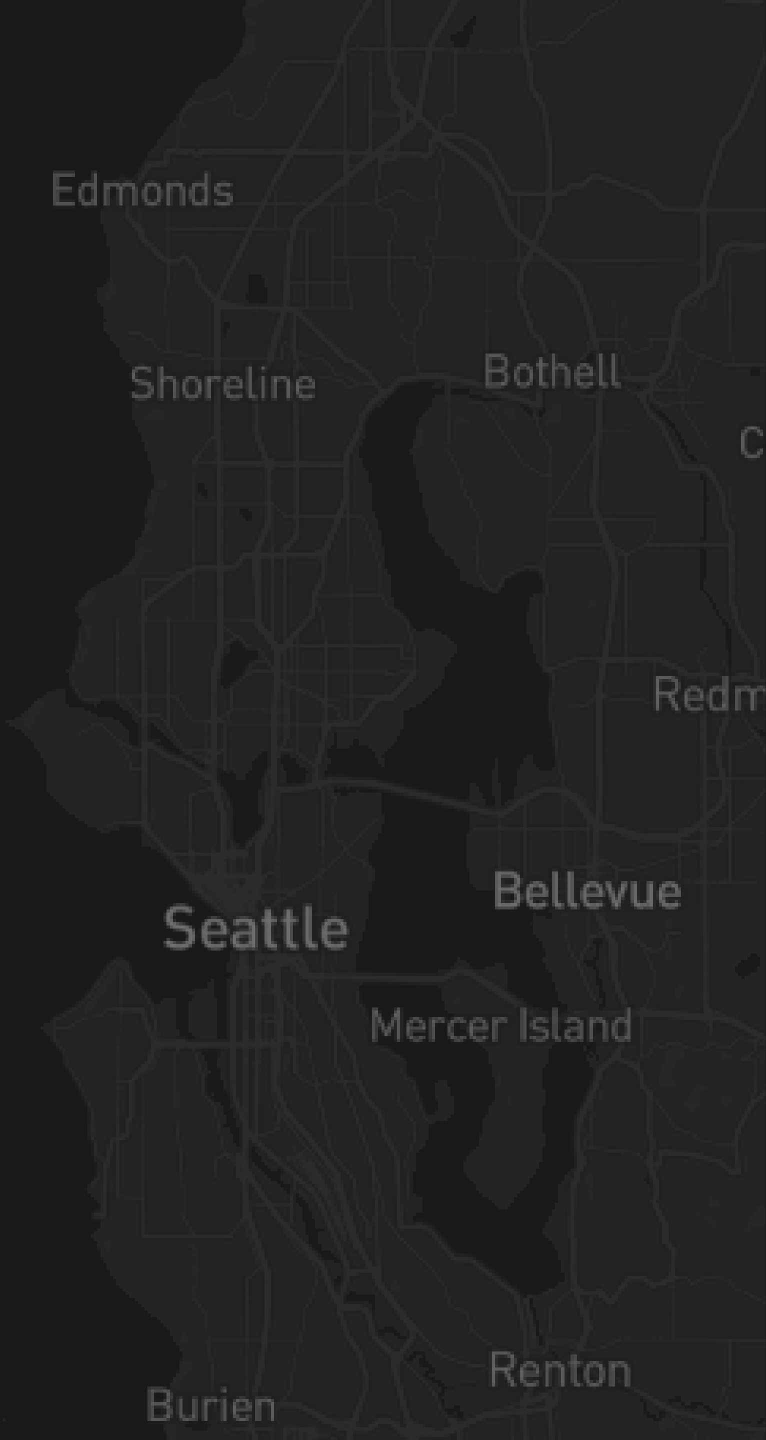
Price vs Accomodates



```
seattle_df.plot.scatter(x='accommodates', y='price', figsize=(20,6))
```

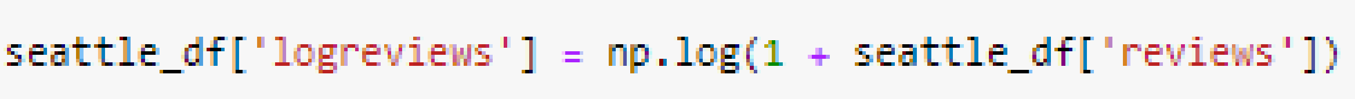


Pre-Processing & Feature Engineering



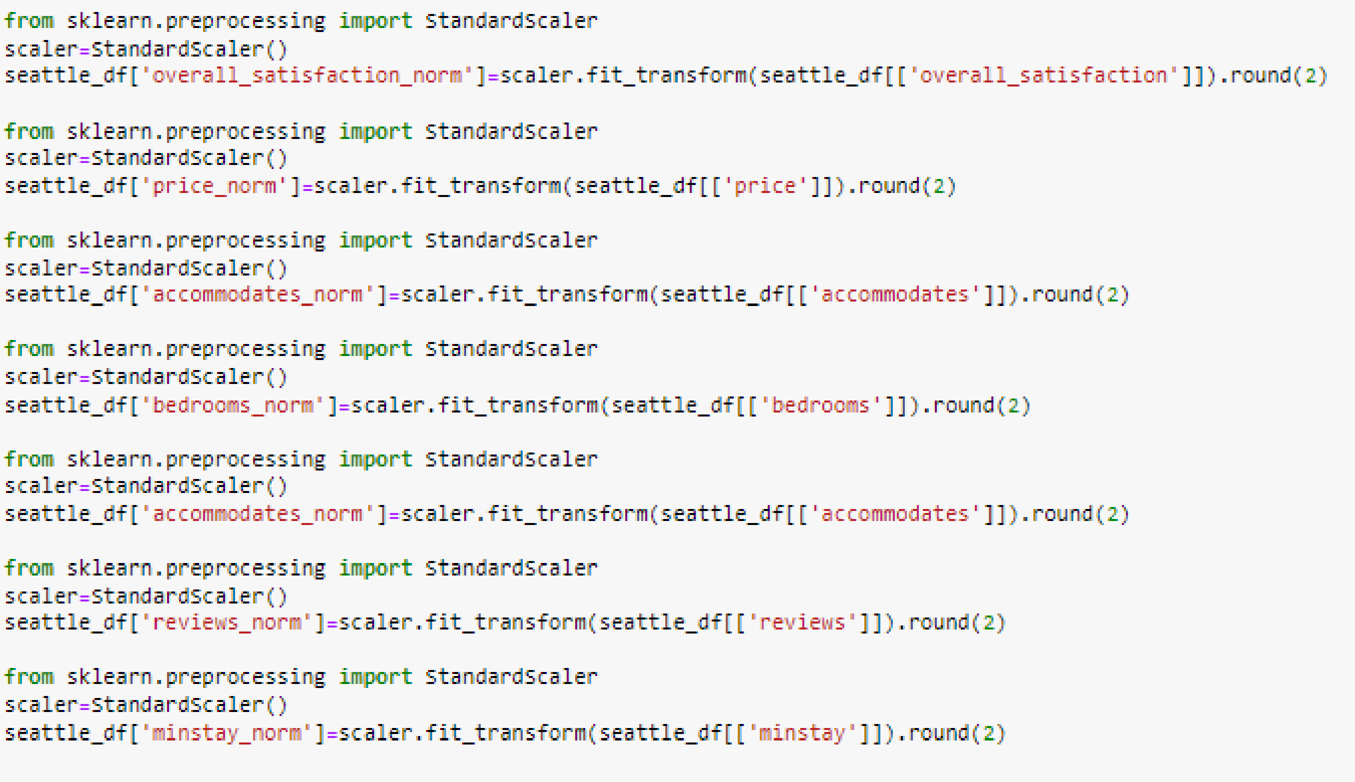
Pre-Processing

- A new column named 'log_reviews' was made to make the magnitude of the reviews more closer to each other
- Also, for better visulaisation by negating any extreme values or outliers.



```
seattle_df['logreviews'] = np.log(1 + seattle_df['reviews'])
```

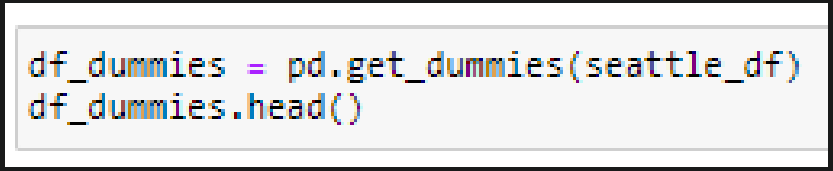
Normalization



```
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
seattle_df['overall_satisfaction_norm']=scaler.fit_transform(seattle_df[['overall_satisfaction']]).round(2)

from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
seattle_df['price_norm']=scaler.fit_transform(seattle_df[['price']]).round(2)

from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
seattle_df['accommodates_norm']=scaler.fit_transform(seattle_df[['accommodates']]).round(2)

from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
seattle_df['bedrooms_norm']=scaler.fit_transform(seattle_df[['bedrooms']]).round(2)

from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
seattle_df['accommodates_norm']=scaler.fit_transform(seattle_df[['accommodates']]).round(2)

from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
seattle_df['reviews_norm']=scaler.fit_transform(seattle_df[['reviews']]).round(2)

from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
seattle_df['minstay_norm']=scaler.fit_transform(seattle_df[['minstay']]).round(2)
```

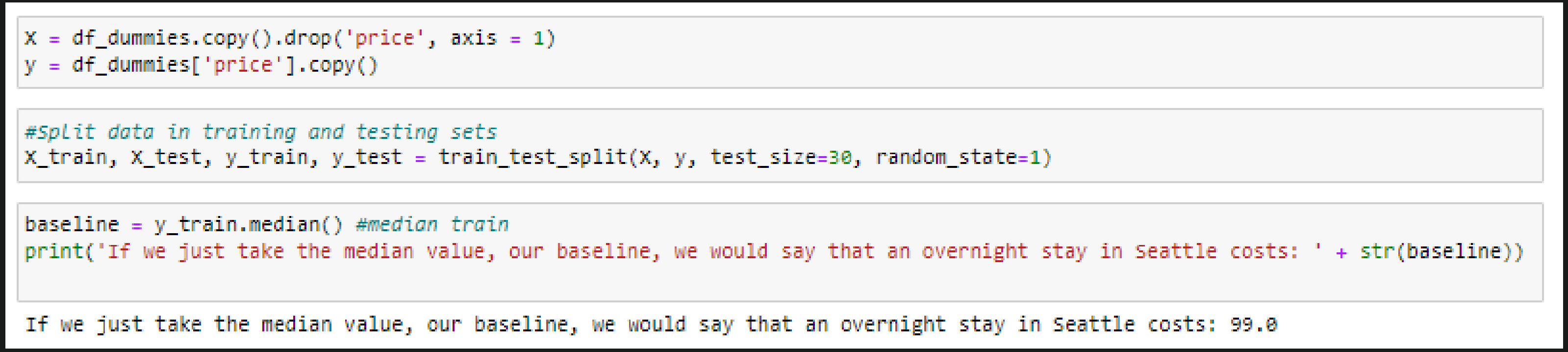
Pre-Processing

One-Hot Encoding



```
df_dummies = pd.get_dummies(seattle_df)
df_dummies.head()
```

Train-Test Split



```
X = df_dummies.copy().drop('price', axis = 1)
y = df_dummies['price'].copy()
```

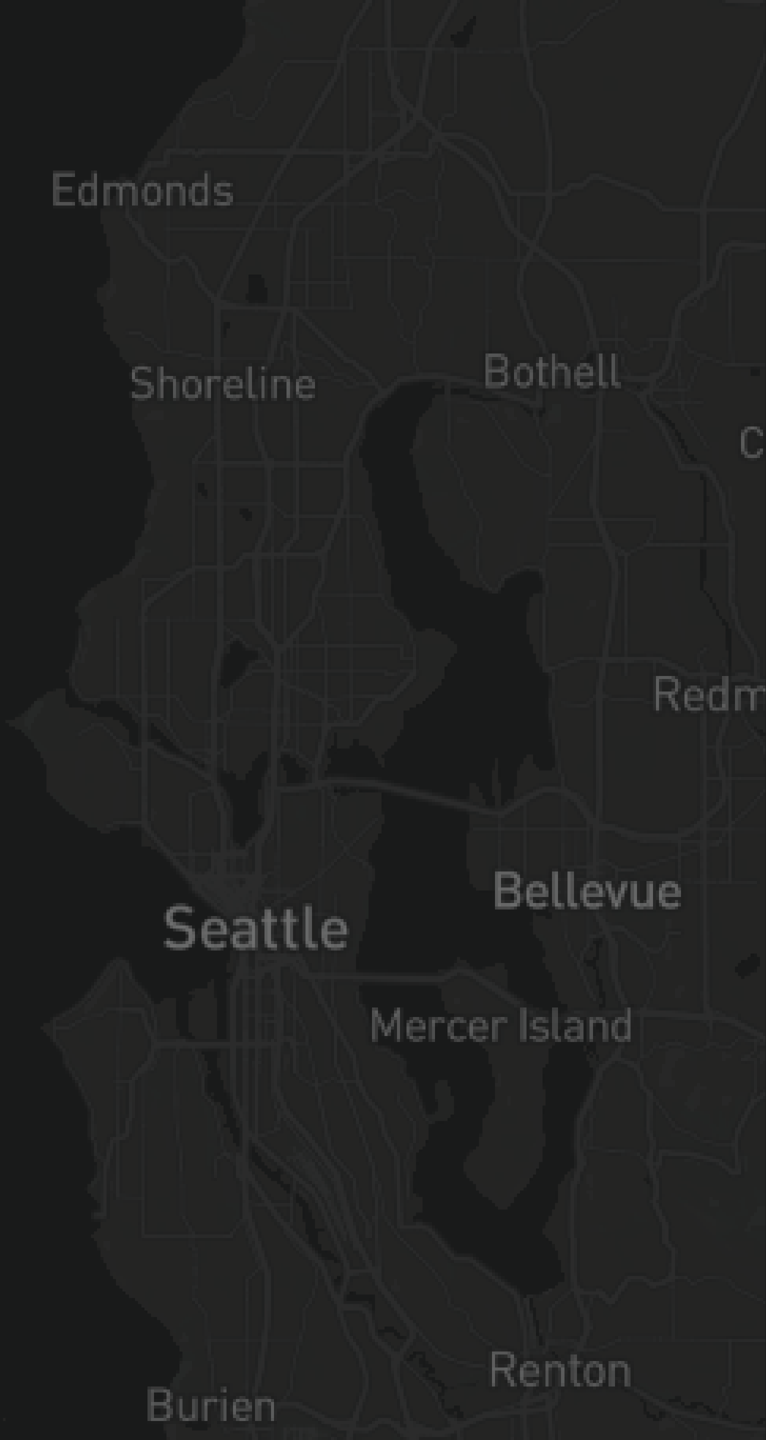
```
#Split data in training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=30, random_state=1)
```

```
baseline = y_train.median() #median train
print('If we just take the median value, our baseline, we would say that an overnight stay in Seattle costs: ' + str(baseline))
```

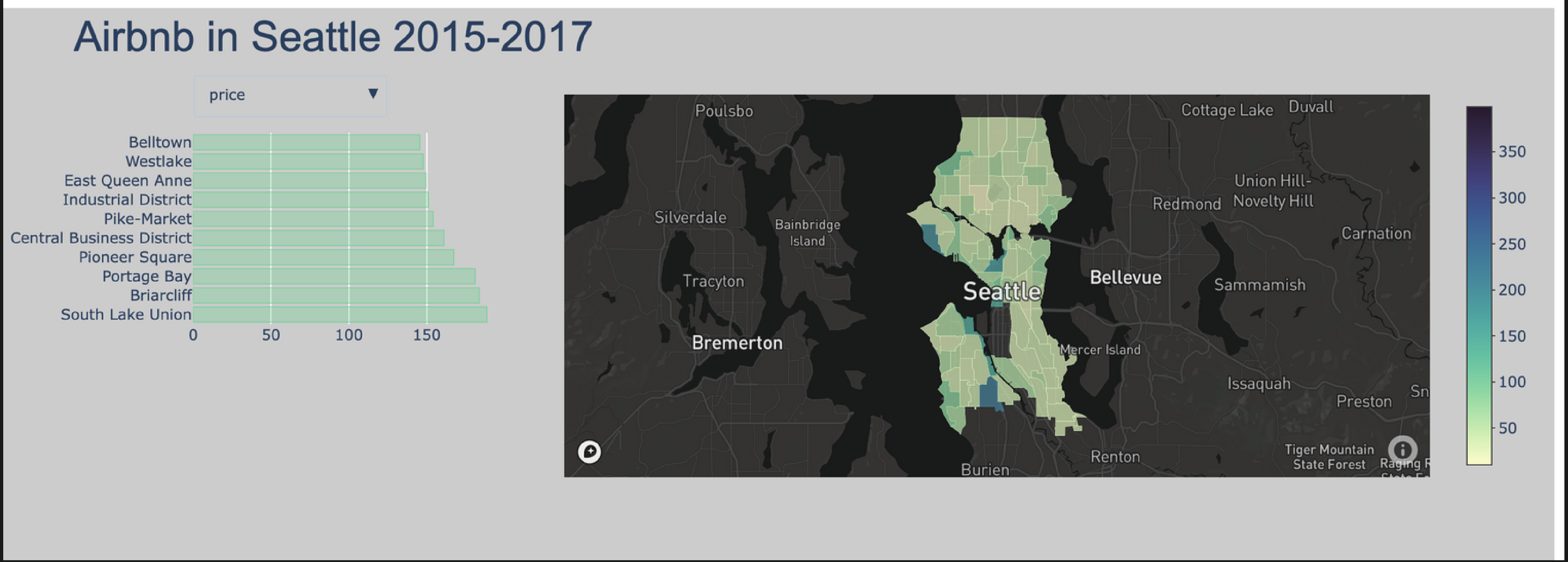
```
If we just take the median value, our baseline, we would say that an overnight stay in Seattle costs: 99.0
```

Dashboard

- Minimum number of reviews = 10
- Price < 400

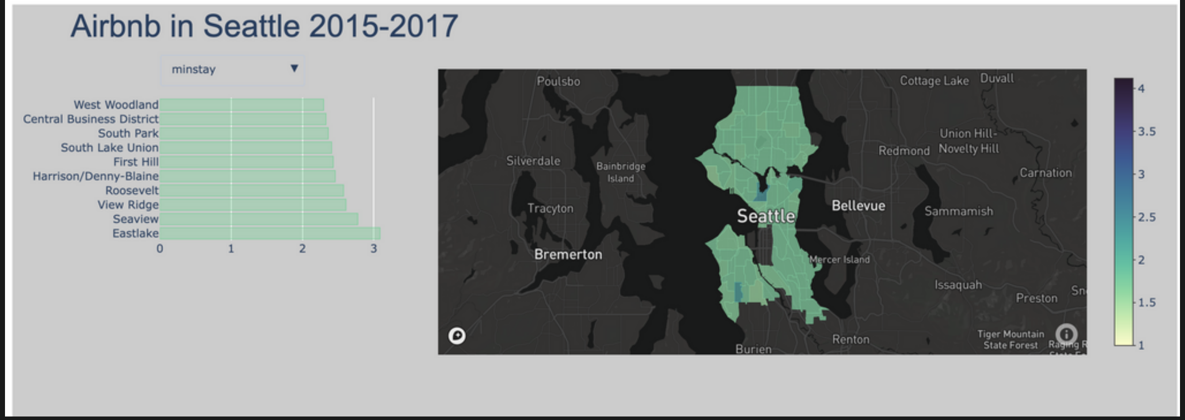


Dashboard



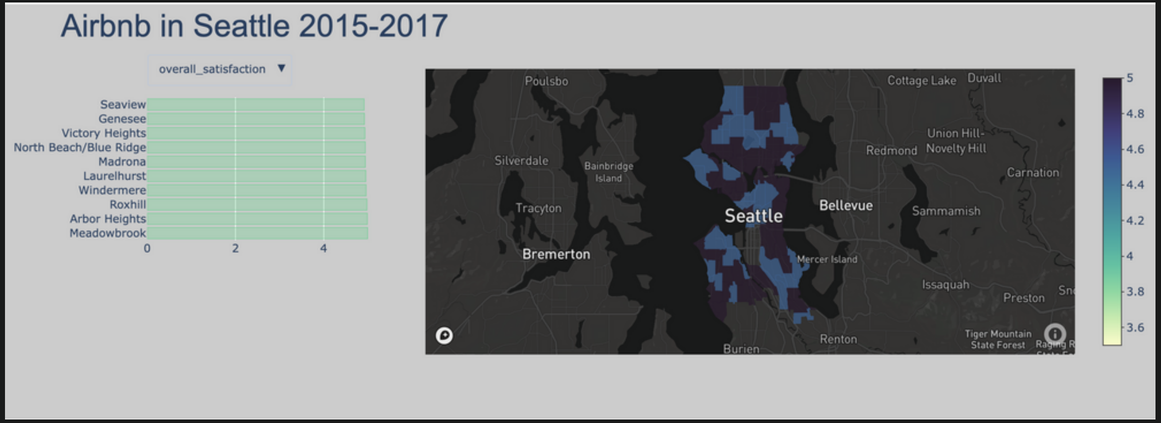
Airbnb in Seattle 2015-2017

price ▼



Airbnb in Seattle 2015-2017

minstay ▼



Airbnb in Seattle 2015-2017

overall_satisfaction ▼

Thank you!