Uniwersytet Warszawski
Wydział Nauk
Ekonomicznych

Predicting House Prices

Agenda

1. Purpose of project
2. Data
3. Problem
4. Prediction
5. Conclusion
6. Git merge (minwu3/RR_Project (github.com))

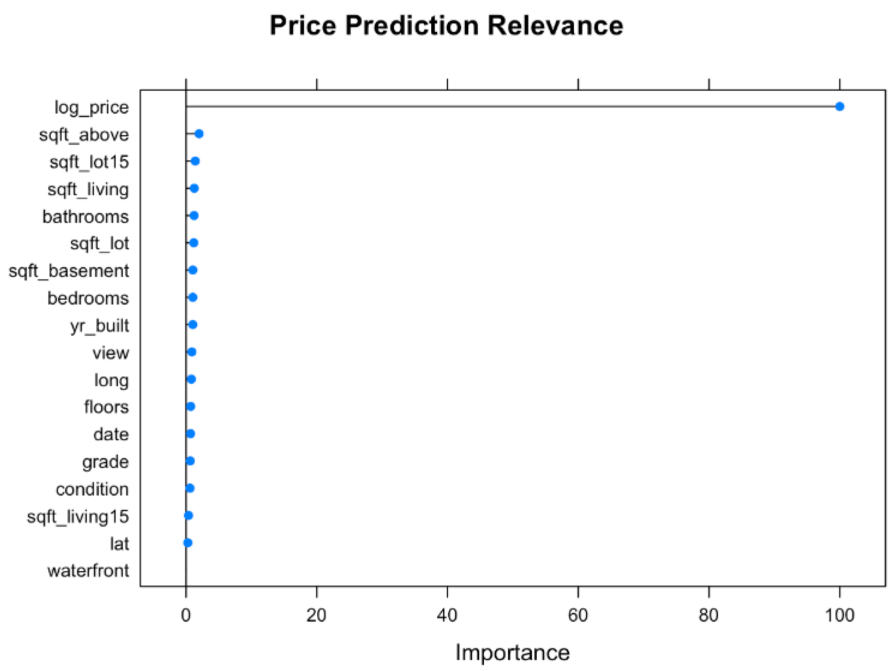
Uniwersytet Warszawski
Wydział Nauk Ekonomicznych

Purpose of project (main idea)

- In our project we will reproduce the result of initial work which was done in R language. (Previous work of my colleque)
- We transfer the code from R to Python.
- Add 2 more models.
- Analysing and choosing one which fits the most our data.

Problem of initial work

Uniwersytet Warszawski
Wydział Nauk Ekonomicznych




```
prediction_rf <- predict(rf_mod,test)
getTrainPerf(rf_mod)
```

```
##   TrainRMSE TrainRsquared TrainMAE method
## 1  11075.28     0.9986855 676.8045     rf
```

```
rse(test$price, prediction_rf)
```

```
## [1] 0.0001274139
```

Problem of initial work

- The RMSEs for the train and test subsets arecompletly different, which suggests that the model will perform differently on different data. We cannot trust the estimates.

- **Model Validation**
- Important step was missing in evaluating the quality of the model.
- Cross-validation, which gives us an idea of how the model would perform with new data for the same variables.

Here are some more reasons why it is a necesity of usage cross-validation:

- It Lets them use all of the data without sacrificing any subset (not valid for the holdout method)
- Reveals the consistency of the data and the algorithm
- Helps avoid overfitting and underfitting

Data

- You can find the dataset on Kaggle page: https://www.kaggle.com/harlfoxem/housesalesprediction
- https://kingcounty.gov/services/gis/PropResearch.aspx

- This dataset contains house sale prices for King County, which includes Seattle. It includes homes sold between May 2014 and May 2015.

Dataset

```
/Users/jcurik4725/Desktop/kc_house_data.csv
```

In [2]:
```python
df = pd.read_csv('C:/Users/tetiana.heorhiichuk/Desktop/RR_project_new/RR_Project/kc_house_data.csv', index_col = 0)
df.head()
```

Out[2]:

id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	grade	sqft_above	sqft_basement	yr_built	yr_renovated	zipcode	lat	long	sqft_living15	sqft_lot15
7129300520	20141013T000000	221900.0	3	1.00	1180	5650	1.0	0	0	3	7	1180	0	1955	0	98178	47.5112	-122.257	1340	5650
6414100192	20141209T000000	538000.0	3	2.25	2570	7242	2.0	0	0	3	7	2170	400	1951	1991	98125	47.7210	-122.319	1690	7639
5631500400	20150225T000000	180000.0	2	1.00	770	10000	1.0	0	0	3	6	770	0	1933	0	98028	47.7379	-122.233	2720	8062
2487200875	20141209T000000	604000.0	4	3.00	1960	5000	1.0	0	0	5	7	1050	910	1965	0	98136	47.5208	-122.393	1360	5000
1954400510	20150218T000000	510000.0	3	2.00	1680	8080	1.0	0	0	3	8	1680	0	1987	0	98074	47.6168	-122.045	1800	7503

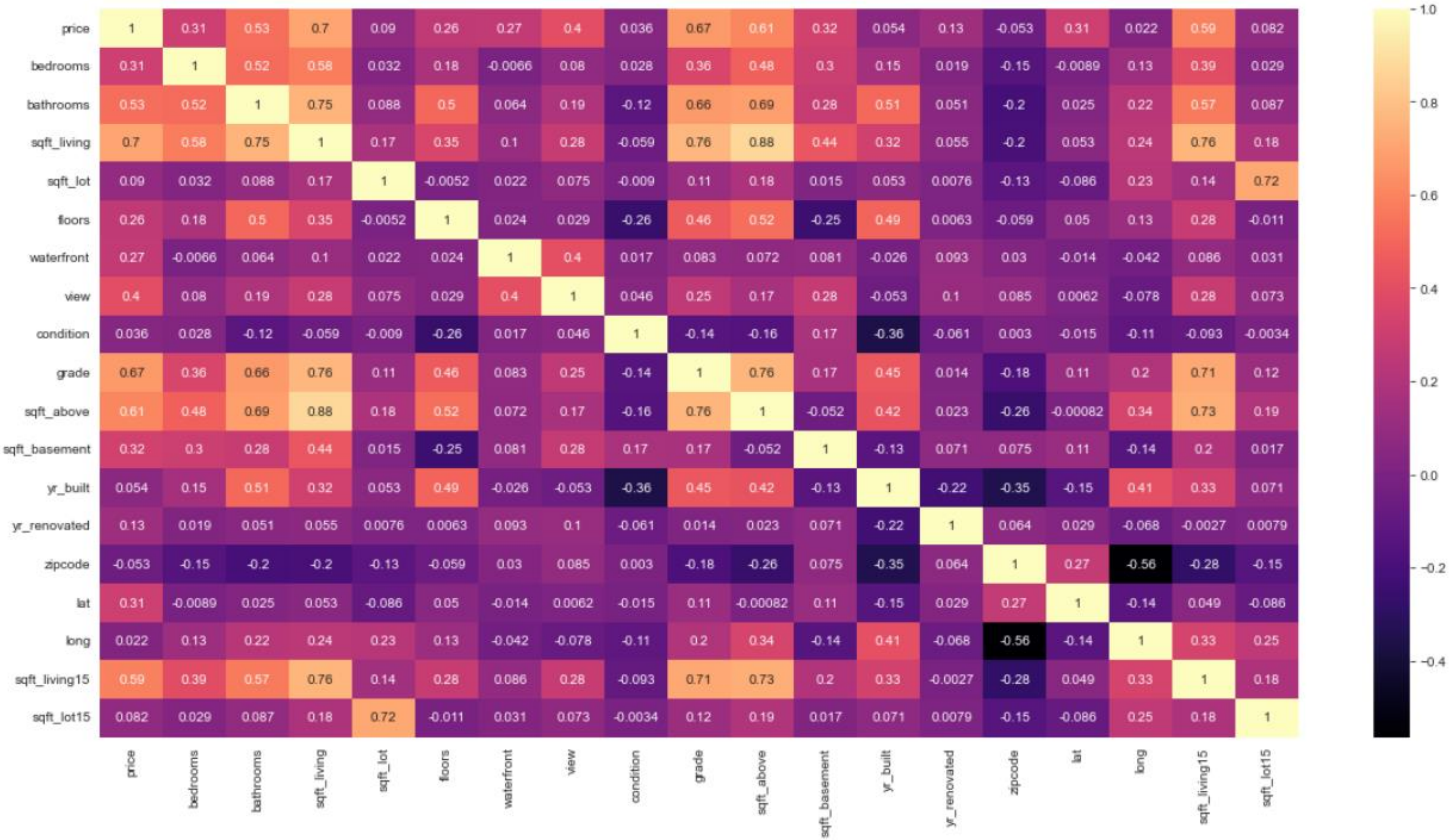
Dataset

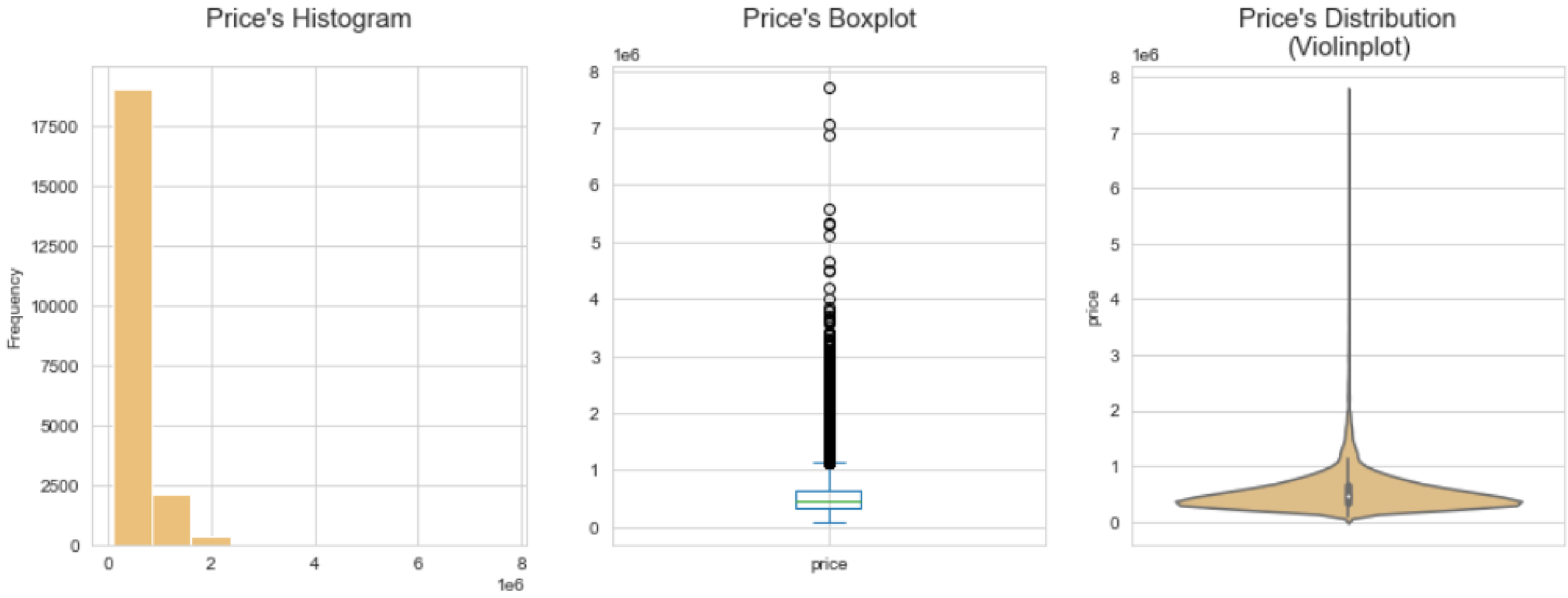
Uniwersytet Warszawski
Wydział Nauk Ekonomicznych

2.1 Definitions of the Variables

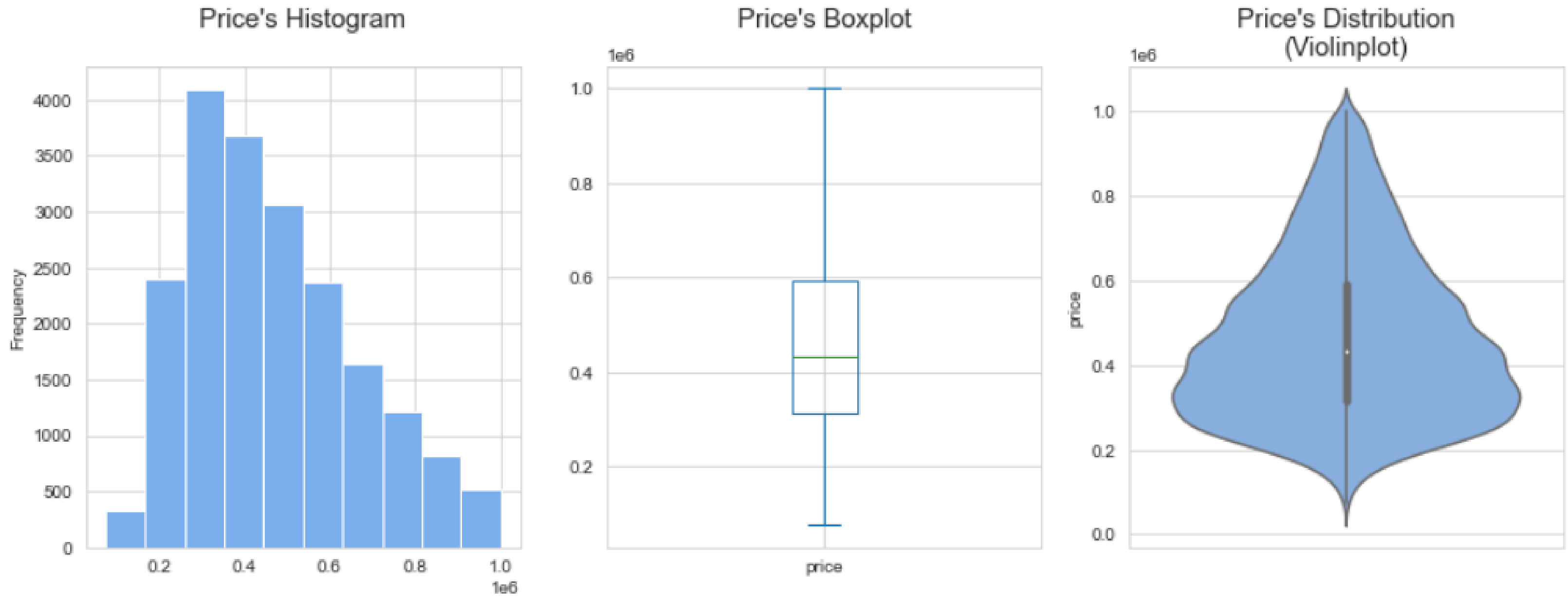
- id - Unique ID for each home sold
- date - Date of the home sale
- price - Price of each home sold
- bedrooms - Number of bedrooms
- bathrooms - Number of bathrooms, where .5 accounts for a room with a toilet but no shower
- sqft_living - Square footage of the apartments interior living space
- sqft_lot - Square footage of the land space
- floors - Number of floors
- waterfront - A dummy variable for whether the apartment was overlooking the waterfront or not
- view - An index from 0 to 4 of how good the view of the property was
- condition - An index from 1 to 5 on the condition of the apartment,
- grade - An index from 1 to 13, where 1-3 falls short of building construction and design, 7 has an average level of construction and design, and 11-13 have a high quality level of construction and design
- sqft_above - The square footage of the interior housing space that is above ground level
- sqft_basement - The square footage of the interior housing space that is below ground level
- yr_built - The year the house was initially built
- yr_renovated - The year of the house's last renovation
- zipcode - What zipcode area the house is in
- lat - Lattitude
- long - Longitude
- sqft_living15 - The square footage of interior housing living space for the nearest 15 neighbors
- sqft_lot15 - The square footage of the land lots of the nearest 15 neighbors

Data Visualization

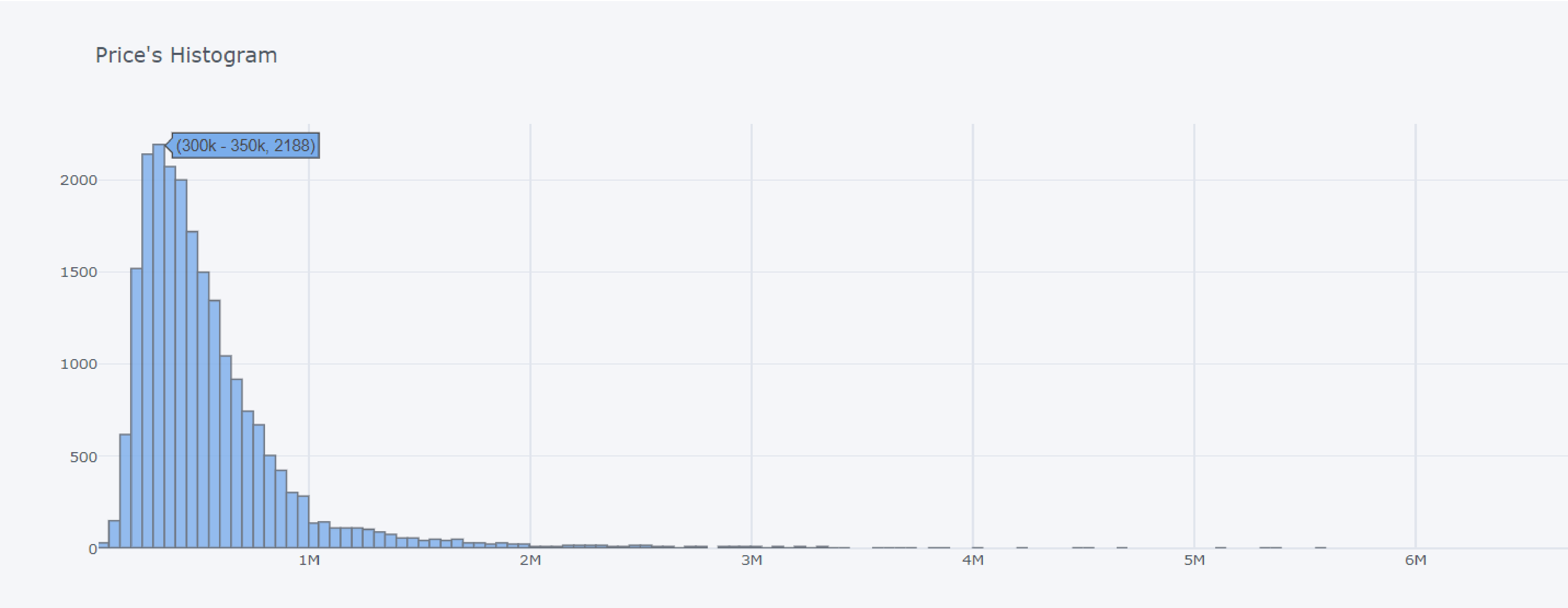




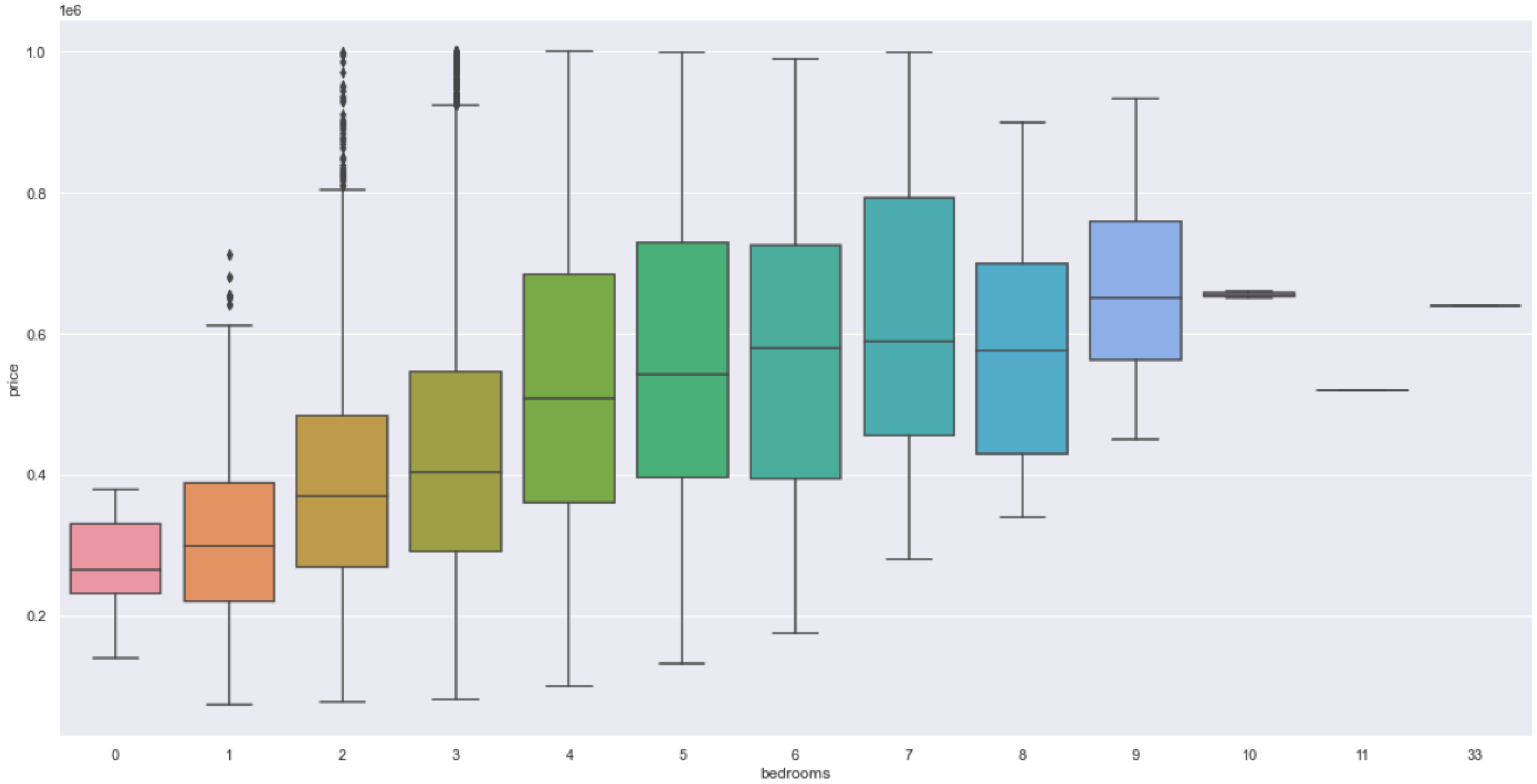
Price's Histogram
Frequency
1e6
Price's Boxplot
1e6
price
Price's Distribution (Violinplot)
1e6
price



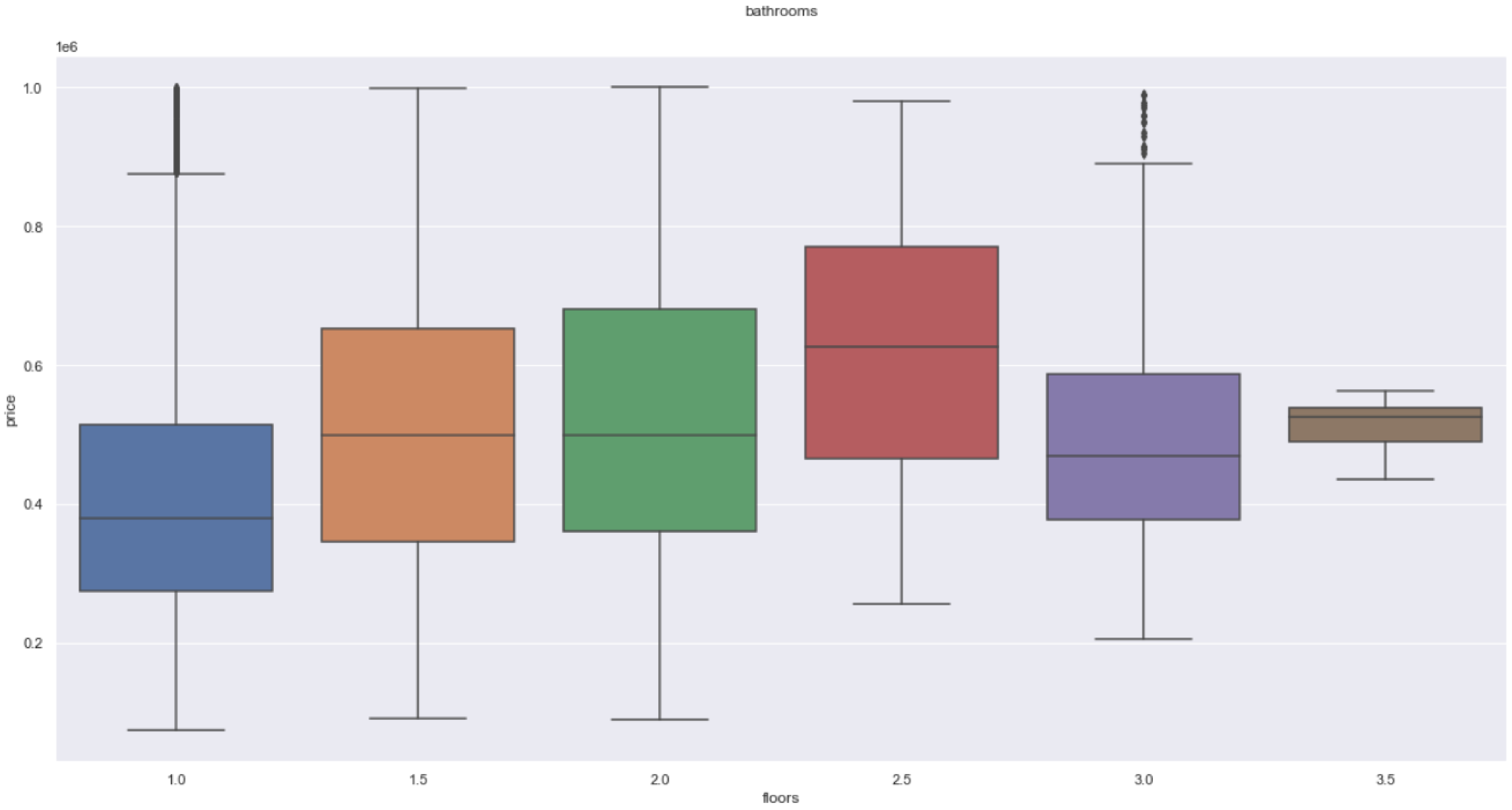
Price's Histogram
Frequency
1e6
Price's Boxplot
1e6
price
Price's Distribution (Violinplot)
1e6
price

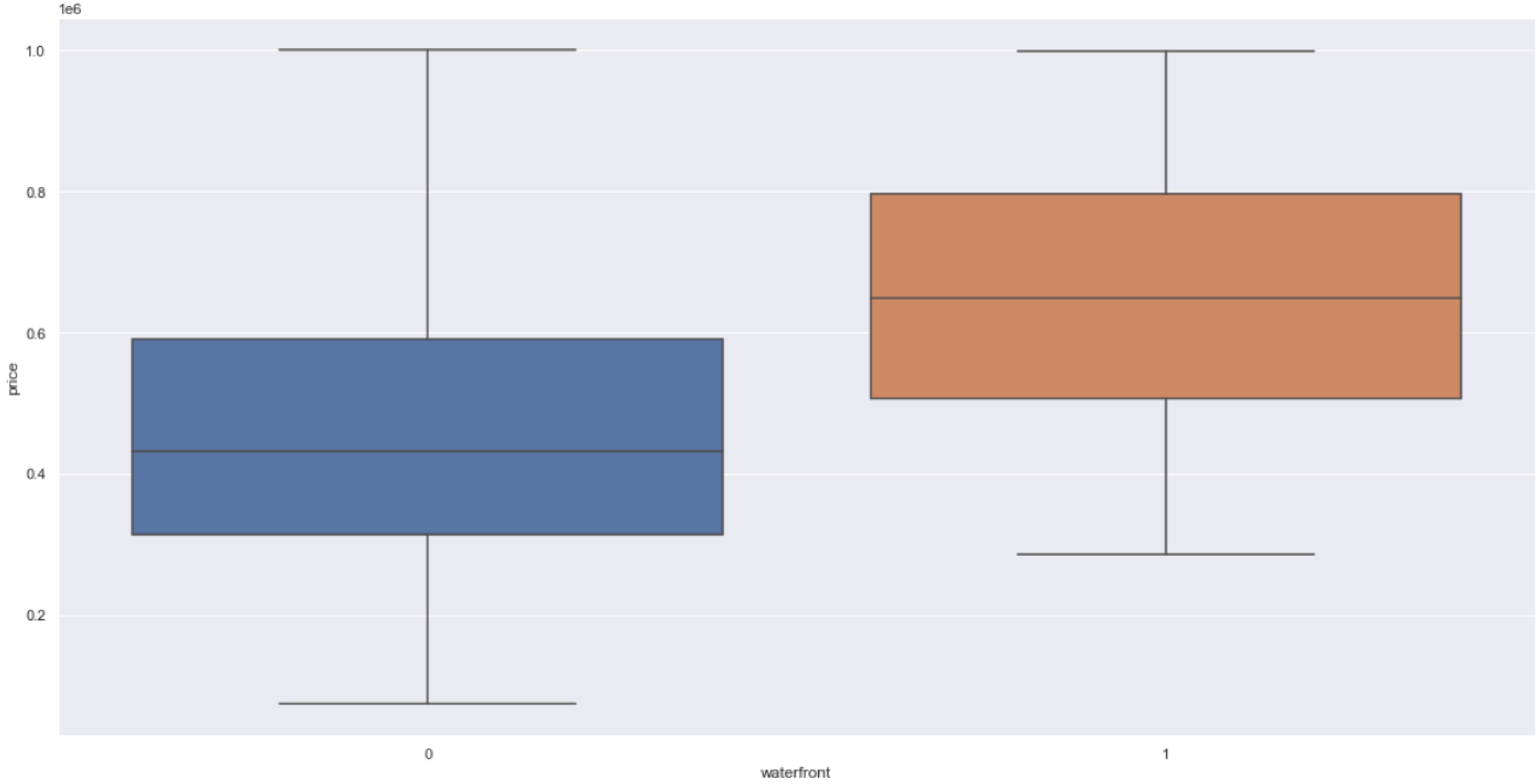


Price's Histogram
(300k - 350k, 2188)
2000
1500
1000
500
0
1M
2M
3M
4M
5M
6M

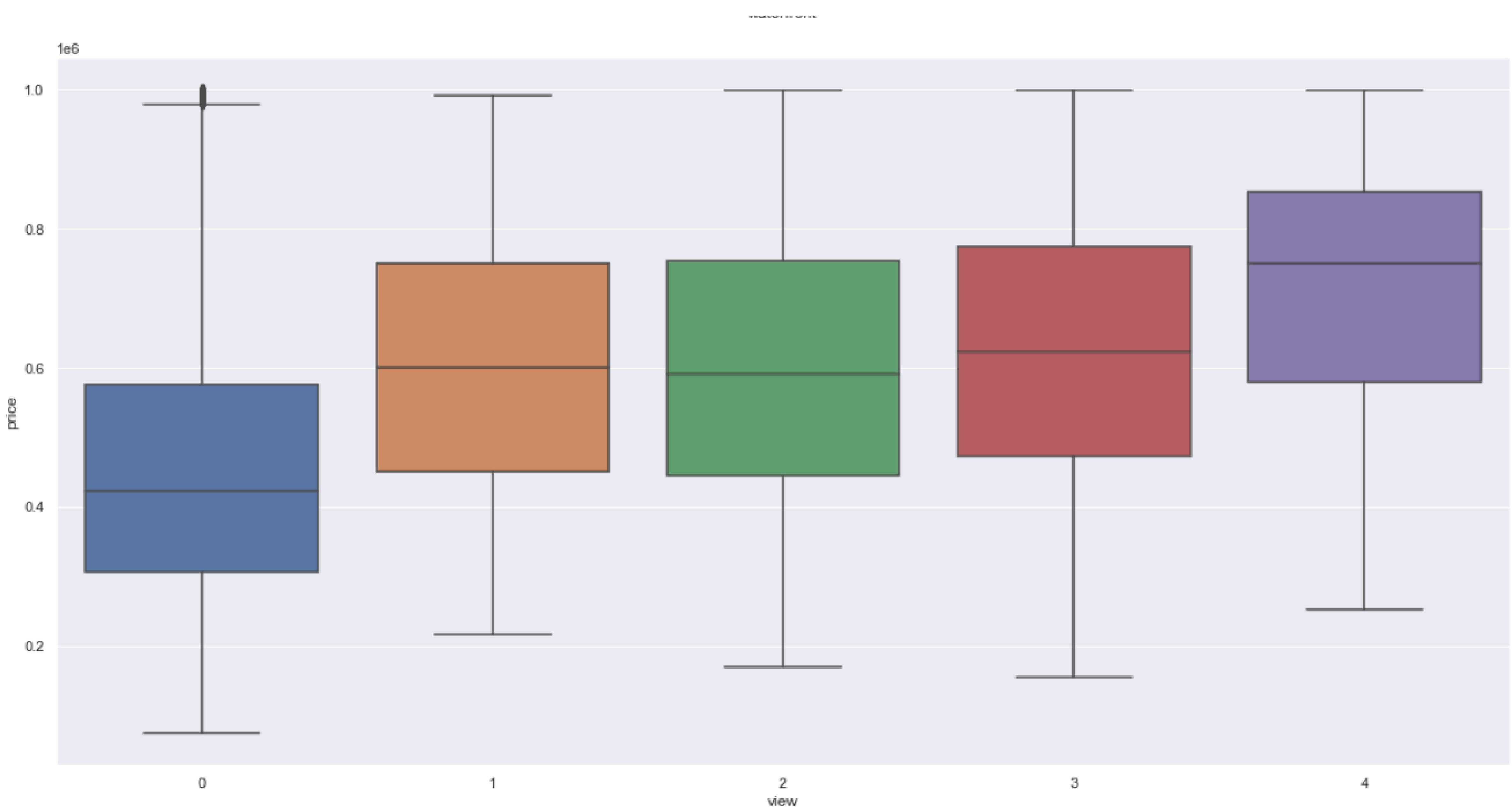


1e6
1.0
0.8
0.6
0.4
0.2
price
0
1
2
3
4
5
6
7
8
9
10
11
33
bedrooms

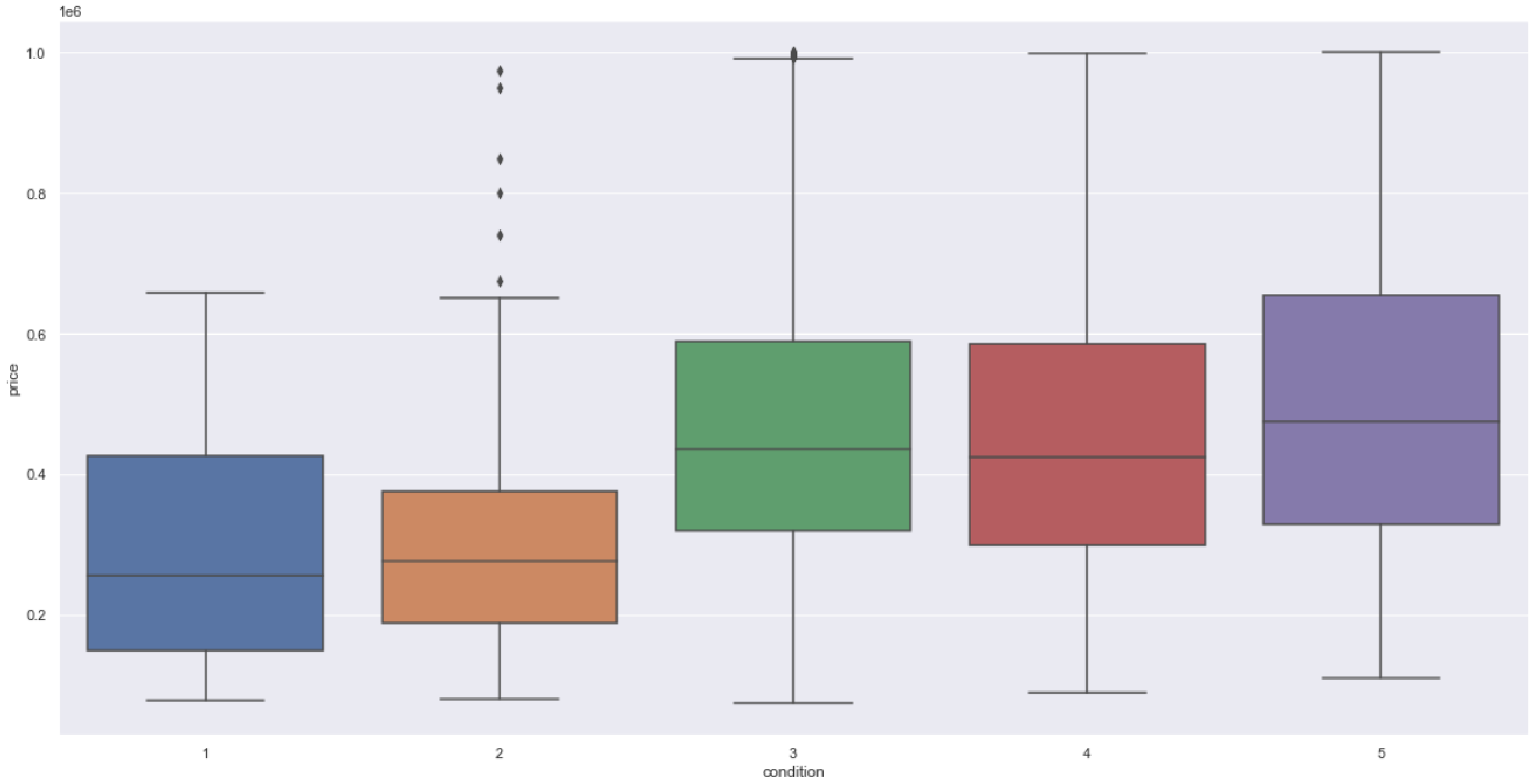

bathrooms
1e6
1.0
0.8
0.6
0.4
0.2
price
1.0
1.5
2.0
2.5
3.0
3.5
floors



1e6
1.0
0.8
0.6
0.4
0.2
price
0
1
waterfront



1e6
1.0
0.8
0.6
0.4
0.2
price
0
1
2
3
4
view



1e6
1.0
0.8
0.6
0.4
0.2
price
1
2
3
4
5
condition

Building a model

In this process, we are going to build and train five different types of linear regression models :

- - OLS model,
- -Ridge regression model,
- -Lasso regression model,
- -Bayesian regression model,
- -Elastic Net regression model.

For all the models, we are going to use the pre-built algorithms provided by the scikit-learn package in python. And the process for all the models are the same, first, we define a variable to store the model algorithm, next, we fit the train set variables into the model, and finally make some predictions in the test set.

Evaluating the model

- We can see that, every model while rounding the output values will result in a score of 0.65 (65.1%) or 0.65 (65%) which means our model still have room for improvement on our dataset.
- Let’s make a polynomial linear regression model to find out if there are some features with a non-linear relationship

40]:

```
# Return the coefficient of determination  of the prediction.
ols_r2 = ols.score(X_test, y_test)
ridge_r2 = ridge.score(X_test, y_test)
lasso_r2 = lasso.score(X_test, y_test)
print(cl('R-SQUARED:', attrs = ['bold']))
print('R-Squared of OLS model is {}'.format(ols_r2))
print('R-Squared of Ridge model is {}'.format(ridge_r2))
print('R-Squared of Lasso model is {}'.format(lasso_r2))
```

```
R-SQUARED:
R-Squared of OLS model is 0.6515929603856018
R-Squared of Ridge model is 0.651631874284701
R-Squared of Lasso model is 0.6515929794742606
```

Evaluating the model

3]:

```
print(cl('EXPLAINED VARIANCE SCORE:', attrs = ['bold']))
print('----------------------------------------------------------------')
print(cl('Explained Variance Score of OLS model is {}'.format(evs(y_test, yhat_test_pipe))))
print(cl('Explained Variance Score of Ridge model is {}'.format(evs(y_test, yhat_ridge))))
print(cl('Explained Variance Score of Lasso model is {}'.format(evs(y_test, yhat_lasso))))
```

```
EXPLAINED VARIANCE SCORE:
----------------------------------------------------------------
Explained Variance Score of OLS model is 0.7299936095524115
Explained Variance Score of Ridge model is 0.7309164844743152
Explained Variance Score of Lasso model is 0.7314865739973428
```

Cross validation

Uniwersytet Warszawski
Wydział Nauk Ekonomicznych

```
cv_scores = cross_val_score(estimator=best_lasso, X=X, y=y, cv=5)
cv_scores
```

```
array([0.76065908, 0.74096726, 0.71101414, 0.71707866, 0.73993212])
```

```
pd.Series(cv_scores).describe()
```

```
count    5.000000
mean     0.733930
std      0.020057
min      0.711014
25%      0.717079
50%      0.739932
75%      0.740967
max      0.760659
dtype: float64
```

```
cv_scores_ridge = cross_val_score(estimator=best_ridge, X=X, y=y, cv=5)
cv_scores_ridge
```

```
array([0.7594072 , 0.74252124, 0.70992529, 0.71228485, 0.73826996])
```

CONCLUSION

- The best model we achieved had a mean cv score of **0.760659** and was a **Lasso Regression model** with as 1000.

Git merge conflict

Uniwersytet Warszawski
Wydział Nauk Ekonomicznych

Marge Conflict

We occurred a conflict while we were merging branches “master” and “newbranch”.

```
(base) jhen@Jhens-Air RR % git merge newbranch
Auto-merging ML-Regression_python.ipynb
CONFLICT (content): Merge conflict in ML-Regression_python.ipynb
Automatic merge failed; fix conflicts and then commit the result.
(base) jhen@Jhens-Air RR %
```

$ git log —oneline —graph —all

check the branch

```
(base) jhen@Jhens-Air RR % git log --oneline --graph --all
* 51f47a6 (HEAD -> newbranch) minor edition
| * b30887b (origin/master) Last check
| * c10de95 Last check, adding comments and explanation
|/
* a5bf303 (master) Add Cross Validation
* 652f86a Added model evaluation
* 120de3c 5.2.2 Ridge Regression
* 3ed513b Added Feature Selection. Builded OLS Regression and Lasso Regression
* 7263366 Add Data Virsualization - heatmap, histogram, etc.
* 2f62518 Basic analysis missing values and data types
* c438714 Add python version file - import libraries and load data
* 640ff06 Add python version file - import libraries and data
* 26ee292 Add the original project and dataset
* f531dab Delete all the test files
* 9f72ac1 Add bbbb file for testing
* 8703b43 This is a second change
* 7d8c35a This is a test change
* 7483937 add words for testing
* 0df0138 add aaa for test
```

Uniwersytet Warszawski
Wydział Nauk Ekonomicznych

$ git checkout master
# main branch

```
RR — -zsh — 92×34
* 26ee292 Add the original project and dataset
* f531dab Delete all the test files
* 9f72ac1 Add bbbb file for testing
* 8703b43 This is a second change
* 7d8c35a This is a test change
* 7483937 add words for testing
* 0df0138 add aaa for test
(base) jhen@Jhens-Air RR % git checkout master
Switched to branch 'master'
Your branch is behind 'origin/master' by 2 commits, and can be fast-forwarded.
  (use "git pull" to update your local branch)
```

$ git diff # check which lines are different.

```
RR — less ‹ git diff — 92×34
~
~
~
~
~
~
~
(END)
@@@ -24808,8 -24646,19 +24808,15 @@@
        }
       ],
       "source": [
++<<<<<<< HEAD
++=======
+      "method_all = [\"OLS\",\"LASSO\",\"RIDGE\"]\n",
+      "prediction_all = [yhat_test_pipe_df,yhat_lasso_df,yhat_ridge_df]\n",
+      "\n",
+      "print(prediction_all)\n",
++>>>>>>> newbranch
        "print (yhat_lasso_df.describe())\n"
       ]
 -  },
 -  {
 -   "cell_type": "code",
 -   "execution_count": null,
 -   "metadata": {},
 -   "outputs": [],
 -   "source": []
    }
   ],
   "metadata": {
~
~
~
```

$ git log —oneline —graph —all

check the branch

```
[master 92aca61] merged newbranch
(base) jhen@Jhens-Air RR % git log --oneline --graph --all
*   92aca61 (HEAD -> master) merged newbranch
|\
| * 51f47a6 (newbranch) minor edition
* | b30887b (origin/master) Last check
* | c10de95 Last check, adding comments and explanation
|/
* a5bf303 Add Cross Validation
* 652f86a Added model evaluation
* 120de3c 5.2.2 Ridge Regression
* 3ed513b Added Feature Selection. Builded OLS Regression and Lasso Regression
* 7263366 Add Data Virsualization - heatmap, histogram, etc.
* 2f62518 Basic analysis missing values and data types
* c438714 Add python version file - import libraries and load data
* 640ff06 Add python version file - import libraries and data
* 26ee292 Add the original project and dataset
* f531dab Delete all the test files
* 9f72ac1 Add bbbb file for testing
* 8703b43 This is a second change
* 7d8c35a This is a test change
* 7483937 add words for testing
* 0df0138 add aaa for test
```

Uniwersytet Warszawski
Wydział Nauk Ekonomicznych

$ git merge new branch

#Merge newbranch to master

```
RR — -zsh — 92×34
* 26ee292 Add the original project and dataset
* f531dab Delete all the test files
* 9f72ac1 Add bbbb file for testing
* 8703b43 This is a second change
* 7d8c35a This is a test change
* 7483937 add words for testing
* 0df0138 add aaa for test
(base) jhen@Jhens-Air RR % git merge newbranch
Auto-merging ML-Regression_python.ipynb
CONFLICT (content): Merge conflict in ML-Regression_python.ipynb
Automatic merge failed; fix conflicts and then commit the result.
(base) jhen@Jhens-Air RR % git add git ML-Regression_python.ipynb
fatal: pathspec 'git' did not match any files
(base) jhen@Jhens-Air RR % git add ML-Regression_python.ipynb
(base) jhen@Jhens-Air RR % git commit -m "merged newbranch"
[master 92aca61] merged newbranch
(base) jhen@Jhens-Air RR % git log --oneline --graph --all
*   92aca61 (HEAD -> master) merged newbranch
|\
| * 51f47a6 (newbranch) minor edition
* | b30887b (origin/master) Last check
* | c10de95 Last check, adding comments and explanation
|/
* a5bf303 Add Cross Validation
* 652f86a Added model evaluation
* 120de3c 5.2.2 Ridge Regression
* 3ed513b Added Feature Selection. Builded OLS Regression and Lasso Regression
* 7263366 Add Data Virsualization - heatmap, histogram, etc.
* 2f62518 Basic analysis missing values and data types
* c438714 Add python version file - import libraries and load data
* 640ff06 Add python version file - import libraries and data
* 26ee292 Add the original project and dataset
* f531dab Delete all the test files
* 9f72ac1 Add bbbb file for testing
```

$ git commit -m “merge newbranch”

```
RR — -zsh — 92×34
* c438714 Add python version file - import libraries and load data
* 640ff06 Add python version file - import libraries and data
* 26ee292 Add the original project and dataset
* f531dab Delete all the test files
* 9f72ac1 Add bbbb file for testing
* 8703b43 This is a second change
* 7d8c35a This is a test change
* 7483937 add words for testing
* 0df0138 add aaa for test
(base) jhen@Jhens-Air RR % git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 831 bytes | 831.00 KiB/s, done.
Total 6 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 3 local objects.
To github.com:minwu3/RR_Project.git
   b30887b..92aca61  master -> master
(base) jhen@Jhens-Air RR % git add ML-Regression_python.ipynb
(base) jhen@Jhens-Air RR % git commit -m "final edition"
[master 0ef90f8] final edition
 1 file changed, 7 deletions(-)
(base) jhen@Jhens-Air RR % git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 391 bytes | 195.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:minwu3/RR_Project.git
   92aca61..0ef90f8  master -> master
(base) jhen@Jhens-Air RR %
```