



# Python Projects

Anime ALS recommendation  
and statistical analysis

NASDAQ missing dates

NYC flights EDA

London houses prices

# Anime ALS recommendation and statistical analysis

```
[58]: mostVotedAnime = rating[rating['rating']>=5]
```

```
[59]: mostVotedAnime.groupby('user_id').count().sort_values('anime_id', ascending=False)
```

```
[59]:
```

	anime_id	rating
--	----------	--------

user_id		
---------	--	--

42635	3747	3747
-------	------	------

53698	2905	2905
-------	------	------

57620	2689	2689
-------	------	------

59643	2632	2632
-------	------	------

51693	2621	2621
-------	------	------

...	...	...
-----	-----	-----

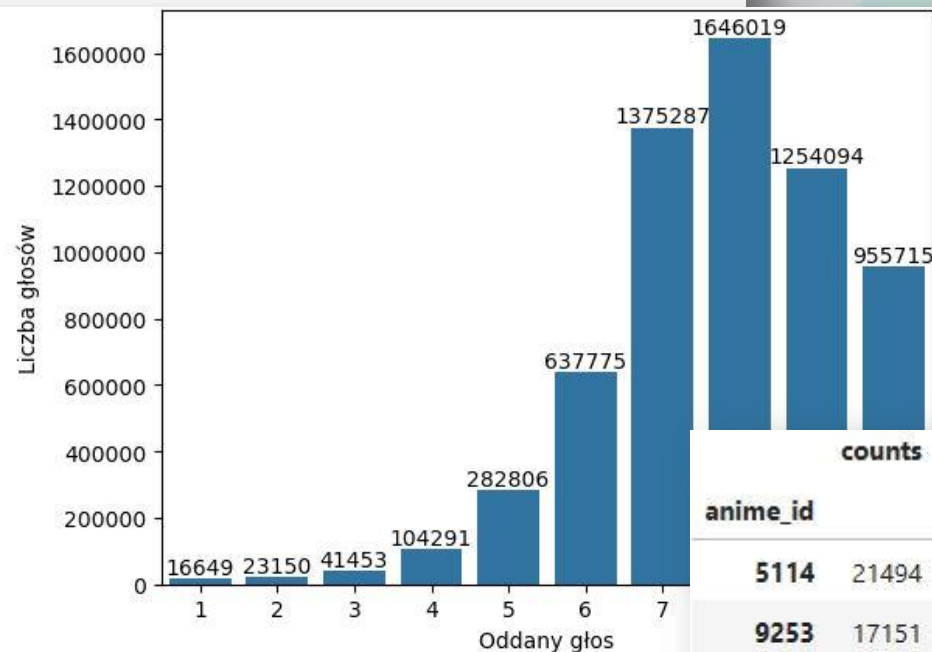
66747	1	1
-------	---	---

35463	1	1
-------	---	---

55557	1	1
-------	---	---

55563	1	1
-------	---	---

22208	1	1
-------	---	---



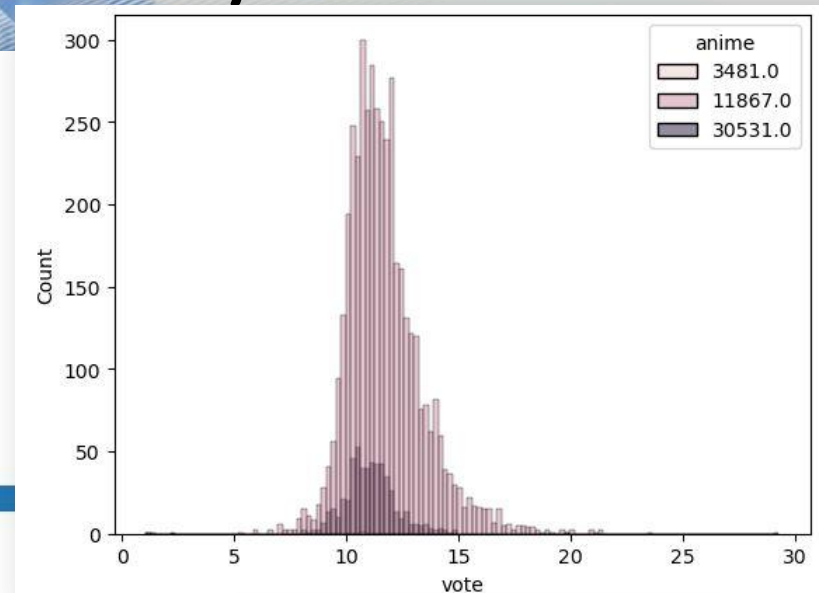
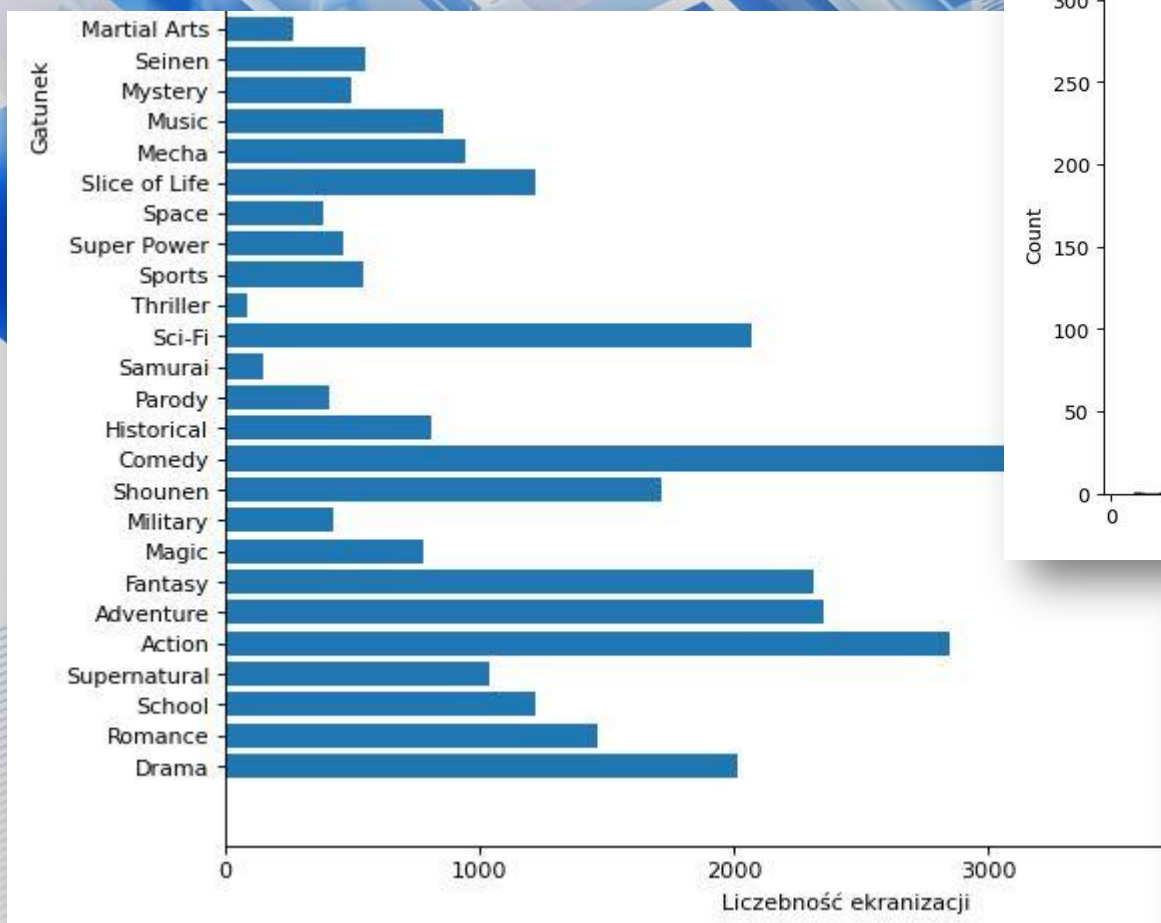
```
[98]: mostVotedAnime['rating'].mean()
```

```
[98]: 7.808496570825244
```

	counts	ratingSum	ratingMean
anime_id			
5114	21494	200383	9.322741
9253	17151	158841	9.261326
4181	15518	142227	9.165292
2904	21124	191380	9.059837
30276	11323	101224	8.939680



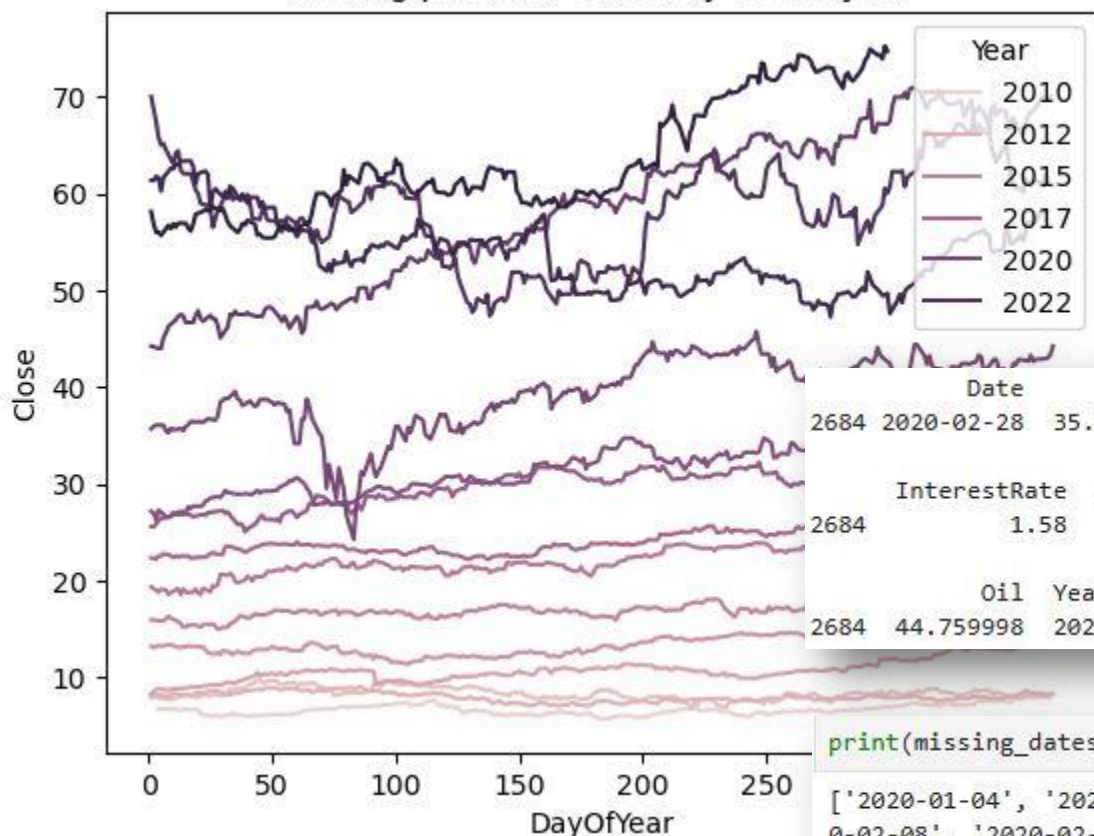
# Anime ALS recommendation and statistical analysis



user_id	anime_id	rating	prediction
148	225	7	5.51359
148	265	7	8.145976
148	356	6	5.727167
148	384	8	5.783889
148	1575	8	8.378609
148	2904	9	8.538317
148	4224	8	7.3941135
148	5114	8	8.998776
148	7674	9	8.064965
148	9130	7	7.903865

# NASDAQ missing dates

Closing price for each day of the year



Date	Open	High	Low	Close	Volume \
2684 2020-02-28	35.076668	35.150002	33.490002	34.183334	6843000.0

InterestRate	ExchangeRate	VIX	TEDSpread	EFFR	Gold \
2684 1.58	1.1001	40.11	0.21	1.58	1564.099976

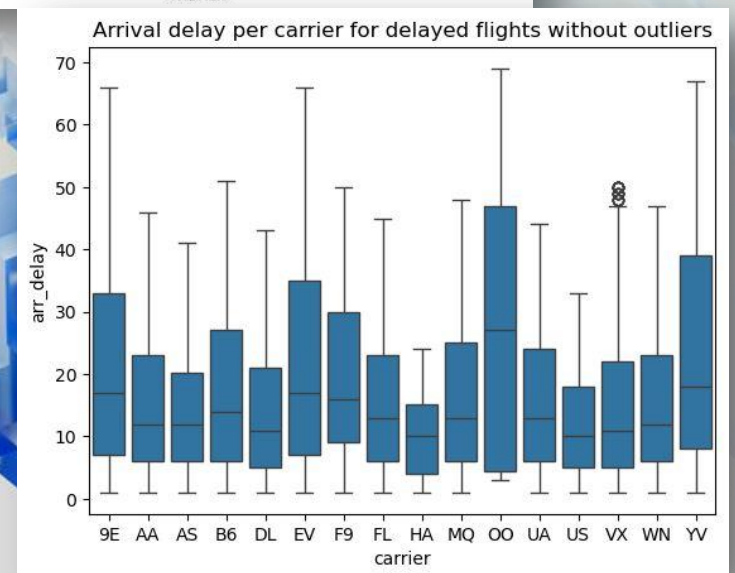
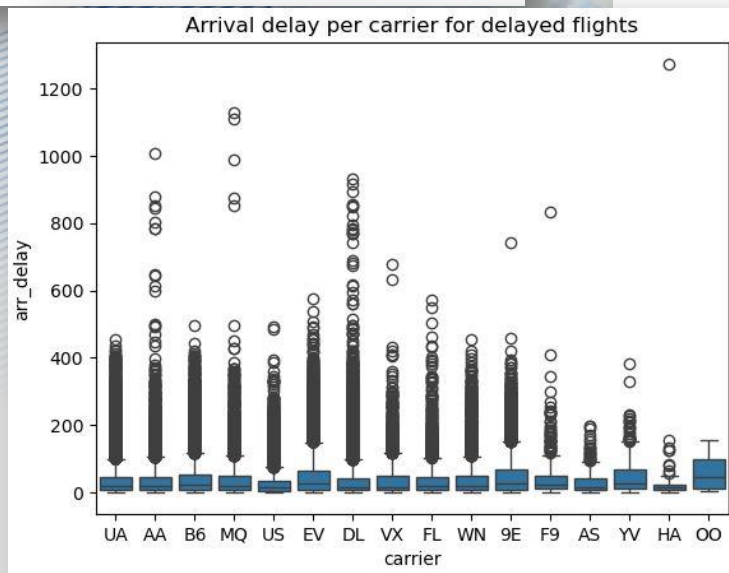
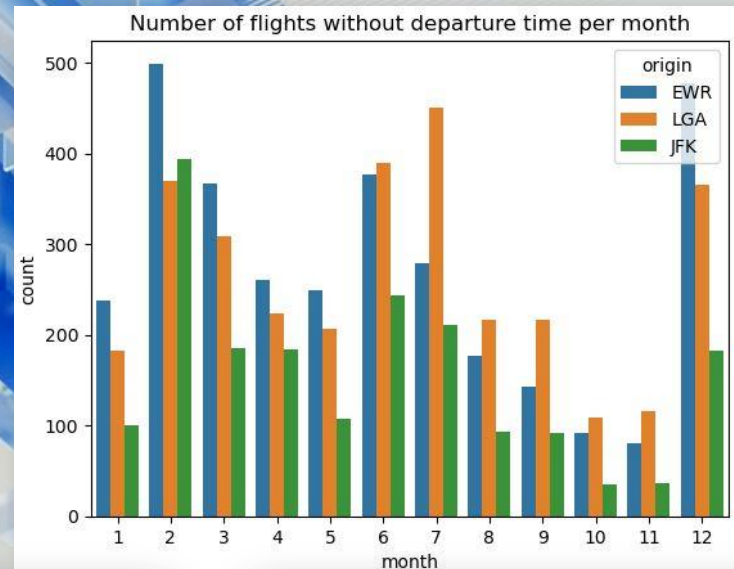
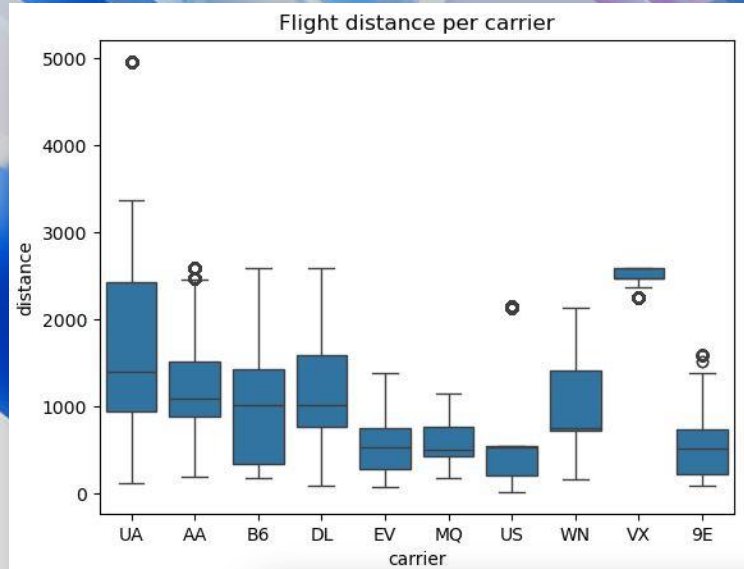
Oil	Year	DayOfYear
2684 44.759998	2020	59

```
print(missing_dates)
```

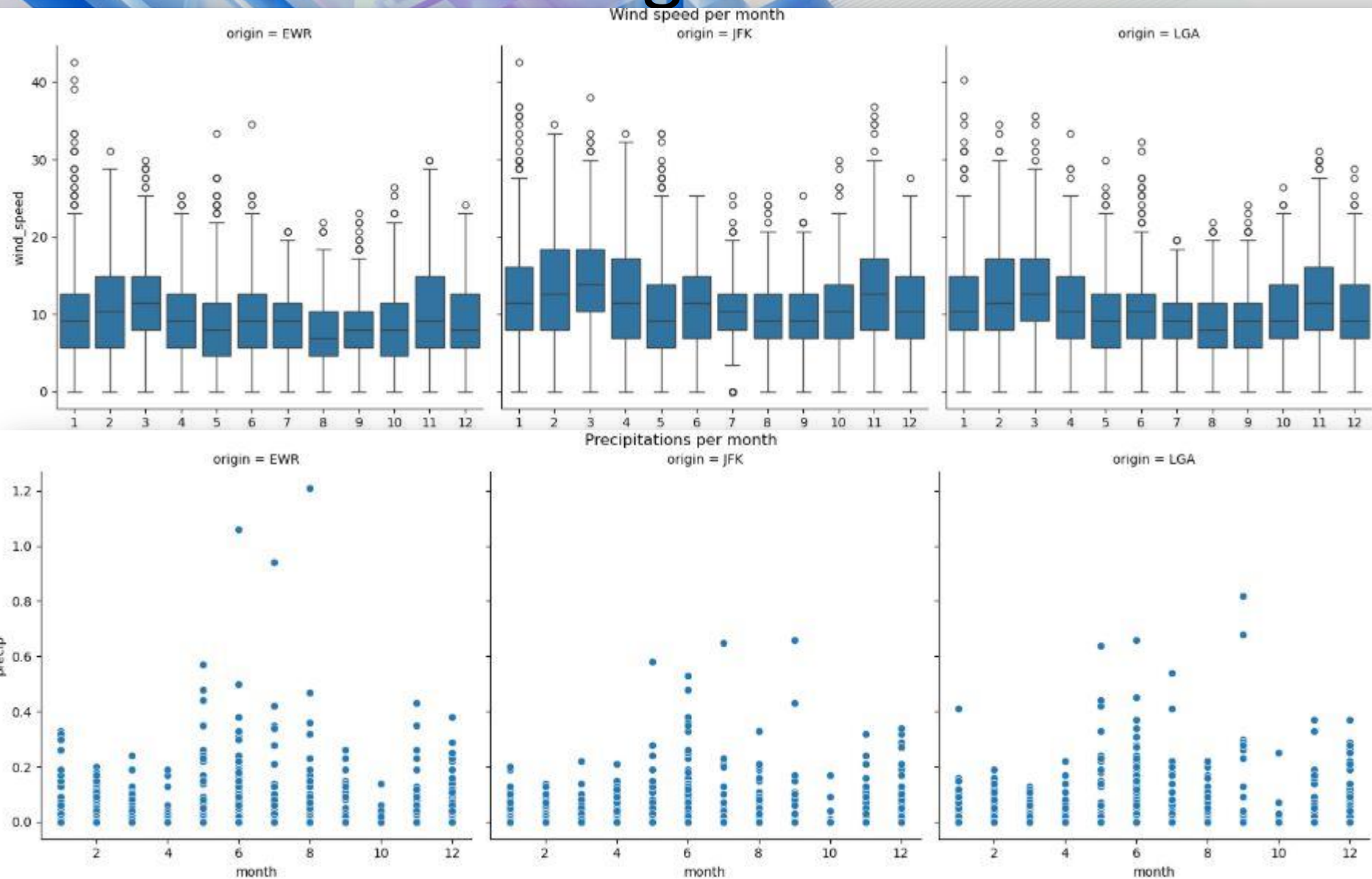
```
['2020-01-04', '2020-01-05', '2020-01-11', '2020-01-12', '2020-01-18', '2020-01-19', '2020-01-20', '2020-01-21', '2020-01-22', '2020-01-23', '2020-01-24', '2020-01-27', '2020-01-28', '2020-01-29', '2020-01-30', '2020-02-03', '2020-02-04', '2020-02-05', '2020-02-06', '2020-02-09', '2020-02-10', '2020-02-11', '2020-02-12', '2020-02-13', '2020-02-16', '2020-02-17', '2020-02-18', '2020-02-19', '2020-02-20', '2020-02-23', '2020-02-24', '2020-02-25', '2020-02-26', '2020-02-27', '2020-02-28', '2020-02-29', '2020-03-02', '2020-03-03', '2020-03-04', '2020-03-05', '2020-03-06', '2020-03-09', '2020-03-10', '2020-03-11', '2020-03-12', '2020-03-13', '2020-03-16', '2020-03-17', '2020-03-18', '2020-03-19', '2020-03-20', '2020-03-23', '2020-03-24', '2020-03-25', '2020-03-26', '2020-03-27', '2020-03-30', '2020-03-31', '2020-04-01', '2020-04-02', '2020-04-03', '2020-04-06', '2020-04-07', '2020-04-08', '2020-04-09', '2020-04-12', '2020-04-13', '2020-04-14', '2020-04-15', '2020-04-16', '2020-04-17', '2020-04-20', '2020-04-21', '2020-04-22', '2020-04-23', '2020-04-24', '2020-04-27', '2020-04-28', '2020-04-29', '2020-04-30', '2020-05-01', '2020-05-04', '2020-05-05', '2020-05-06', '2020-05-07', '2020-05-11', '2020-05-12', '2020-05-13', '2020-05-14', '2020-05-15', '2020-05-18', '2020-05-19', '2020-05-20', '2020-05-21', '2020-05-22', '2020-05-26', '2020-05-27', '2020-05-28', '2020-05-29', '2020-05-31', '2020-06-01', '2020-06-02', '2020-06-03', '2020-06-04', '2020-06-05', '2020-06-08', '2020-06-09', '2020-06-10', '2020-06-11', '2020-06-12', '2020-06-15', '2020-06-16', '2020-06-17', '2020-06-18', '2020-06-19', '2020-06-22', '2020-06-23', '2020-06-24', '2020-06-25', '2020-06-26', '2020-06-29', '2020-06-30', '2020-07-01', '2020-07-02', '2020-07-03', '2020-07-06', '2020-07-07', '2020-07-08', '2020-07-09', '2020-07-12', '2020-07-13', '2020-07-14', '2020-07-15', '2020-07-16', '2020-07-19', '2020-07-20', '2020-07-21', '2020-07-22', '2020-07-23', '2020-07-26', '2020-07-27', '2020-07-28', '2020-07-29', '2020-07-30', '2020-08-02', '2020-08-03', '2020-08-04', '2020-08-05', '2020-08-06', '2020-08-09', '2020-08-10', '2020-08-11', '2020-08-12', '2020-08-13', '2020-08-14', '2020-08-17', '2020-08-18', '2020-08-19', '2020-08-20', '2020-08-23', '2020-08-24', '2020-08-25', '2020-08-26', '2020-08-27', '2020-08-30', '2020-08-31', '2020-09-01', '2020-09-02', '2020-09-03', '2020-09-04', '2020-09-07', '2020-09-08', '2020-09-09', '2020-09-10', '2020-09-11', '2020-09-14', '2020-09-15', '2020-09-16', '2020-09-17', '2020-09-18', '2020-09-21', '2020-09-22', '2020-09-23', '2020-09-24', '2020-09-25', '2020-09-28', '2020-09-29', '2020-09-30']
```



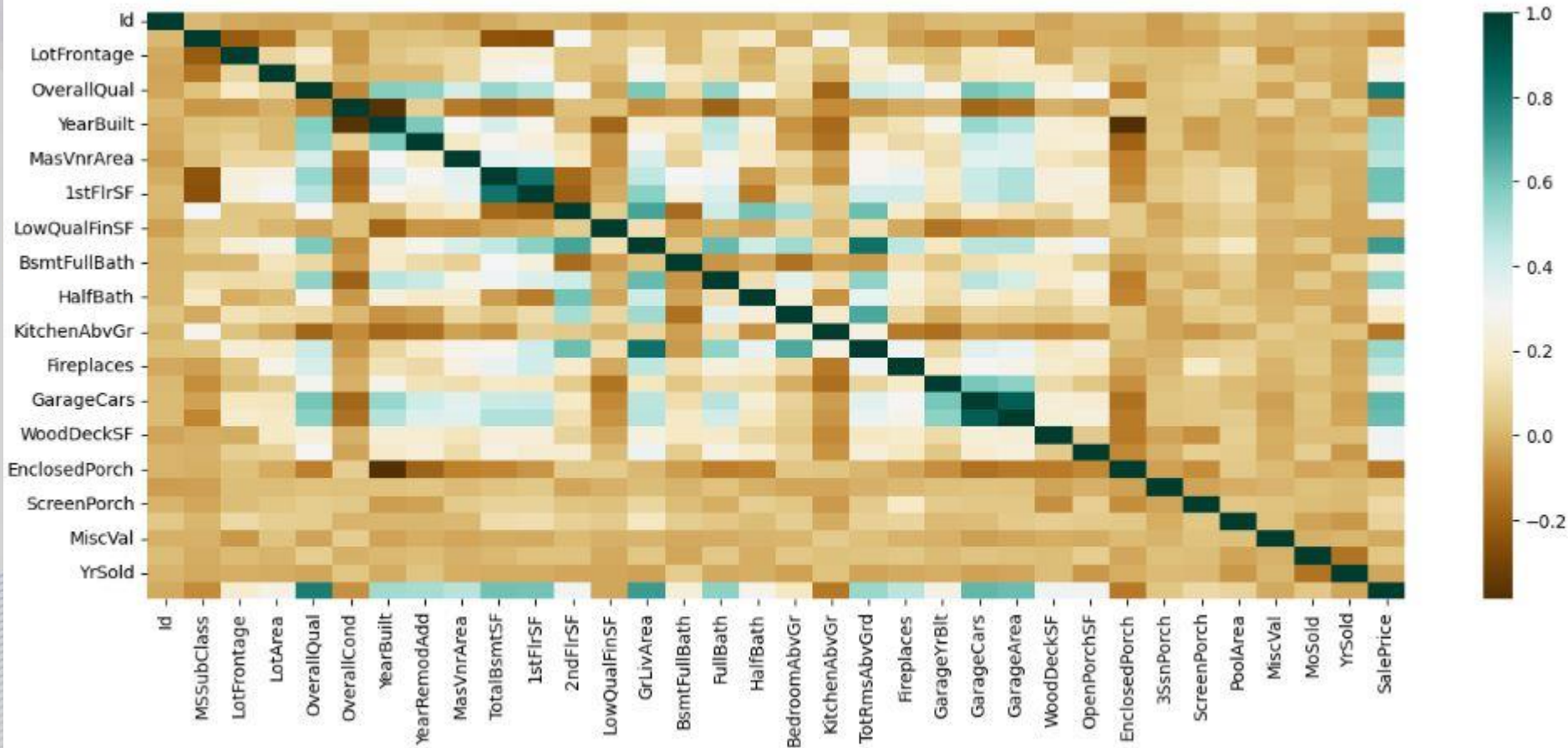
# NYC flights EDA



# NYC flights EDA



# London houses prices





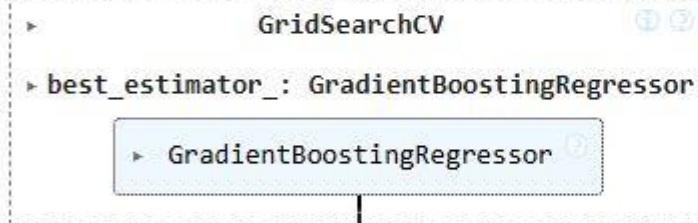
# London houses prices

```
param_grid_gbr={  
    "learning_rate":[0.05, 0.1, 0.2],  
    "n_estimators" : [40, 100, 250],  
    "max_depth" : [3, 5, 8]  
}
```

```
gbr2 = GradientBoostingRegressor(criterion='squared_error', random_state=42)
```

```
CV_gbr = GridSearchCV(estimator=gbr2, param_grid=param_grid_gbr, cv=5)
```

```
CV_gbr.fit(X_train_best_num, y_train)
```



```
CV_gbr.best_params_
```

```
{'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 40}
```

```
CV_gbr.best_score_
```

```
0.8250241125102503
```

```
lasso_pred = lasso.predict(X_valid_num)
```

```
print(root_mean_squared_error(lasso_pred, y_valid))
```

```
39254.88822704823
```

```
importance_df = pd.DataFrame({  
    'Feature': X_train_num.columns,  
    'Importance': np.abs(lasso.coef_)  
}).sort_values(by='Importance', ascending=False)
```

```
importance_df.head(10)
```

	Feature	Importance
18	KitchenAbvGr	17878.939810
22	GarageCars	16658.577801
4	OverallQual	16086.699326
14	BsmtFullBath	14969.486666
17	BedroomAbvGr	11215.729890
19	TotRmsAbvGrd	5526.306931
5	OverallCond	5370.636825