Topic Modeling and Machine Learning on Hotel Review Data

Yanhua Hou

Mentor: Alex Rutherford

Data Science Intensive Capstone Project

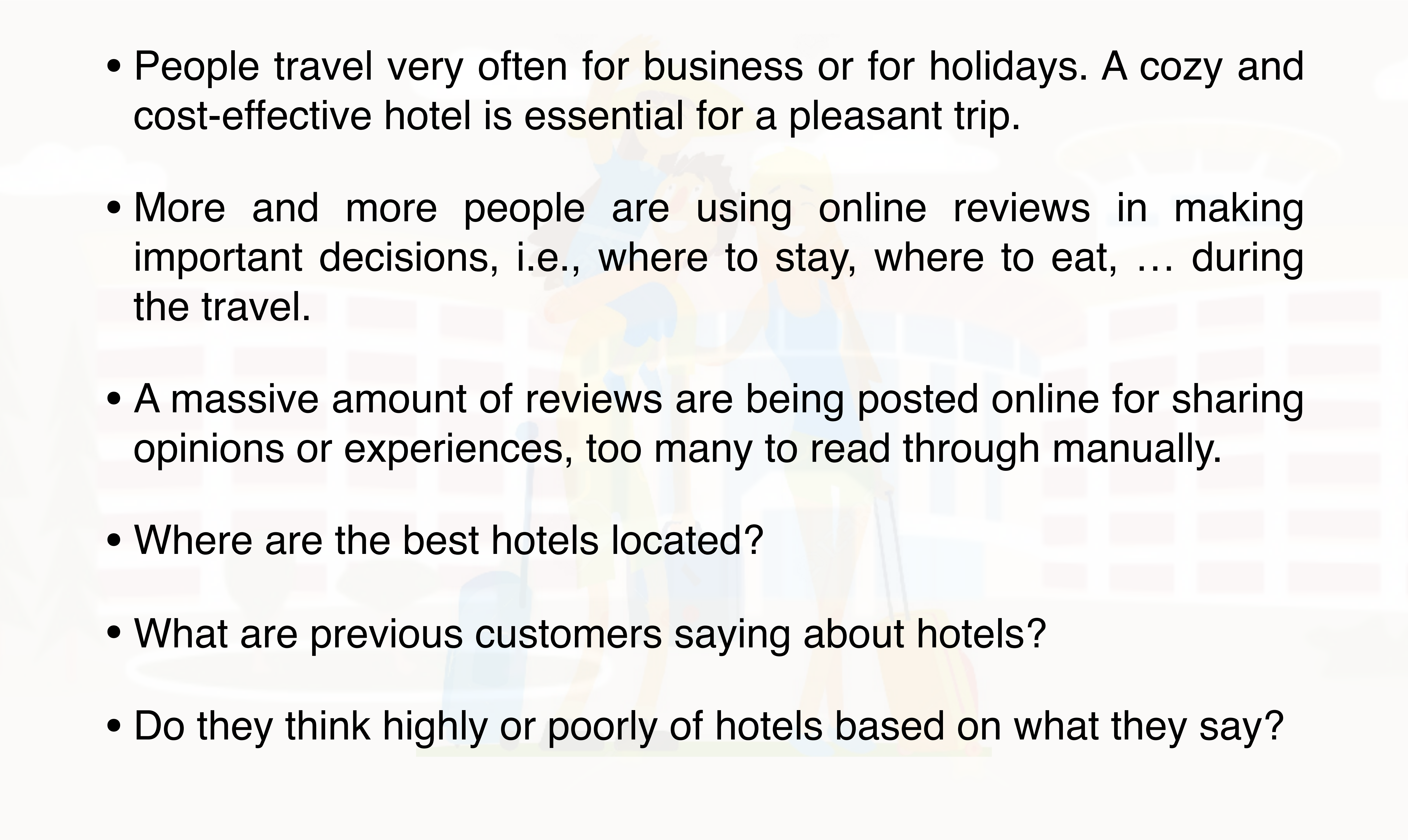
Github: https://github.com/phyhouhou

Springboard

Outlines

- Problems and Clients
- Data Information and Data Cleaning
- Exploratory Data Analysis
- Machine Learning Models for Predictions
- Conclusions and Future Work

Problems



- People travel very often for business or for holidays. A cozy and cost-effective hotel is essential for a pleasant trip.

- More and more people are using online reviews in making important decisions, i.e., where to stay, where to eat, … during the travel.

- A massive amount of reviews are being posted online for sharing opinions or experiences, too many to read through manually.

- Where are the best hotels located?

- What are previous customers saying about hotels?

- Do they think highly or poorly of hotels based on what they say?

Clients

Review sites, travel intelligences, hotels, mobile apps…

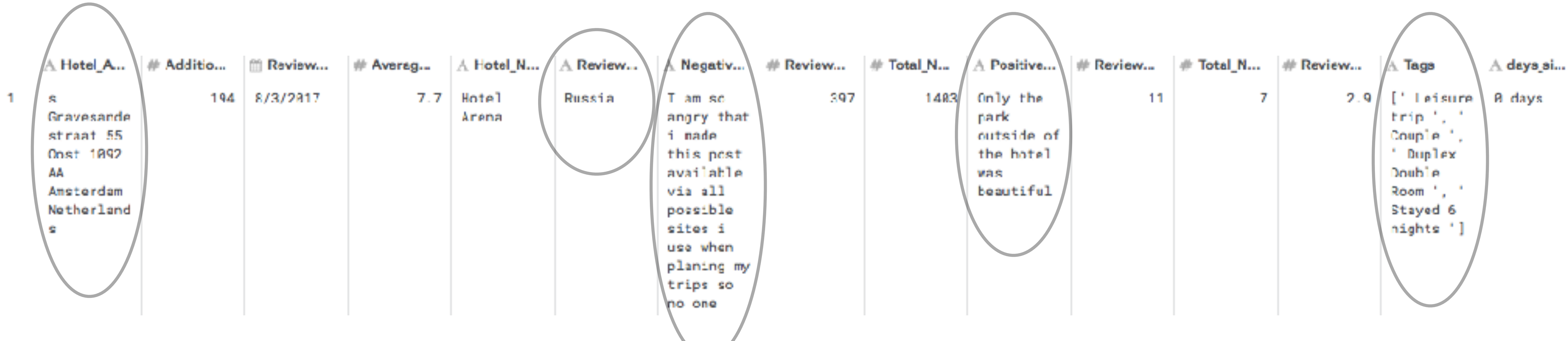
Data Information

- Data is from kaggle. It’s about 515k+ reviews data and scoring for 1493 luxury hotels across Europe from 2015-08-09 to 2017-08-03 with 17 features, among which 8 'object', 5 'int64' and 4 'float64'.

- Hotel features : ‘*Hotel_Name’*, ‘*Hotel_Address*’, ‘*Average_Score*’, ‘*lat*’, ‘*lng*’, ’*Total_Number_of_Reviews*’, etc.

- Review features: ‘*Review_Date*’, ‘*Negative_Review*’, ‘*Positive_Review’*, ‘Review_Score’, ‘Tags’, etc.

- Reviewer features: *Reviewer_Nationality*’, ‘*Total_Number_of_Reviews_Reviewer_Has_Given*’, etc.

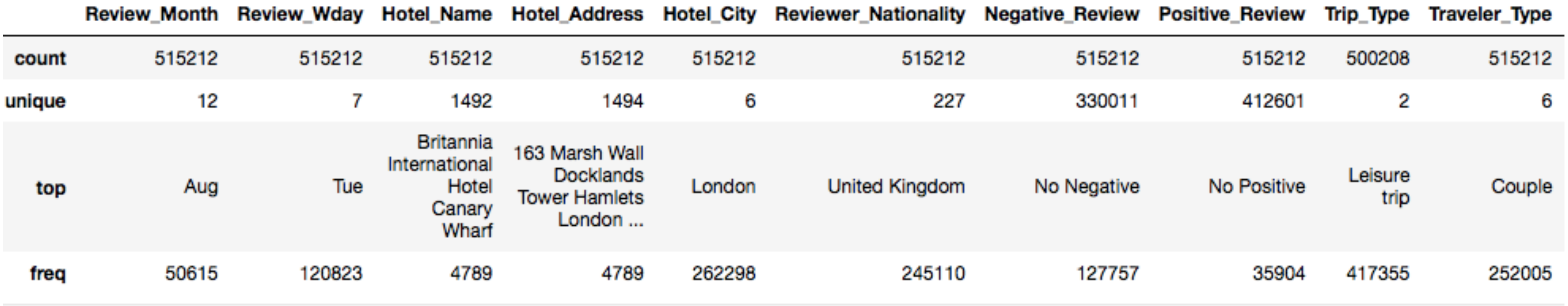


	Hotel_A...	Additio...	Review...	Averag...	Hotel_N...	Review...	Negativ...	Review...	Total_N...	Positive...	Review...	Total_N...	Review...	Tags	days_si...
1	s Gravesande straat 55 Oost 1092 AA Amsterdam Netherlands	194	8/3/2017	7.7	Hotel Arena	Russia	I am so angry that i made this post available via all possible sites i use when planing my trips so no one	397	1403	Only the park outside of the hotel was beautiful	11	7	2.9	[' Leisure trip ', ' Couple ', ' Duplex Double Room ', ' Stayed 6 nights ']	0 days

Data Wrangling

- Missing values and duplicates;
- Extract cities of hotels from its address, i.e., ‘Hotel_Address’, days from 'days_since_review';
- Add month and day that reviews are posted;
- Add 'Pos_Rev_WCRatio' and 'Neg_Rev_WCRatio';
- Extract features from ‘Tag’, i.e., ‘Trip_Type’, ‘Traveler_Type’, ‘Num_Nights’;
- Preprocess review texts, i.e., remove non-letters characters, stopwords, whitespaces, word tokenization and lemmatization.

A summary statistics of ‘object’ features

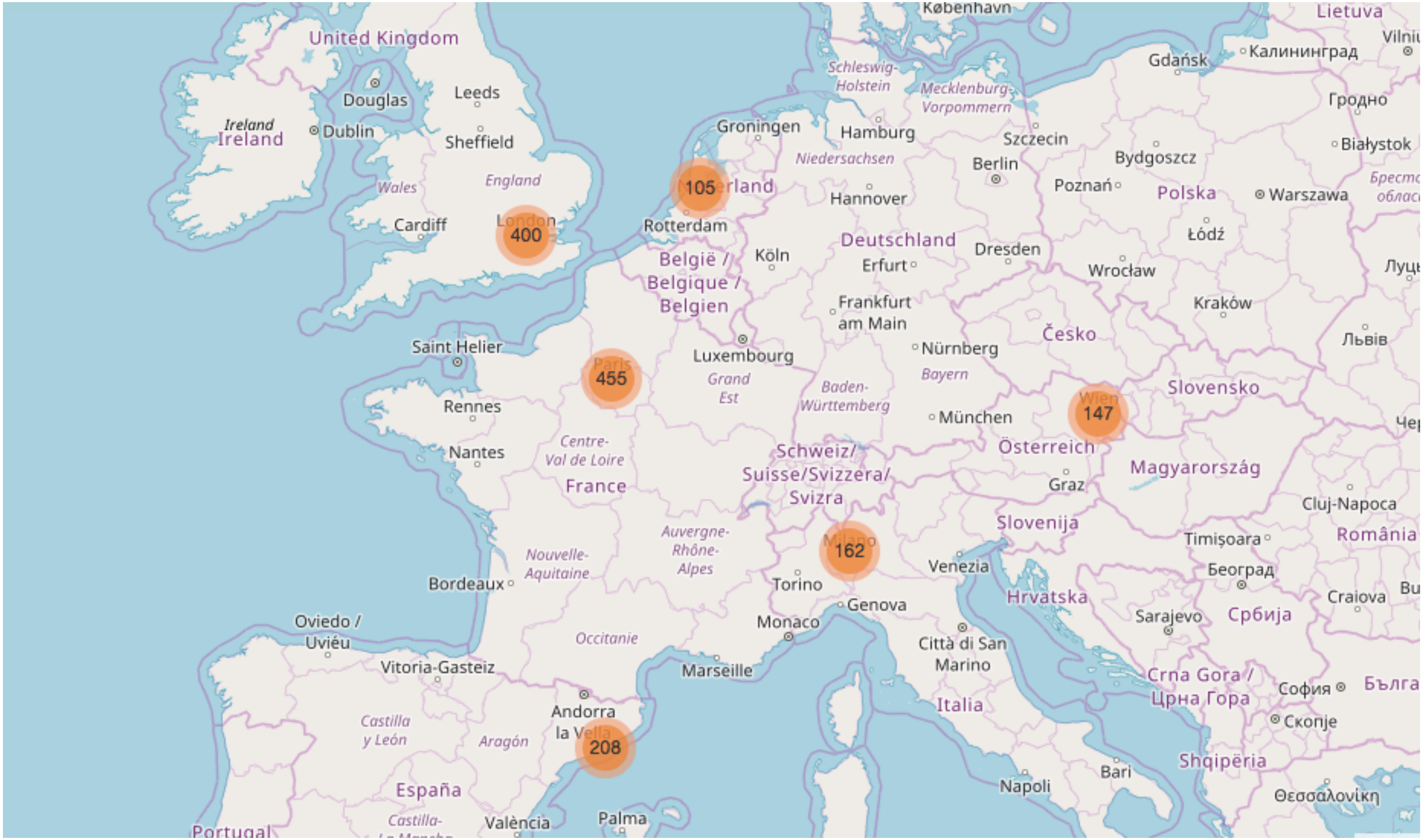


	Review_Month	Review_Wday	Hotel_Name	Hotel_Address	Hotel_City	Reviewer_Nationality	Negative_Review	Positive_Review	Trip_Type	Traveler_Type
count	515212	515212	515212	515212	515212	515212	515212	515212	500208	515212
unique	12	7	1492	1494	6	227	330011	412601	2	6
top	Aug	Tue	Britannia International Hotel Canary Wharf	163 Marsh Wall Docklands Tower Hamlets London ...	London	United Kingdom	No Negative	No Positive	Leisure trip	Couple
freq	50615	120823	4789	4789	262298	245110	127757	35904	417355	252005

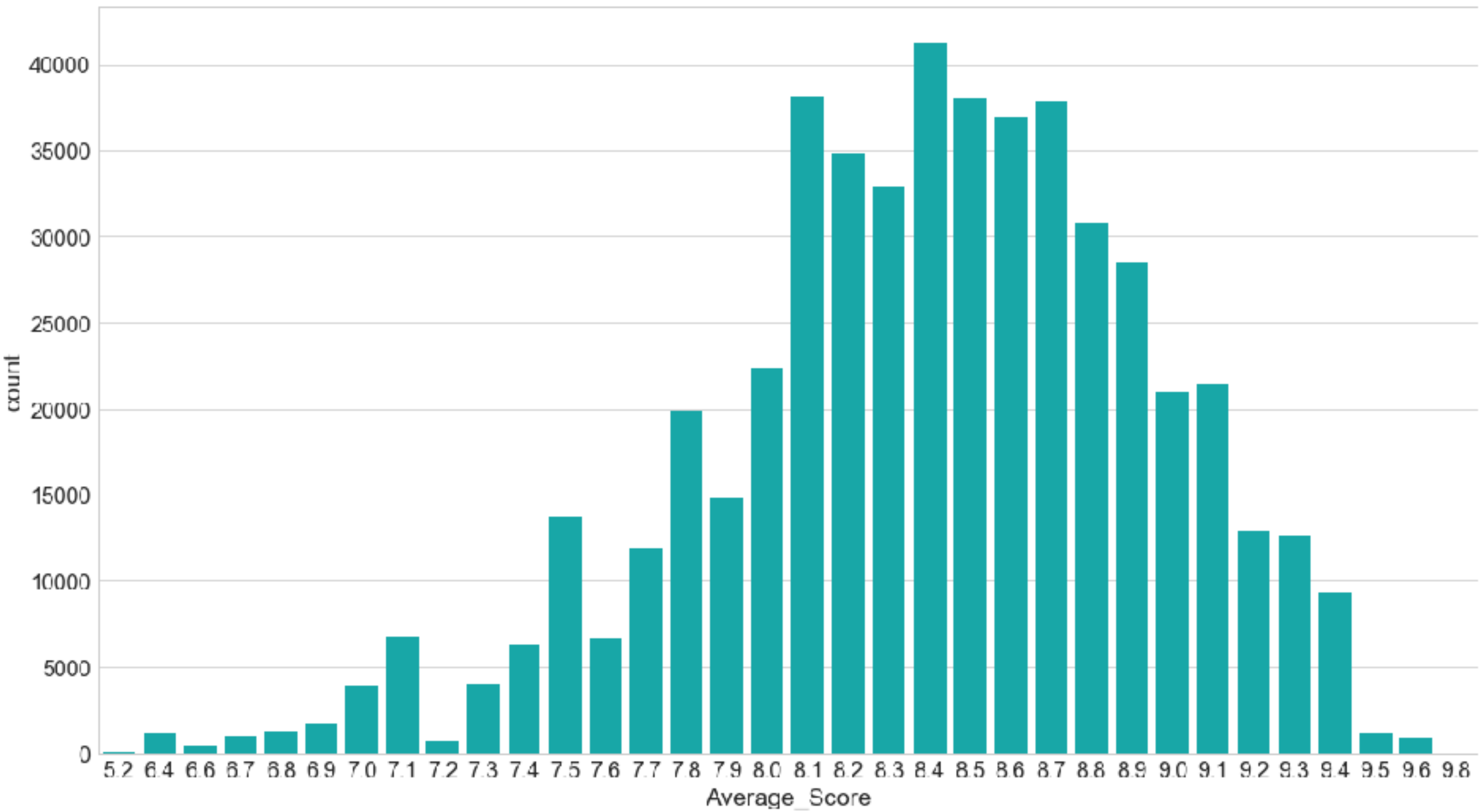
Data Analysis

- Hotels
- Reviewers
- Reviews

Visualization of Hotels

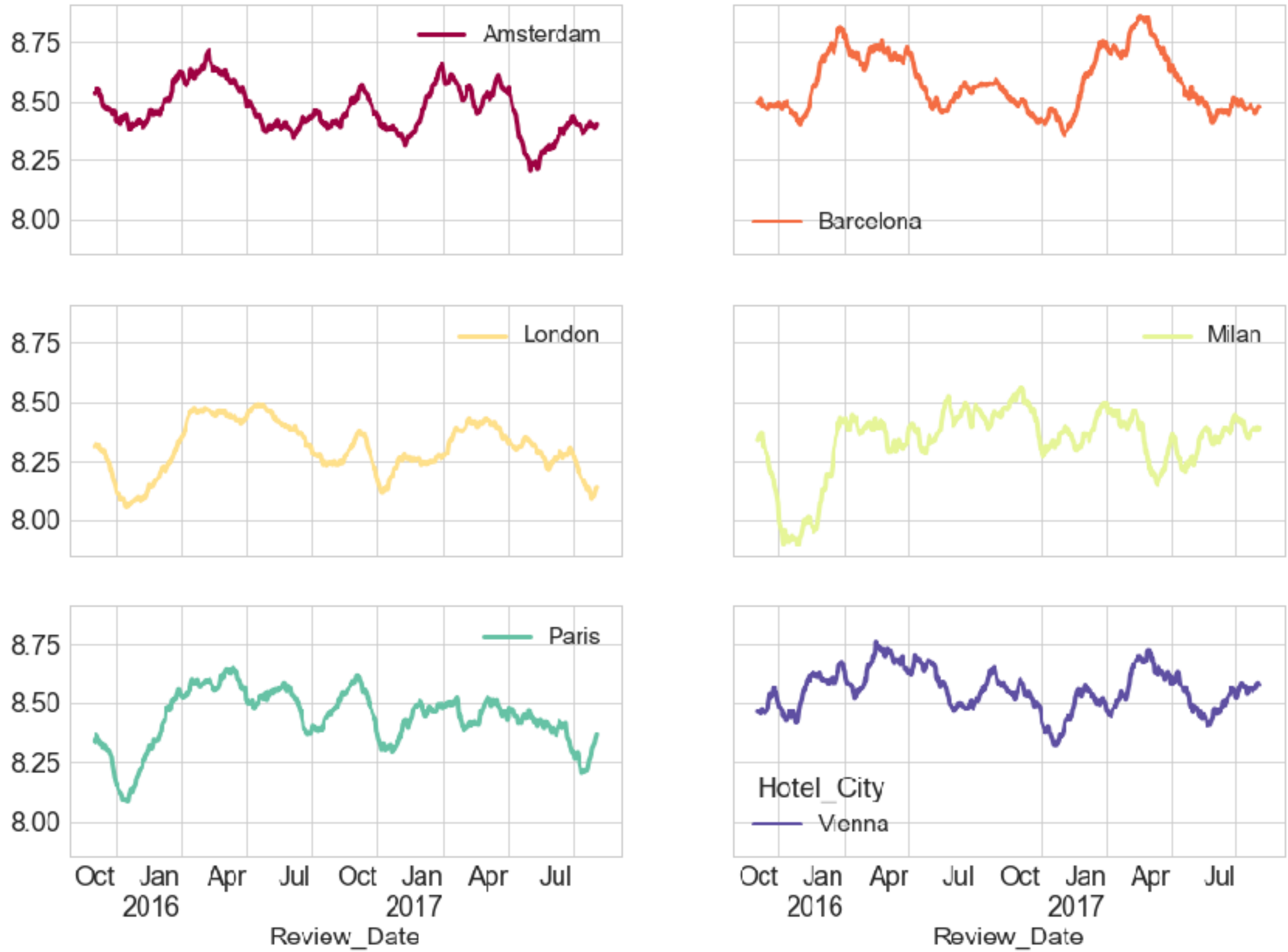


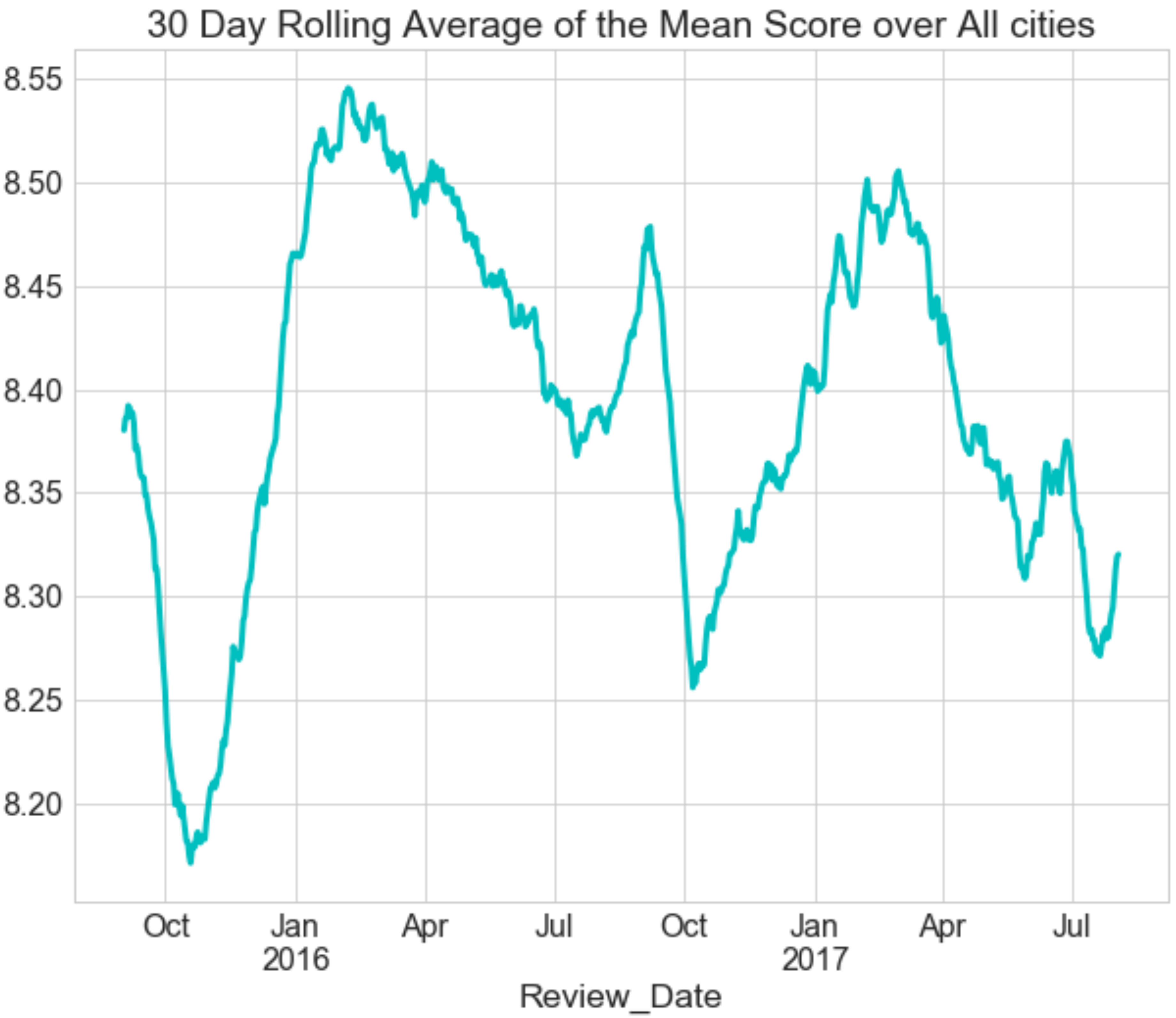
Average Score of Hotels



All average scores are above 5.0, most are above 8.0

Time Series Plot of Reviewer_Score for Hotels located in 6 Cities:
30 Days Rolling Mean of the Average Reviewer Score

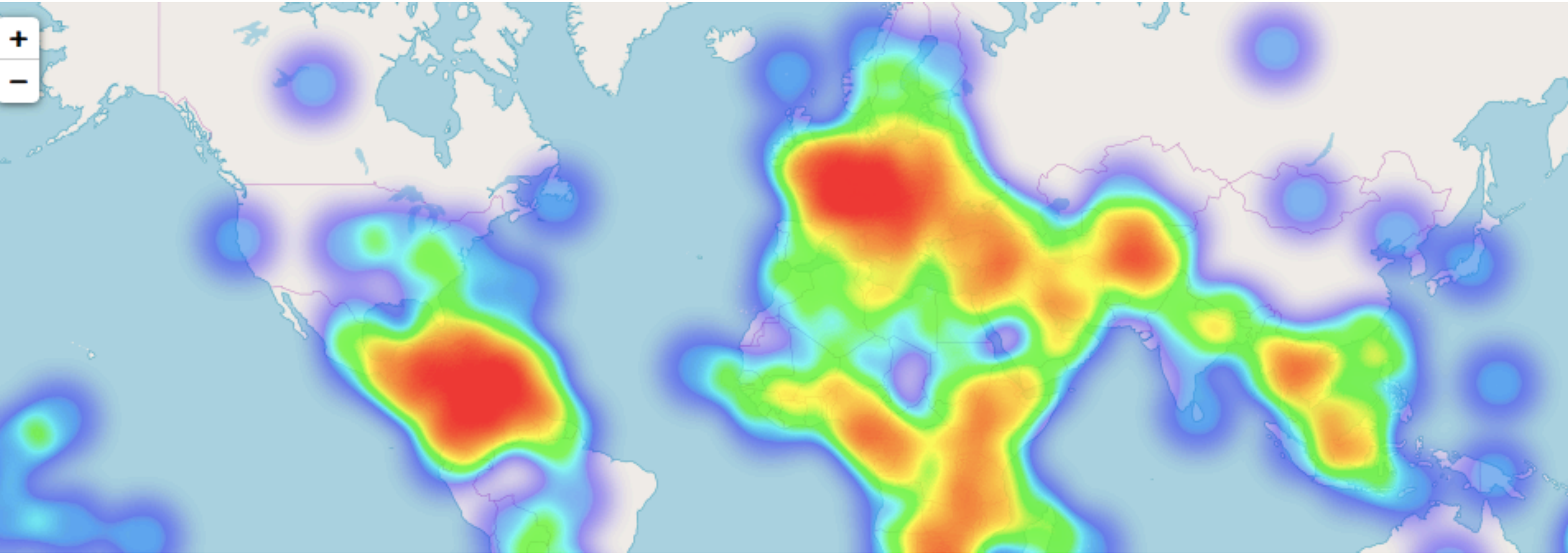




The average reviewer score of hotels is higher in January and low in October.

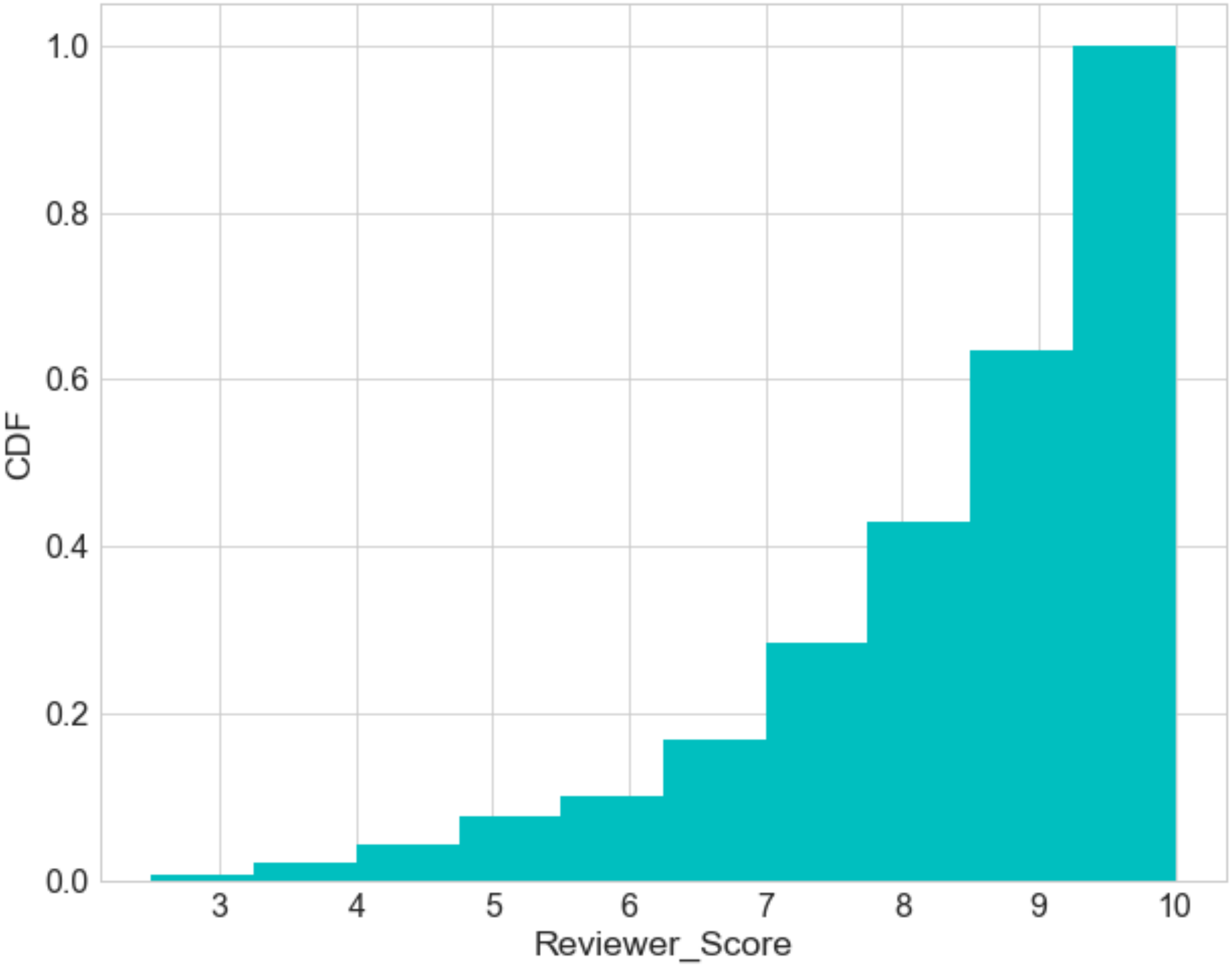
Visualization of Reviewers

Where are reviewers from?



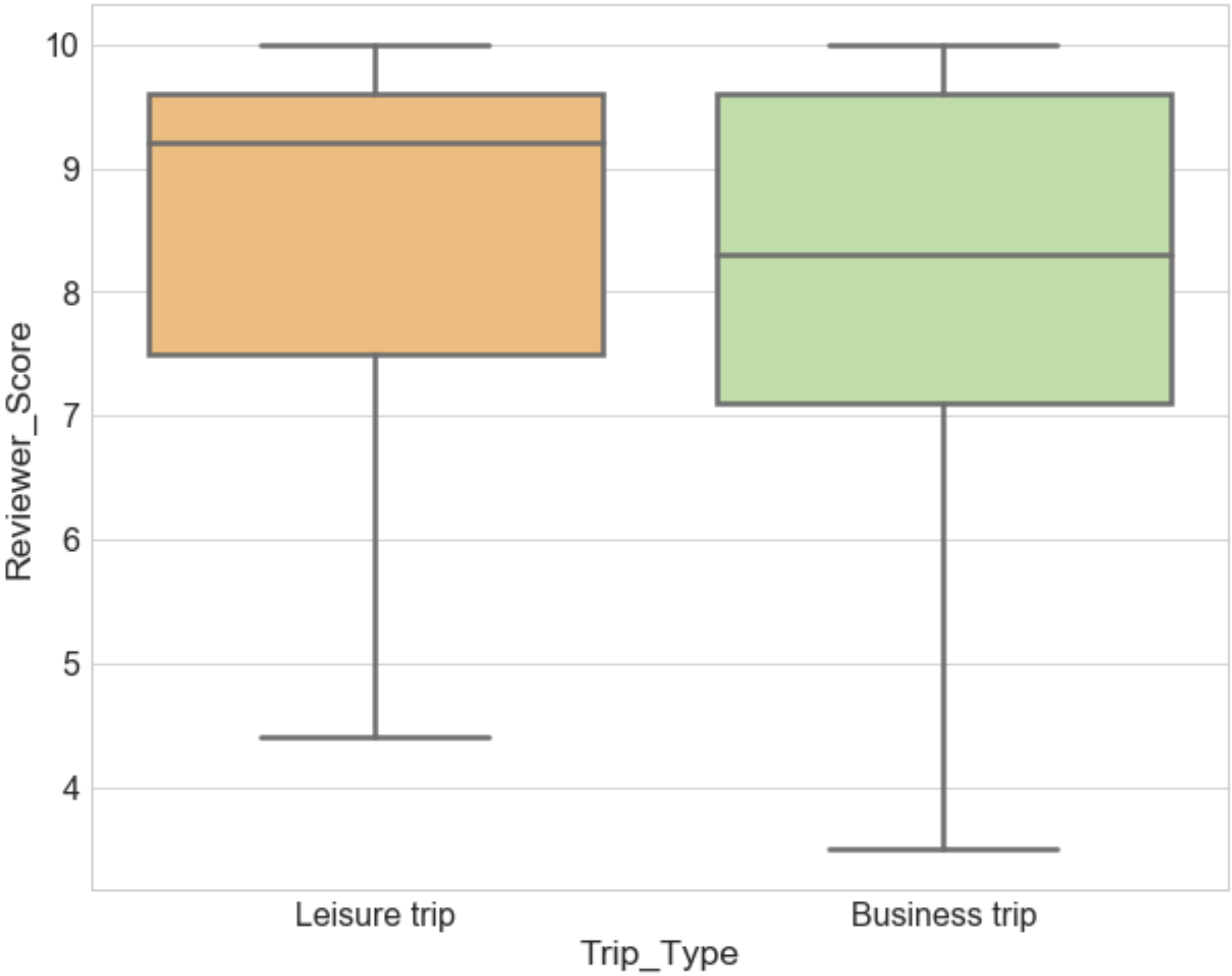
Redder indicates more reviewers. While hotels are located in Europe, many reviewers are from America and Europe.

CDF of Reviewer Score



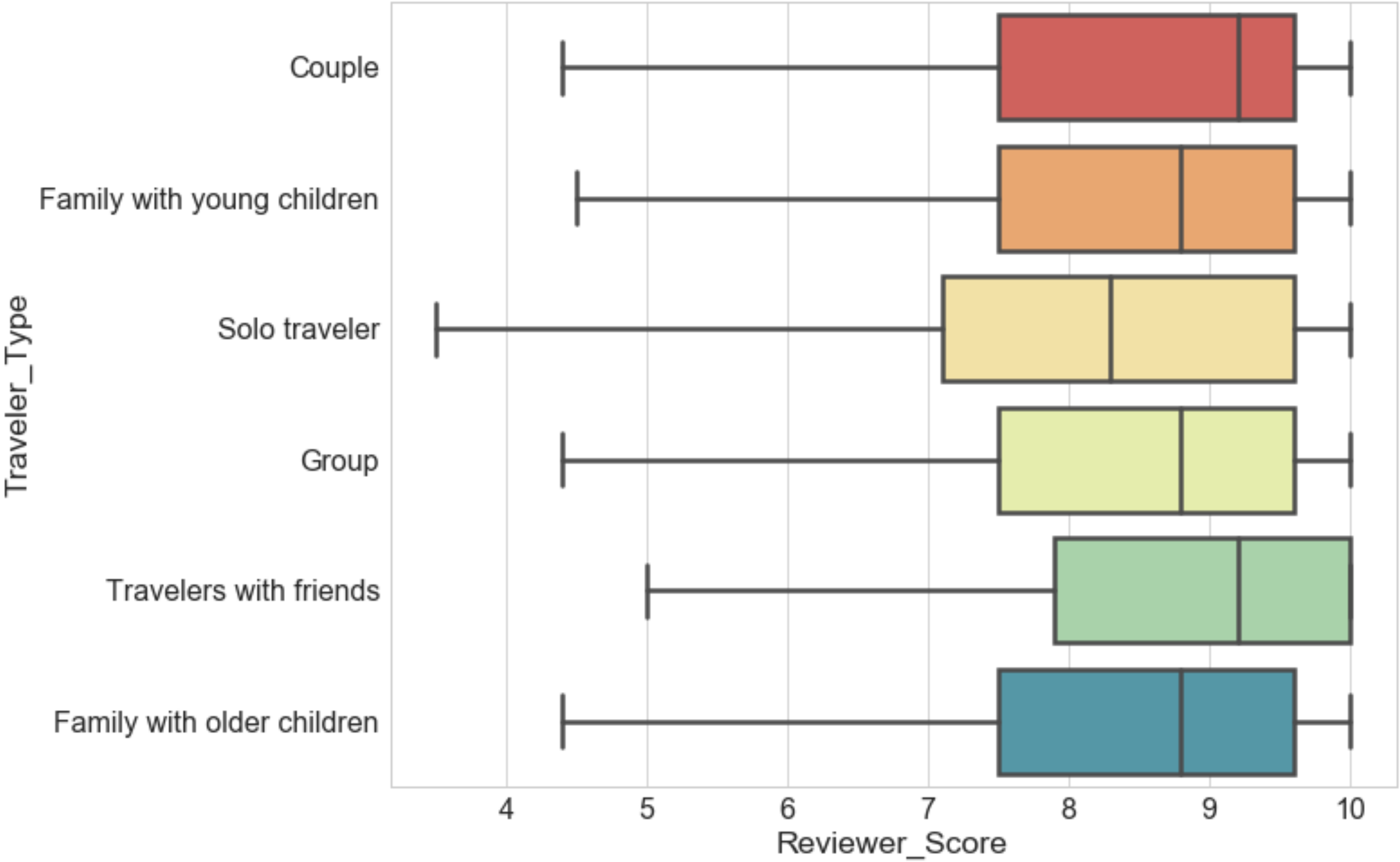
A majority of scores are high except a minority of very low scores.

How does type of trip affect scores?



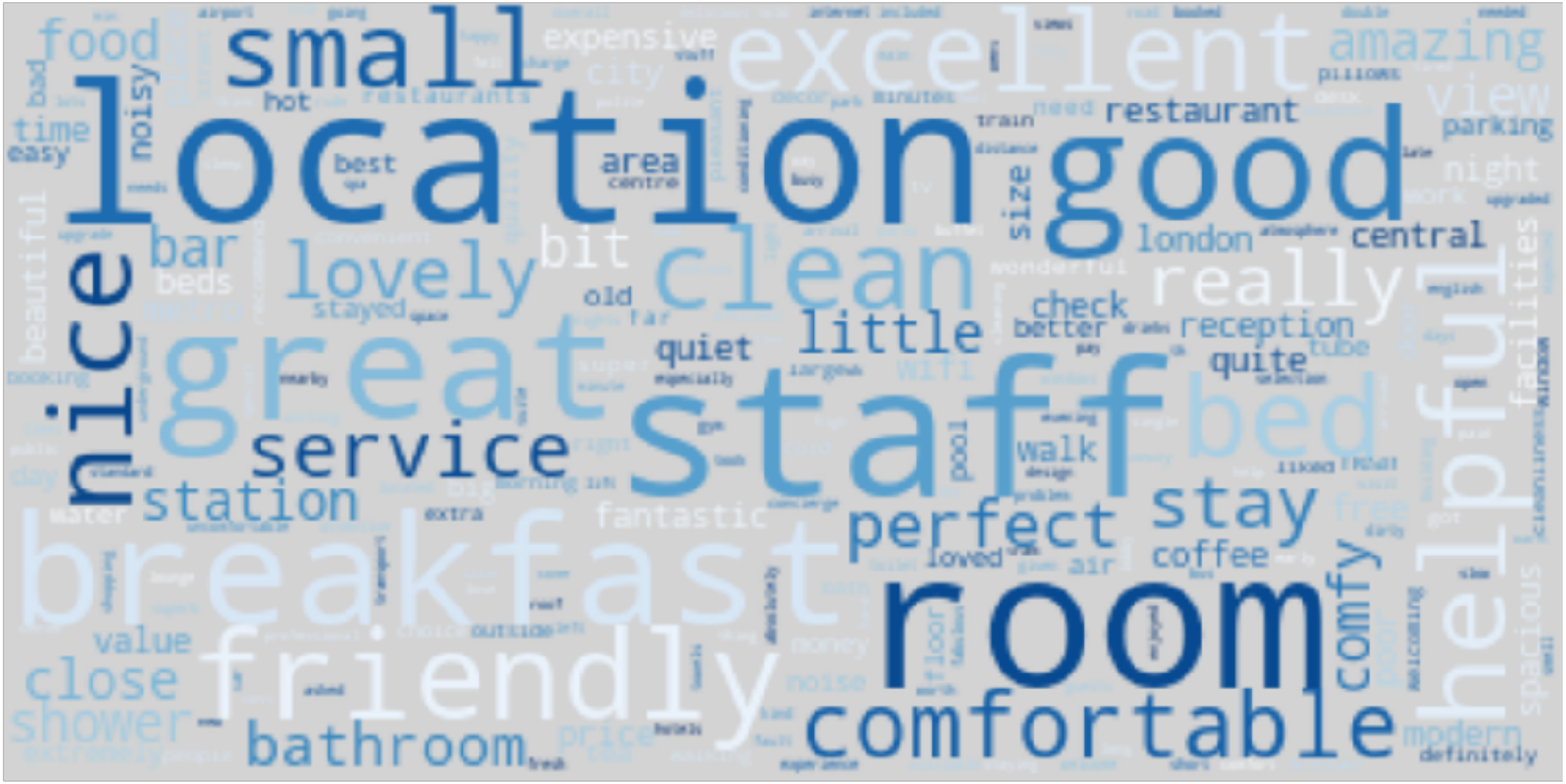
Reviewers on a leisure trip tend to rate higher than those on a business trip.

How does type of traveler affect scores?



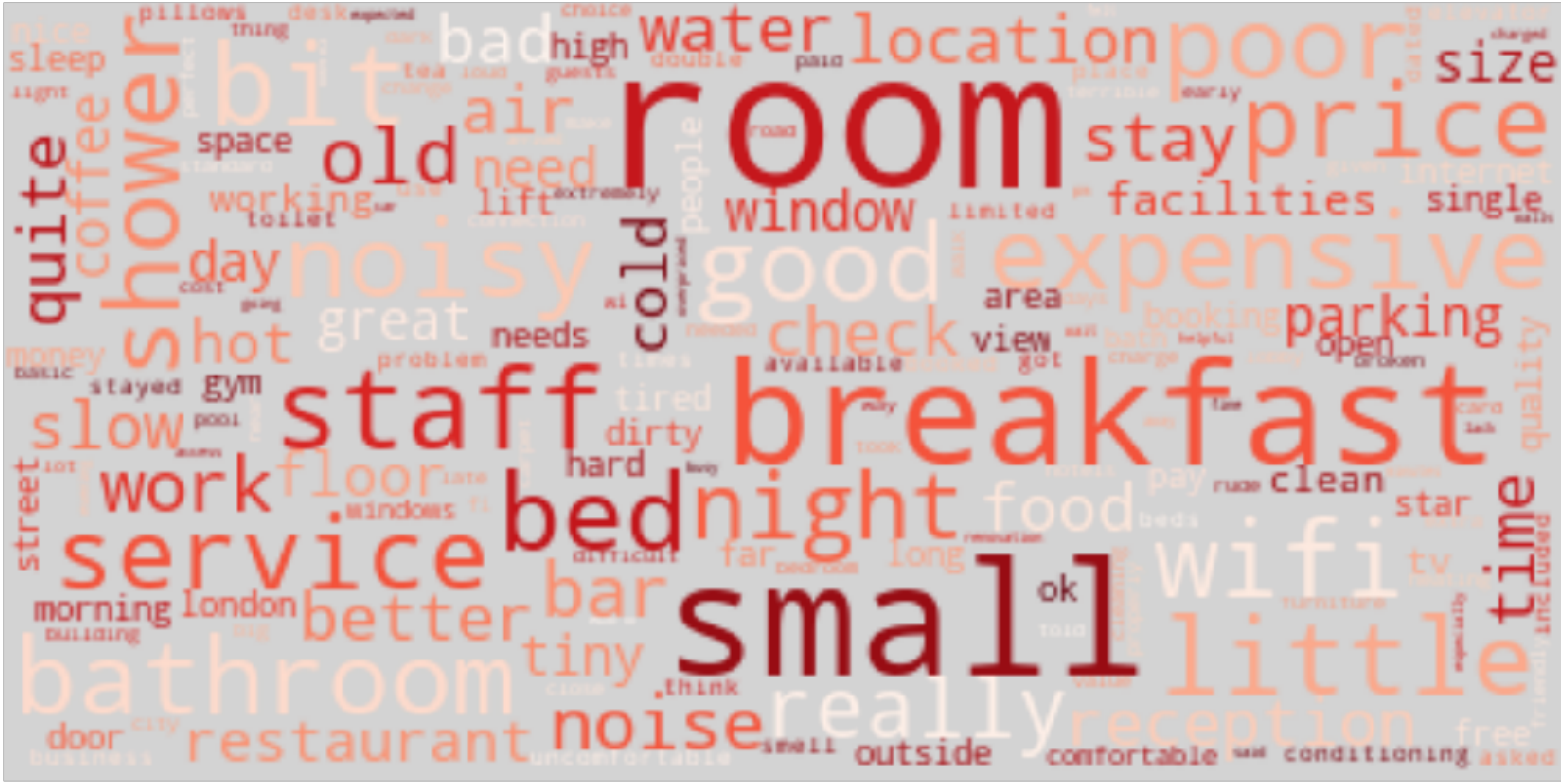
The type of travelers is a relevant factor affecting scores. For instance, ‘solo traveler’ tends to rate lower than ‘couples’.

Word Cloud of Reviews



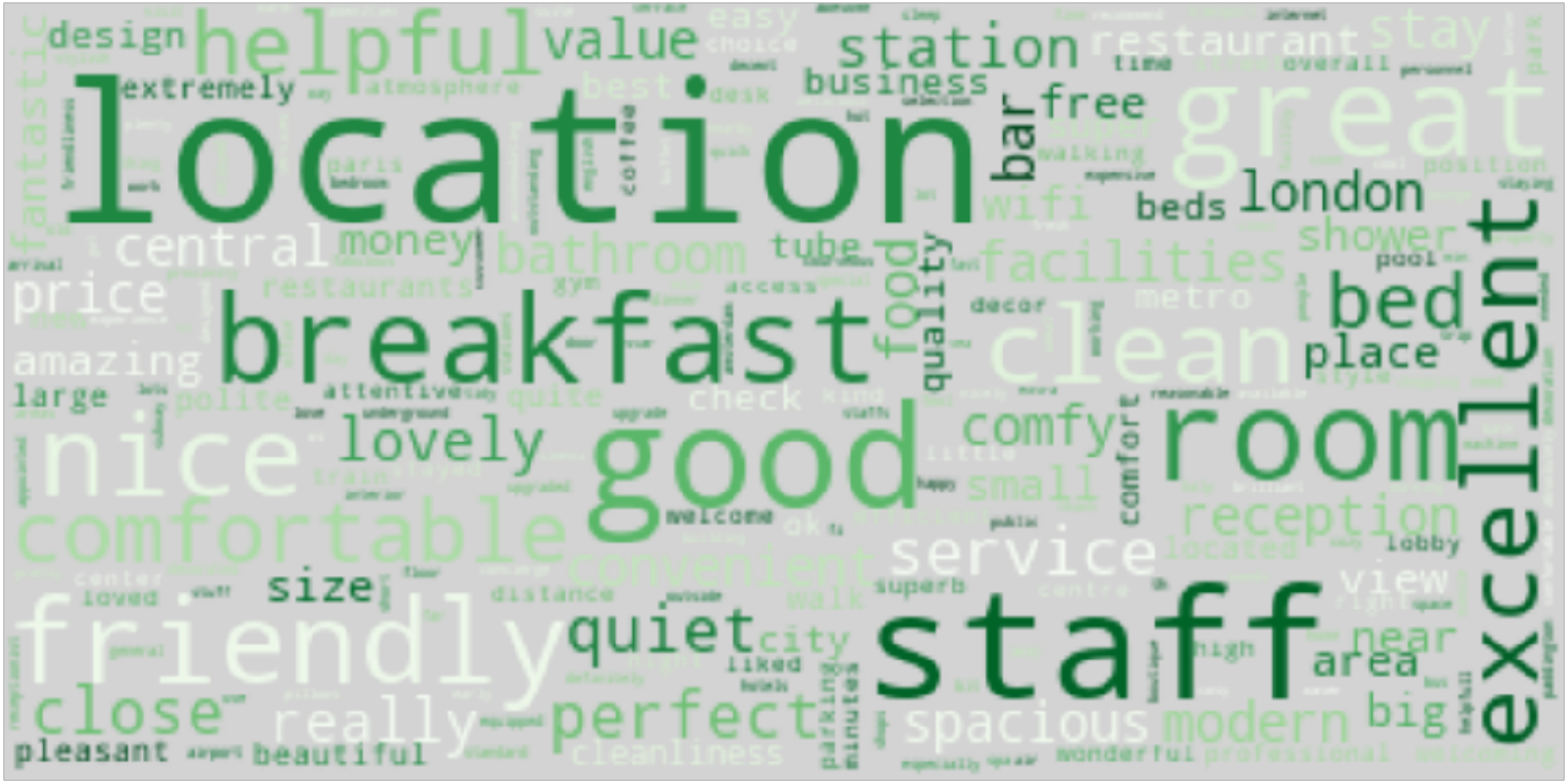
Most frequent words indicates content on location, cleanliness, staff, facilities, food, comfort, value for money (price)…

Negative Reviews on Business Trip



Complaining on room size, breakfast, staff, service, noise, old, bad,…

Positive Reviews on Business Trip

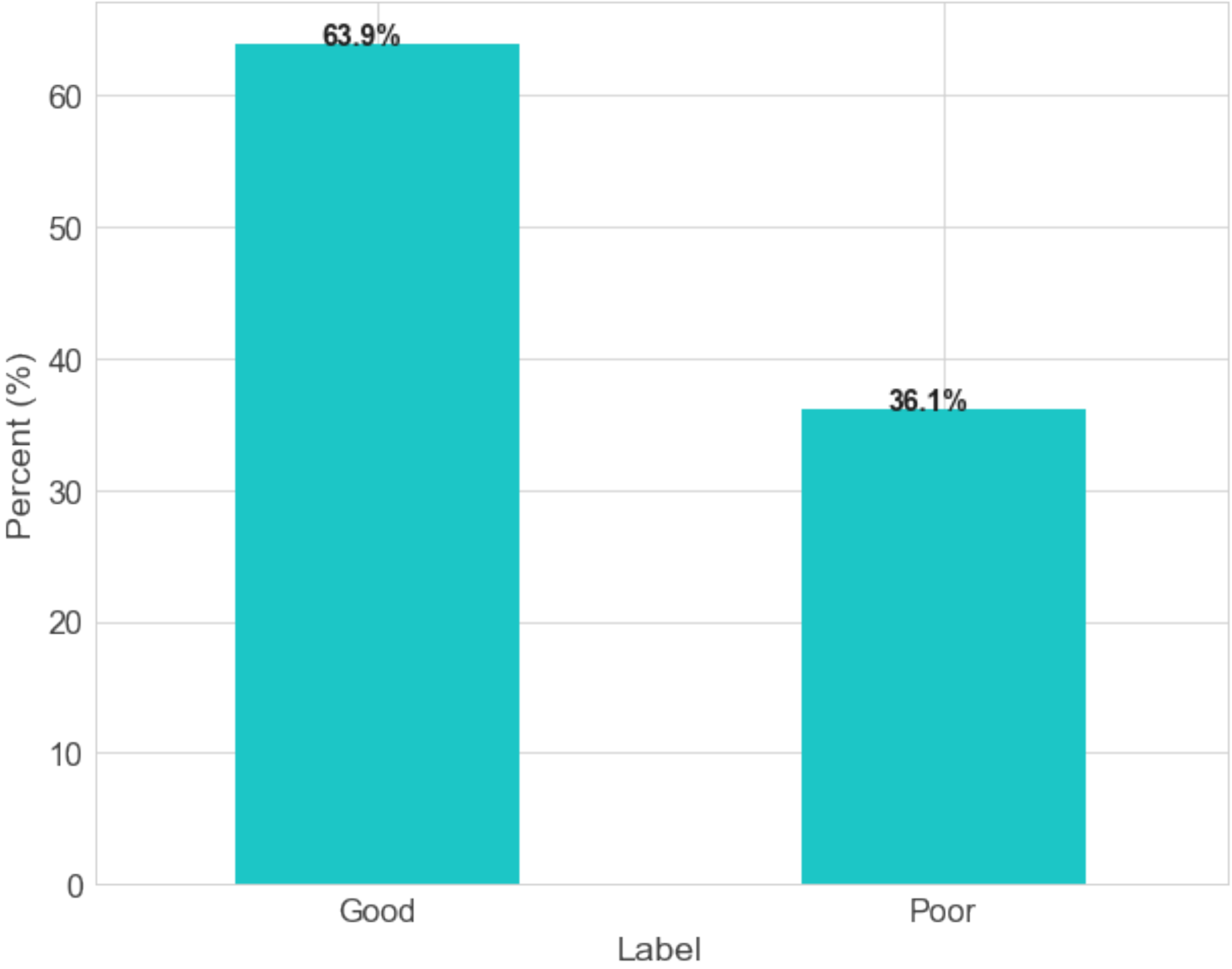


Positive aspects on location, breakfast, staff, room…words such as excellent, good, lovely, nice, friendly occur very frequently

Classifier

Can we build a model to tell if reviewer thinks highly or poorly of hotels based on what they posted?

Classes

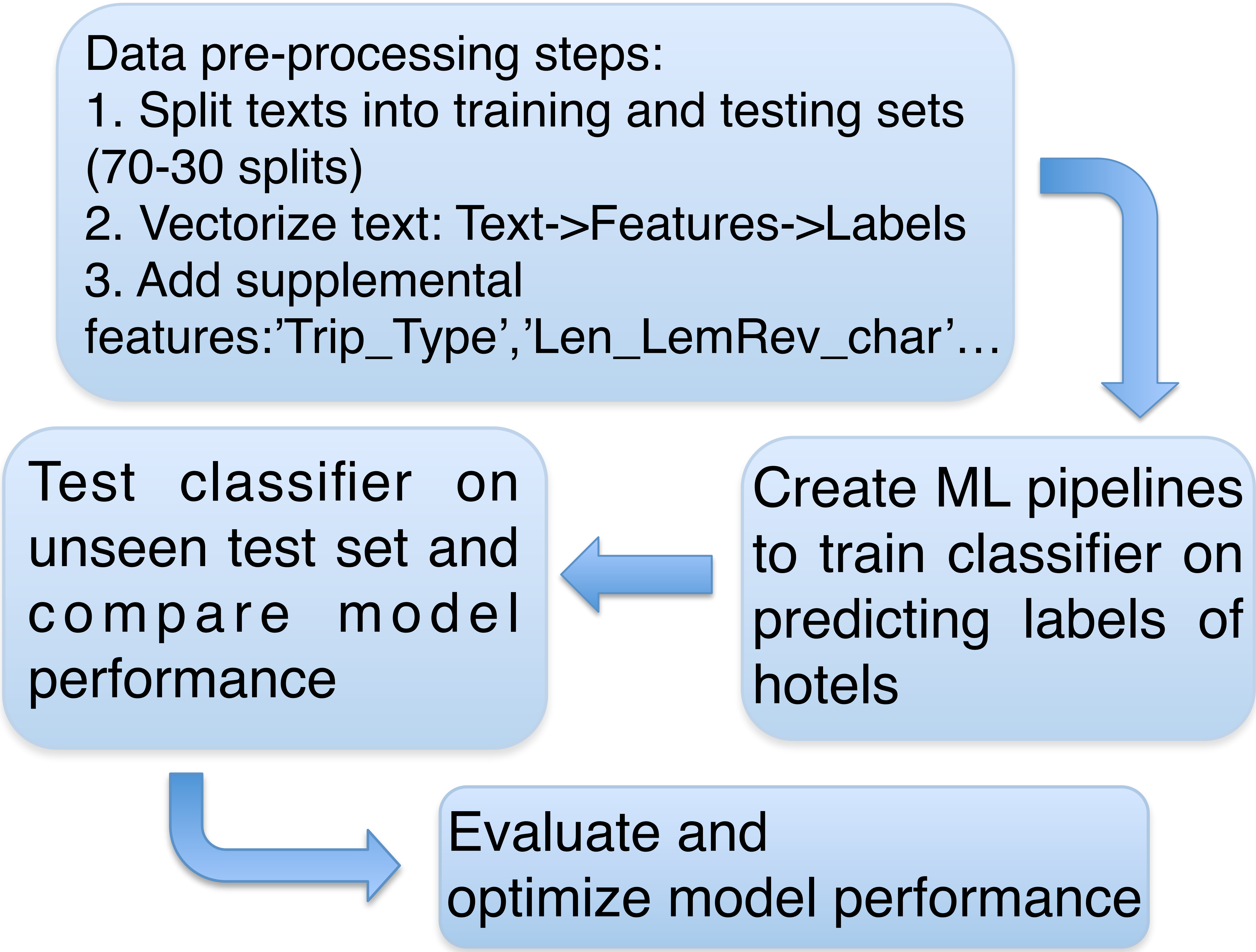


Add label: ‘poor’ (reviewer_score<8), ‘good’ otherwise

Machine Learning Model for Predictions

- Feature engineering / feature selection
- Model comparison/evaluation
- Model optimization

Classification Steps



Data pre-processing steps:

1. Split texts into training and testing sets (70-30 splits)
2. Vectorize text: Text->Features->Labels
3. Add supplemental features:'Trip_Type','Len_LemRev_char'…

Create ML pipelines to train classifier on predicting labels of hotels

Test classifier on unseen test set and compare model performance

Evaluate and optimize model performance

Classifier Ingredients: Vectorizer

- Classifier interacts with features, not the text itself.

- Vectorizer or feature extractor transforms a text into quantifiable information about the text.

- Common word feature extractors:
  - Bag-of-Words (word counts)
  - TF-IDF weighting
  - LDA model topics

Bag-of-Words (BOW) Representation

Bag of words is a kind of feature extraction where:

- The set of features is the set of words in the text.
- A single text is represented by how many of each word appears in it.

Simple example

Original text is
Hop on pop
Hop off pop
Hop Hop hop

Words for each feature:
['hop', 'off', 'on', 'pop']

Transformed text vector is
[1 0 1 1]
[1 1 0 1]
[3 0 0 0]

	hop	off	on	pop
Hop on pop	1	0	1	1
Hop off pop	1	1	0	1
Hop Hop hop	3	0	0	0

Bag-of-Words (BOW) Representation

Use CountVectorizer to create document-word matrix:

```
CountVectorizer(   min_df=10,   #words have occurred at least 10 times

                  stop_words=‘english',

                  lowercase=True,

                  token_pattern='[a-zA-Z]{3,}', # char length at least 3

                )
```

Shape of Sparse Matrix: (429464, 10670)
Amount of Non-Zero occurrence: 6941938
sparsity: 0.15%

TF-IDF Weighting Representation

- Term-Frequency * Inverse Document Frequency (weighted by the inverse of its popularity in all documents).
- TF-IDF is essentially a measure of term importance, and of how discriminative a word is in a corpus.
- TF-IDF weighted features can be used as inputs to any classifier.

Simple example

Original text is
Hop on pop
Hop off pop
Hop Hop hop

IDF: ‘hop': 1.0, 'off': 1.69, 'on': 1.69, 'pop': 1.29

Transformed text vector is

```
[1 0 1 1]        [1  0.     1.69   1.29]
[1 1 0 1]   →    [1  1.69   0.     1.29]
[3 0 0 0]        [3  0.     0.     0.  ]
```

LDA Topics Representation

- Group words into topics applicable for prediction
- Each document is represented by a vector of its topic portions.
- Num_topics needs to be determined empirically
- Provide dimension reduction(10670 features->5, …100)
- Implemented via Scikit-learn library and Gensim.

Display top 20 keywords in topics

Topic 0: bed room bathroom shower comfortable small comfy clean nice water good location pillow big bath size great coffee double large

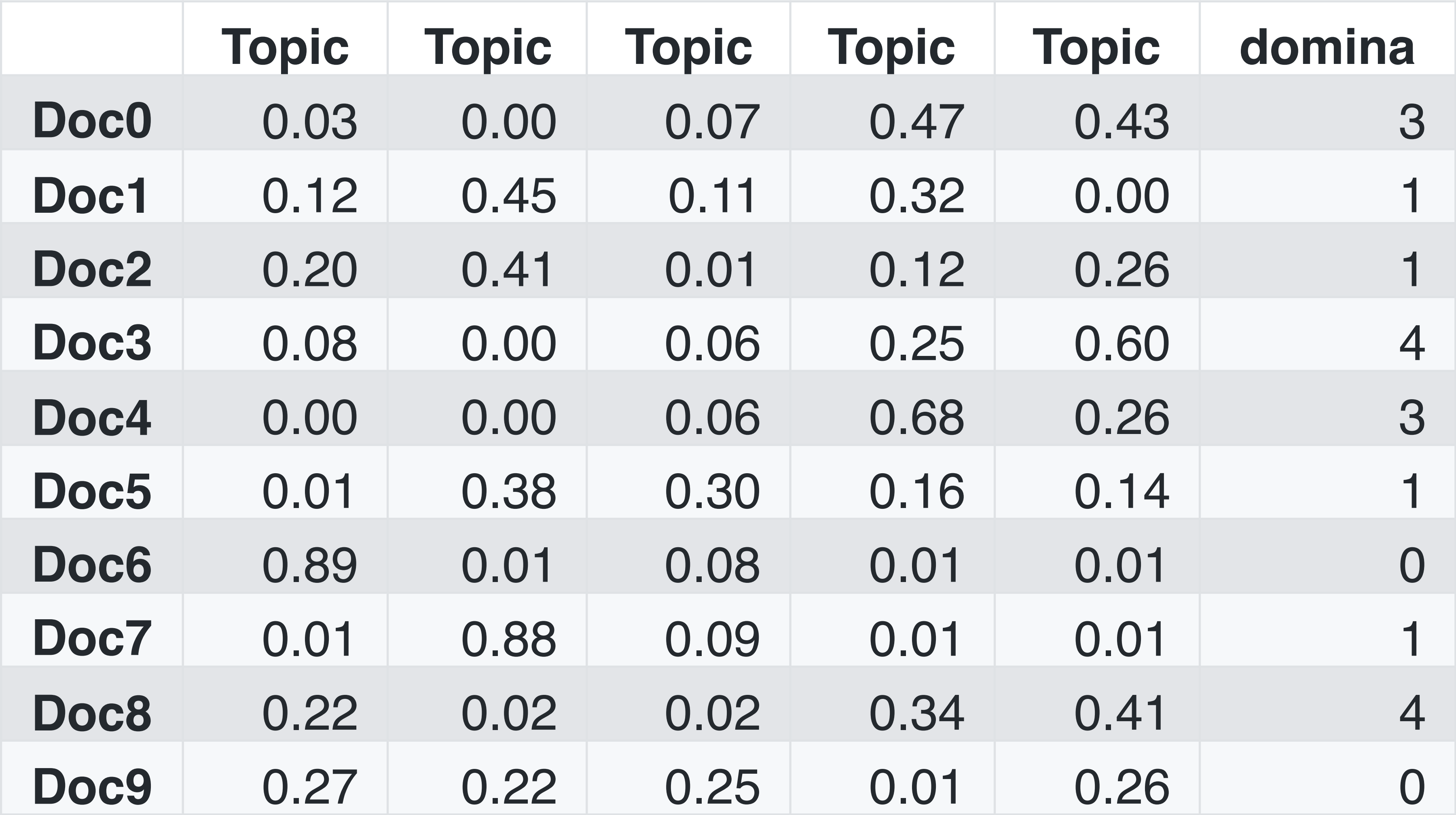
Topic 1: staff breakfast room location friendly helpful good great excellent clean nice bar food comfortable service lovely really restaurant facility stay

Topic 2: location close walk station good great city metro nice restaurant pool minute easy room area clean train centre value parking

Topic 3: stay staff check room time day book make pay ask service reception night say charge come tell help leave extra

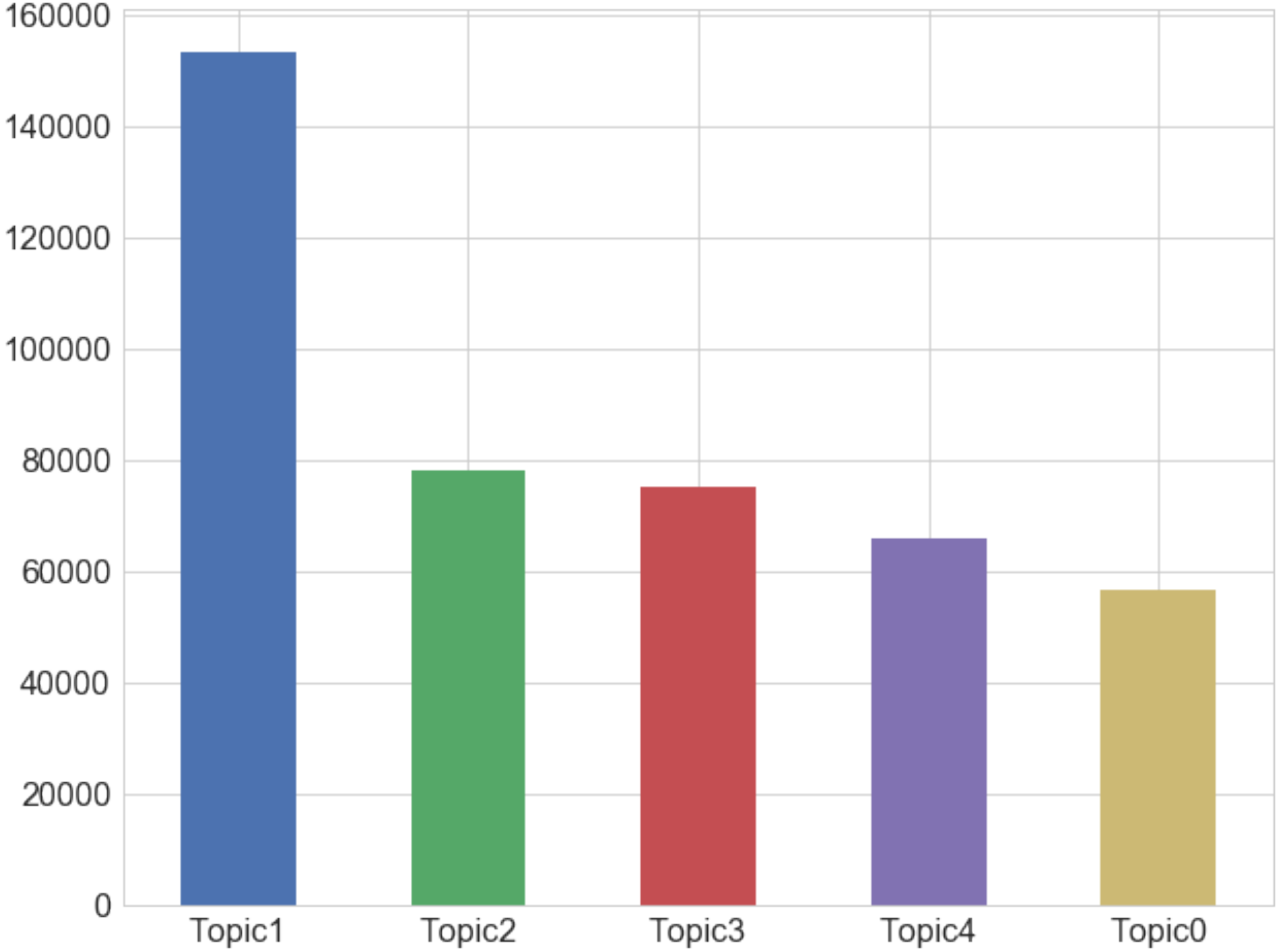
Topic 4: room location good work night breakfast floor window air door noise star open noisy small old bad need clean poor

Display a Document - Topic Table for the First 10 Reviews



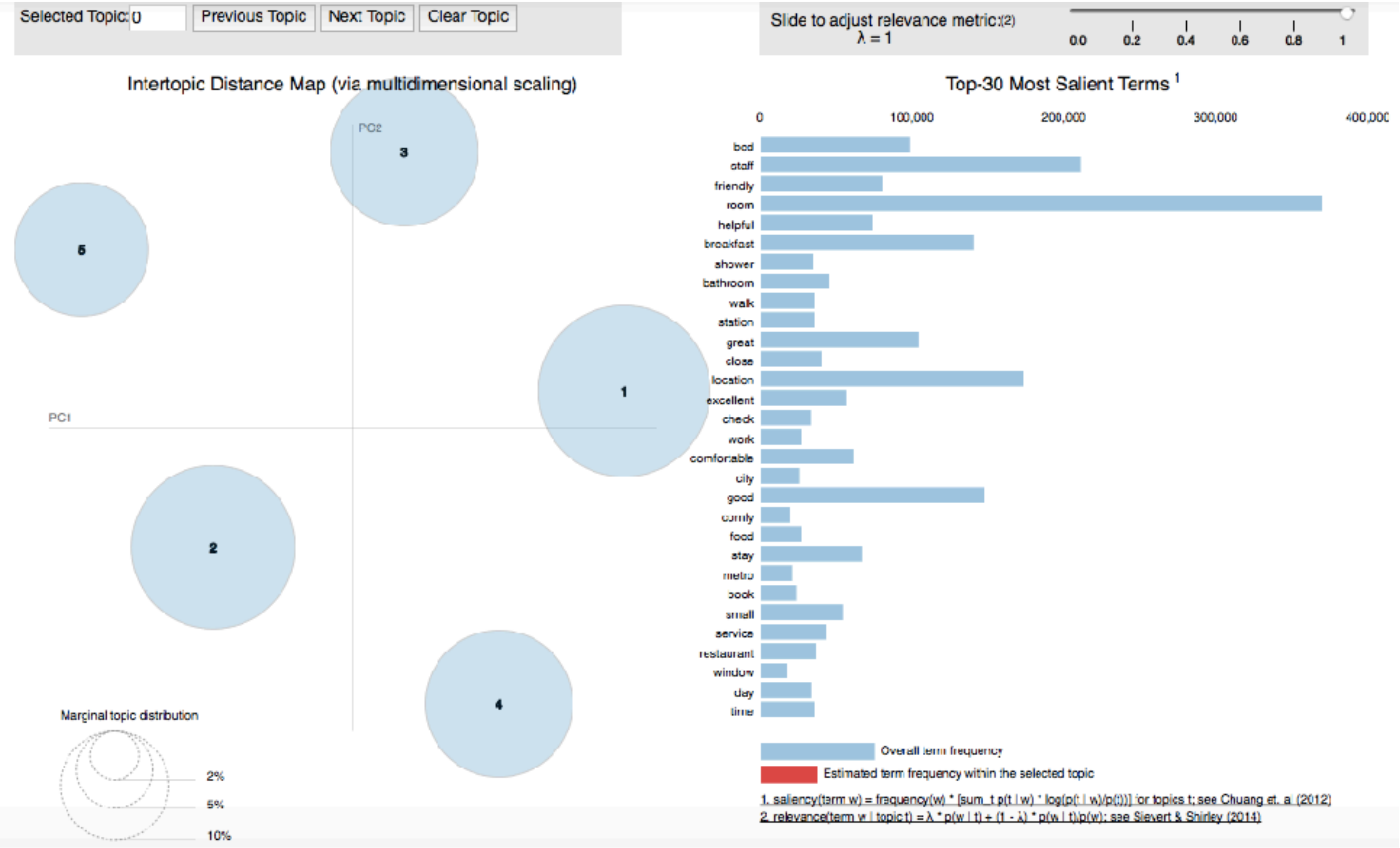
	Topic	Topic	Topic	Topic	Topic	domina
Doc0	0.03	0.00	0.07	0.47	0.43	3
Doc1	0.12	0.45	0.11	0.32	0.00	1
Doc2	0.20	0.41	0.01	0.12	0.26	1
Doc3	0.08	0.00	0.06	0.25	0.60	4
Doc4	0.00	0.00	0.06	0.68	0.26	3
Doc5	0.01	0.38	0.30	0.16	0.14	1
Doc6	0.89	0.01	0.08	0.01	0.01	0
Doc7	0.01	0.88	0.09	0.01	0.01	1
Doc8	0.22	0.02	0.02	0.34	0.41	4
Doc9	0.27	0.22	0.25	0.01	0.26	0

Topics distribution across documents



Most of the documents in our sample seems to about topic 1.

pyLDAvis



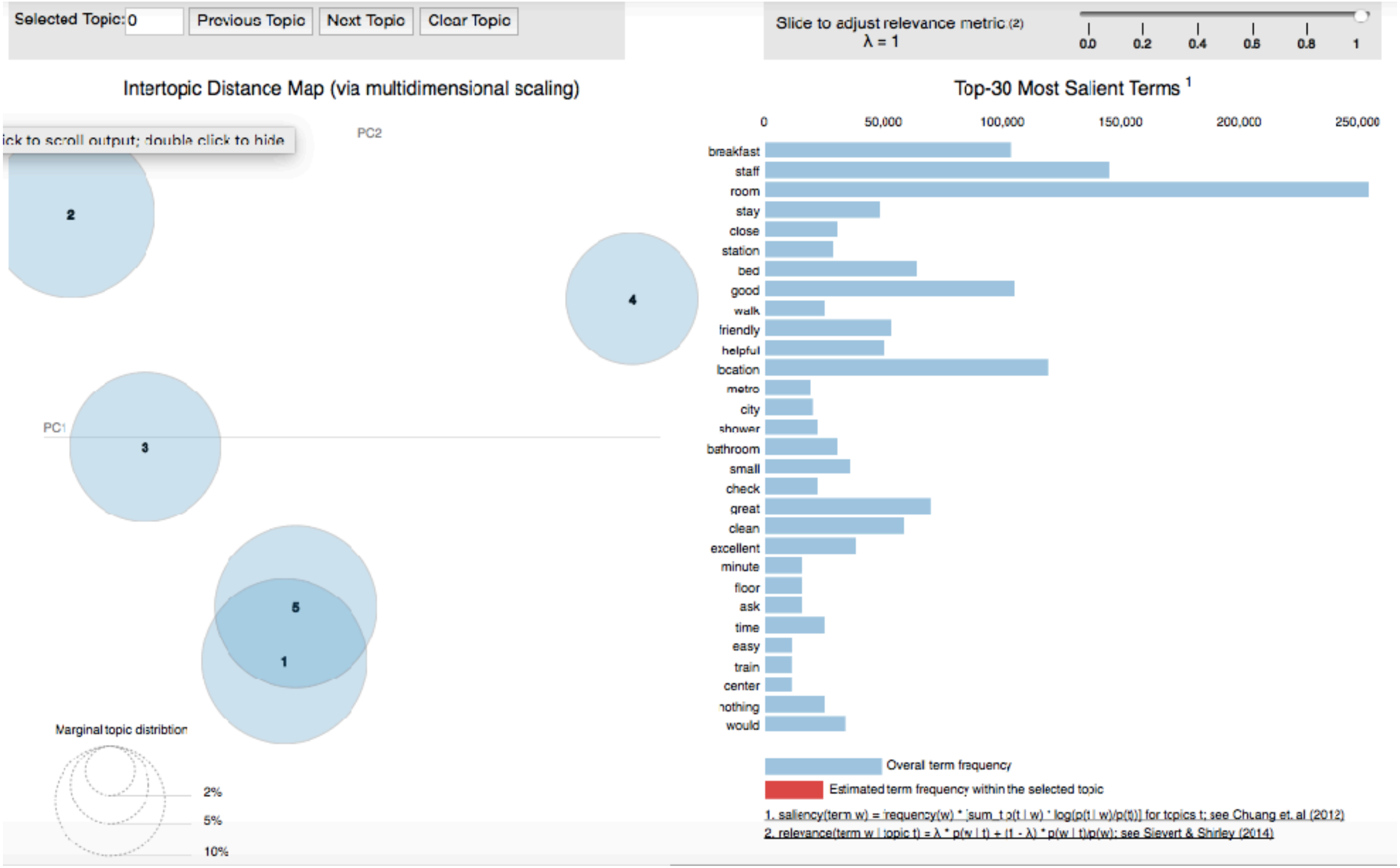
Selected Topic: 0

Slide to adjust relevance metric:(2) λ = 1

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)

2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

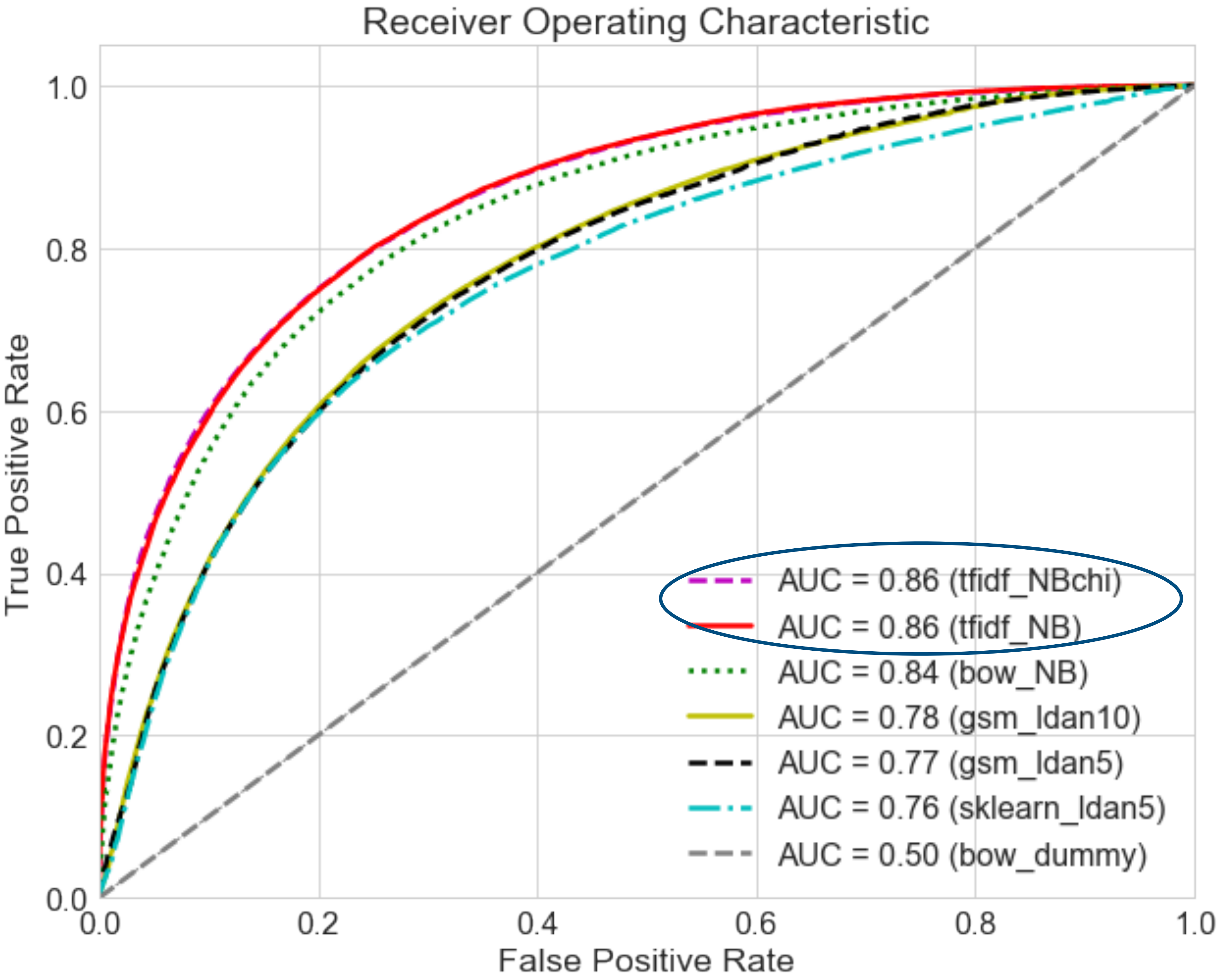
LDA with Gensim



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)

2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

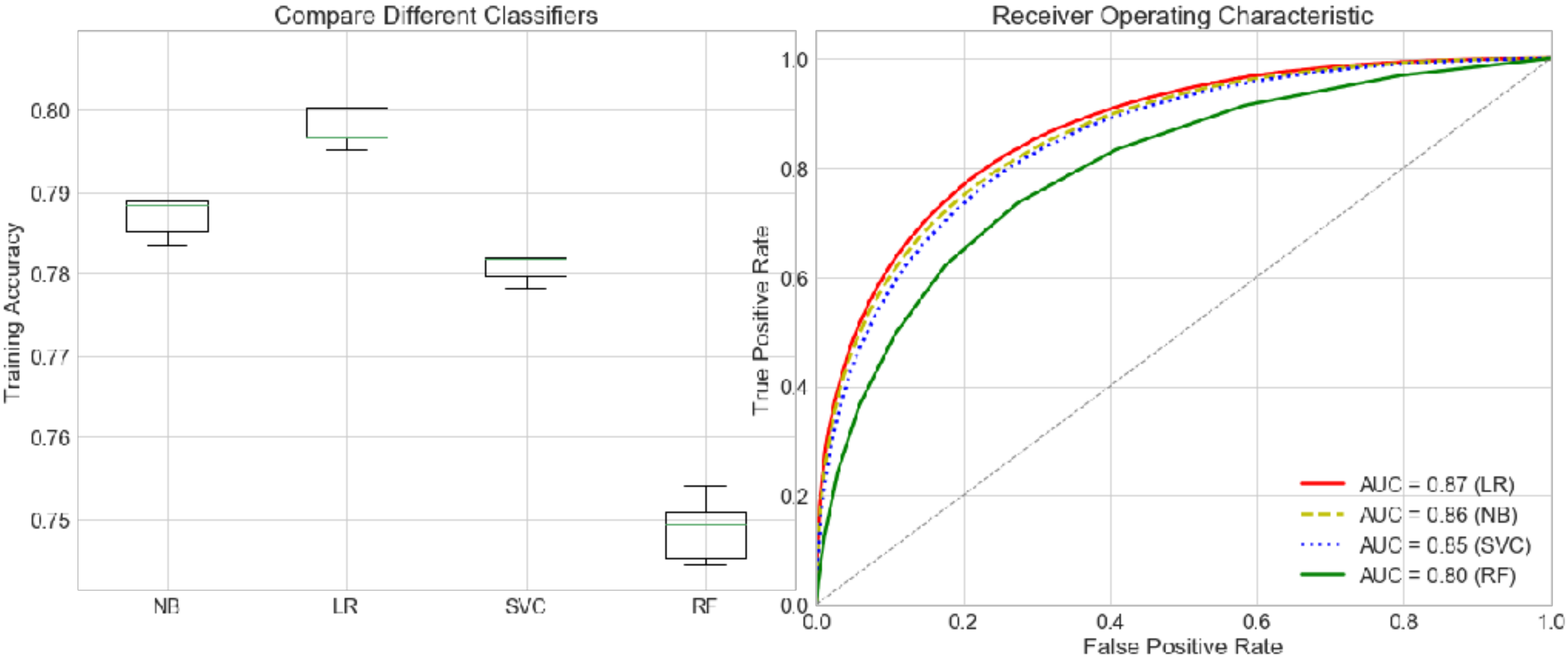
Pick Features



Concerning texts, the TF-IDF vectorized features as inputs give the best model performance.

Pick Classifiers

Take TF-IDF vectorized features to represent review texts.



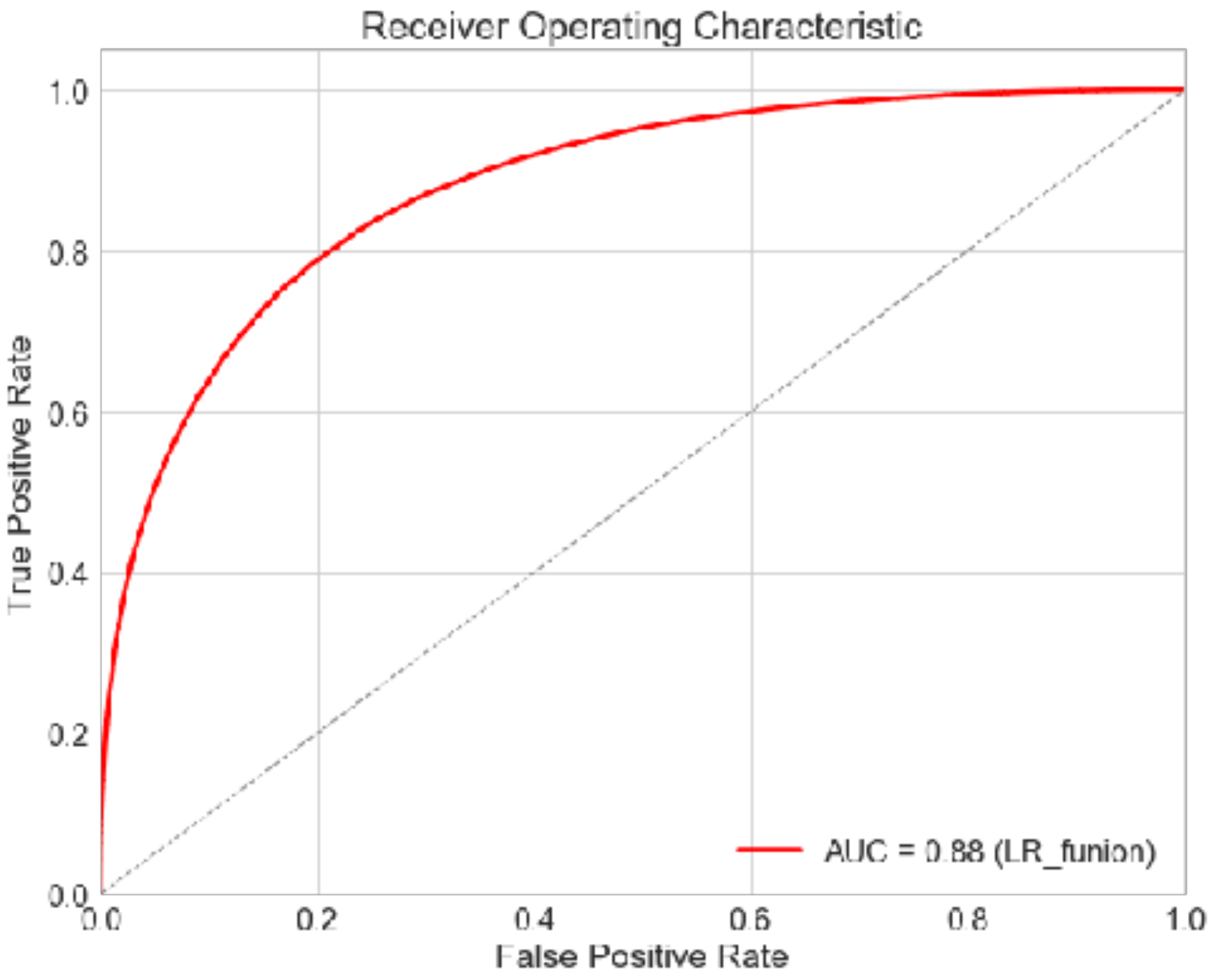
LR: LogisticRegression, **NB: MultinomialNB**, **SVC: SGDClassifier**, **RF: RandomForestClassifier**

Further Improvement

- Grid search hyperparameters for LogiticRegression classifier
- Enrich predictors with categorical and numerical features via FeatureUnion

Classification Report:

	precision	recall	f1-score	support
Good	0.83	0.88	0.85	82592
Poor	0.76	0.68	0.72	46248
avg / total	0.8	0.81	0.8	128840



How to Use the Model

- Select features from the dataset
- Extract texts and vectorize texts
- Extract and perform one-hot encoding on categorical features
- Extract, impute and scale numerical features
- Union all features
- Run model pipeline and predict the label of hotels

Conclusions

- We’ve performed data wrangling and exploratory analysis on hotel review data in aspects of hotels, reviewers and reviews.

- We’ve implemented topic modeling on review texts. The topics are not as interpretable as expected.

- We’ve conducted NLP analysis and generated vectorized (BOW, TF-IDF, LDA topics) text features.

- Out of different types of vectorized features TF-IDF weighting works best.

- With TF-IDF features as input, logistic regression classifier performs best. With 70%-30% splitting, the test data set gave ROC AUC=0.88

Next Steps

Improve performance:

- Improve text cleaning, I.e., correcting mis-spelling, including n-grams
- Add interactions between features
- Resample to tackle imbalanced dataset

Other interesting questions include:

- Extract topics by months to explore if there is some trend in topics in a time series.
- Adjust the model and tune to a potential client.
- …

Thank you!

Yanhua Hou
Email: alicehou18@gmail.com
Github: https://github.com/phyhouhou
LinkedIn: https://www.linkedin.com/in/yanhuahou/