**Sentiment Analysis:**

We wanted to check the sentiment score of the reviews. In order to do that we decided to use the VADER sentiment analysis tool because it checks also the intensity of each review using several parameters: punctuation, capitalization, degree modifiers, shift in polarity due to “but” and examining the tri-gram before a sentiment-laden lexical feature to catch polarity negation.  
let’s start –

**import** twython *#actively maintained, pure Python wrapper for the Twitter API***import** pandas **as** pd *#in order to read the CSV file***import** numpy **as** np *#for mathematic and conditions***import** nltk *#download VADER library*nltk.download(**'vader\_lexicon'**)  
**from** nltk.sentiment.vader **import** SentimentIntensityAnalyzer *#importing the sentiment analysis tool  
  
  
#read Hotel Reviews file*HotelReview = pd.read\_csv(**'C:\Users\pamel\Desktop\College Papers\Data Analytics\\Hotel\_Reviews\_Fixed2.csv'**)

Before the sentiment analysis we used:

*#change all the reviews to lower letters*HotelReview[**'Negative\_Review'**] = [x.lower().strip() **for** x **in** HotelReview[**'Negative\_Review'**]]  
HotelReview[**'Positive\_Review'**] = [x.lower().strip() **for** x **in** HotelReview[**'Positive\_Review'**]]

It’s incorrect since Vader sentiment analysis also check the intensity using upper and lower case.

Moreover, in the data we can see that we have on both the negative and the positive reviews: ‘nothing’, ‘everything’, ‘no negative/positive’.

In order to separate those reviews, we used that way:

NegativeCount = HotelReview.apply(**lambda** x: 1 **if** x[**'Positive\_Review'**] == **'no positive' or** \  
 x[**'Positive\_Review'**] == **'nothing' or** \  
 x[**'Negative\_Review'**] == **'everything'** \  
 **else** 0, axis=1)  
  
PositiveCount = HotelReview.apply(**lambda** x: 1 **if** x[**'Negative\_Review'**] == **'no negative' or** \  
 x[**'Negative\_Review'**] == **'nothing' or** \  
 x[**'Positive\_Review'**] == **'everything'** \  
 **else** 0, axis=1)

it doesn’t work well when we want to use the Vader sentiment analysis because the analyzer doesn’t know the words came from a positive or a negative review, therefore those words seems to be all neutral. Moreover, it doesn’t reflect the sentiment analysis of the full review.

If we look at the combinations we can see that:  
1. If the positive review is everything **and** the negative review is nothing or no negative - > the review is fully positive.  
2. If the negative review is everything **and** the positive review is nothing or no positive - > the review is fully negative.  
3. If the negative review is nothing **or** no negative we need to check the intensity of the positive review.  
4. If the positive review is nothing **or** no positive we need to check the intensity of the negative review.  
5. All the other reviews can be checked regularly.

*#set up conditions to classify the reviews*Conditions = [((HotelReview[**'Lower\_Positive'**] == **'everything'**) & (HotelReview[**'Lower\_Negative'**] == **'no negative'**)) |  
 ((HotelReview[**'Lower\_Positive'**] == **'everything'**) & (HotelReview[**'Lower\_Negative'**] == **'nothing'**)),  
 ((HotelReview[**'Lower\_Negative'**] == **'everything'**) & (HotelReview[**'Lower\_Positive'**] == **'no positive'**)) |  
 ((HotelReview[**'Lower\_Negative'**] == **'everything'**) & (HotelReview[**'Lower\_Positive'**] == **'nothing'**)),  
 (HotelReview[**'Lower\_Negative'**] == **'no negative'**) | (HotelReview[**'Lower\_Negative'**] == **'nothing'**),  
 (HotelReview[**'Lower\_Positive'**] == **'no positive'**) | (HotelReview[**'Lower\_Positive'**] == **'nothing'**)]  
  
  
*#set up choices for the conditions*Choices1 = [1, 2, 3, 4]  
  
  
*#apply conditions and choices on the Hotel Review data*HotelReview[**'Review\_Count'**] = np.select(Conditions, Choices1, default=5)

Set up conditions to classify the reviews as we spoke about before.

Set up the groups based on the conditions of each one of them.

We created a column named “Review Count” that labels all the reviews by the groups 1-5.

After we classified the reviews, we need to analyze them base on the group they are in.

for example, if the review count is 1 it means the review sentiment score will be 1, total positive and if the review count is 2 it means the review sentiment score will be -1, total negative. Furthermore, if the review count is 3 we need to analyze only the positive review and if the review count is 4 we need to analyze only the negative review. In reviews that has a 5 score we need to analyze both.

We linked the VADER sentiment intensity analyzer in order to use it for our sentiment analysis. We created a function that uses the review count classification and return 1 for label 1, -1 for label 2 and the compound sentiment score of each review that has labels 3,4 or 5.

*#link the Vader sentiment intensity analyzer*sid = SentimentIntensityAnalyzer()  
  
  
*#create a sentiment analyze function***def** SentimentAnalyze():  
 **for** i,count **in** enumerate(HotelReview[**'Review\_Count'**]):  
 **if** count == 1:  
 HotelReview.loc[i,**'Review\_Sentiment\_Score'**] = 1 *#return 1 for review count 1* **elif** count == 2:  
 HotelReview.loc[i, **'Review\_Sentiment\_Score'**] = -1 *#return -1 for review count 2* **elif** count == 3:  
 review = HotelReview.loc[i,**'Positive\_Review'**]  
 HotelReview.loc[i, **'Review\_Sentiment\_Score'**] = sid.polarity\_scores(review)[**'compound'**] *#return compound score for review count 3* **elif** count == 4:  
 review = HotelReview.loc[i,**'Negative\_Review'**]  
 HotelReview.loc[i, **'Review\_Sentiment\_Score'**] = sid.polarity\_scores(review)[**'compound'**] *#return compound score for review count 4* **else**:  
 review = HotelReview.loc[i,**'Negative\_Review'**] + **' '** + HotelReview.loc[i,**'Positive\_Review'**]  
 HotelReview.loc[i, **'Review\_Sentiment\_Score'**] = sid.polarity\_scores(review)[**'compound'**] *#return compound score for review count 5*

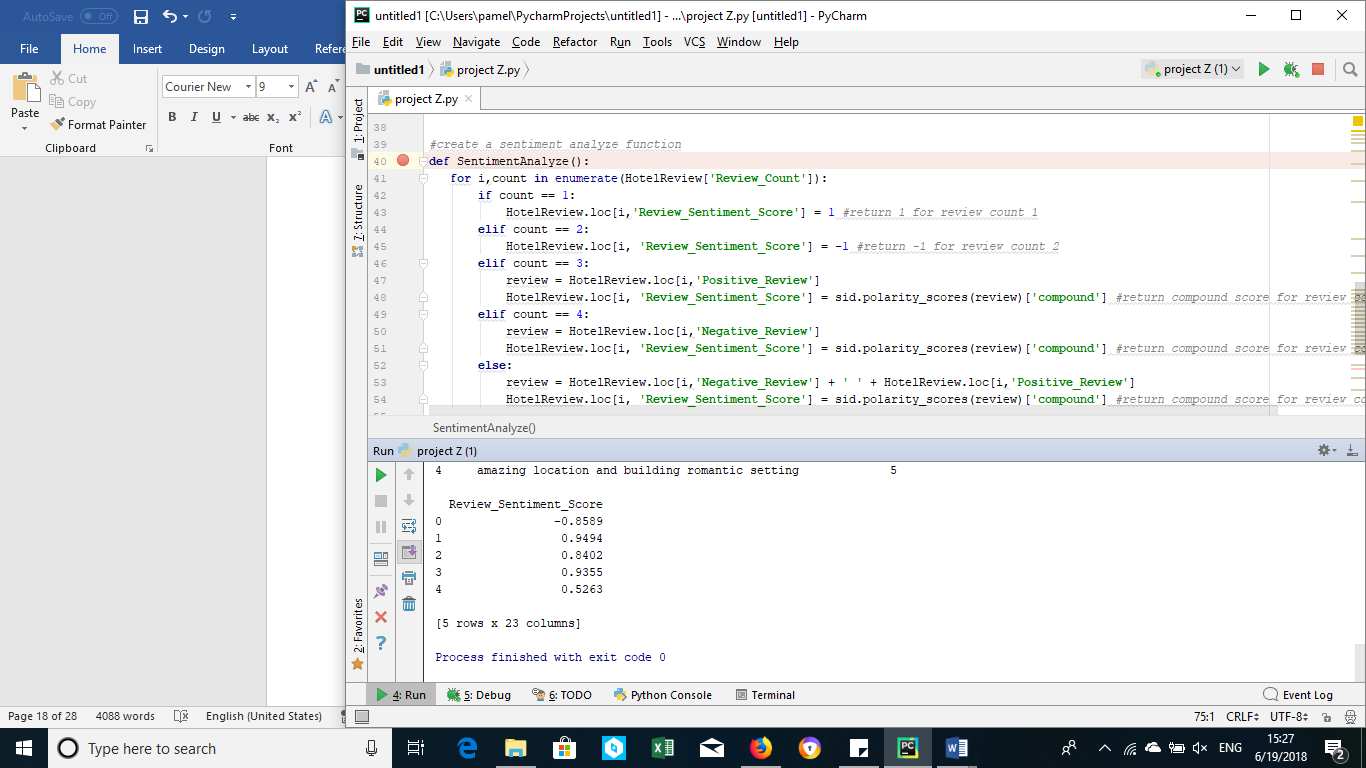
The function checks the index of each review and return its sentiment compound score based on its label.

The compound score is a sum of the normal scores: negative, neutral and positive of each review.

The reason we decided the range will be -1 to 1 (total negative to total positive) is because that’s the range of scores the VADER analyzer uses.

UTF-8: we decided to use this variable because it capable of encoding all 1,112,064 valid code points in Unicode using one to four 8-bit bytes.

After creating the function –

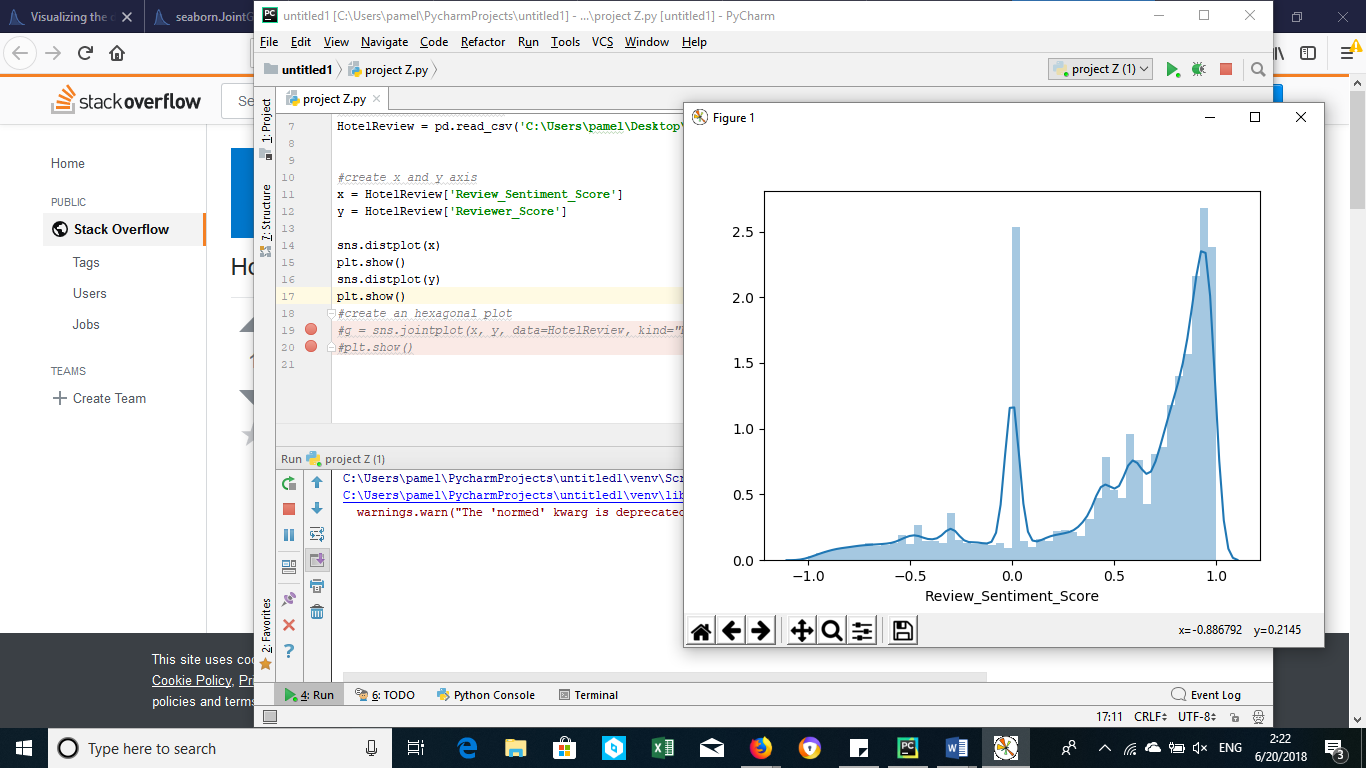
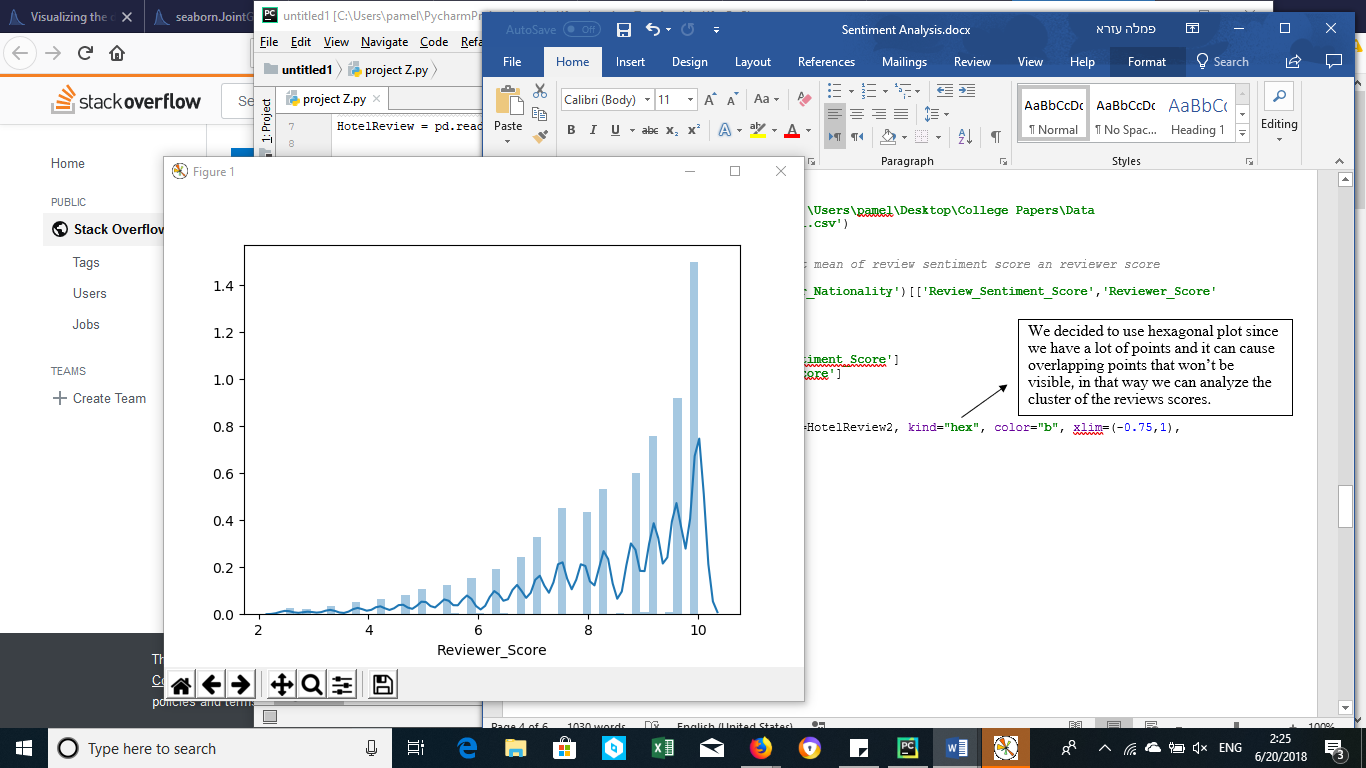
*#apply the function*SentimentAnalyze()  
  
*#save the data as new CSV file*HotelReview.to\_csv(**'C:\Users\pamel\Desktop\College Papers\Data Analytics\\Hotel\_Review\_Final.csv'**, encoding=**'utf-8'**)

Now we would like to see what the pattern of the reviews is based on their sentiment score and the reviewer score and if there’s a correlation between them.

**import** matplotlib.pyplot **as** plt *#show the plot***import** pandas **as** pd *#in order to read the CSV file and use correlation***import** seaborn **as** sns *#create a plot  
  
  
#read Hotel Reviews file*HotelReview = pd.read\_csv(**'C:\Users\pamel\Desktop\College Papers\Data Analytics\\Hotel\_Review\_Final.csv'**)  
  
  
*#create x and y axis*x = HotelReview[**'Review\_Sentiment\_Score'**]  
y = HotelReview[**'Reviewer\_Score'**]  
  
sns.distplot(x)  
plt.show()

and then –

sns.distplot(y)  
plt.show()

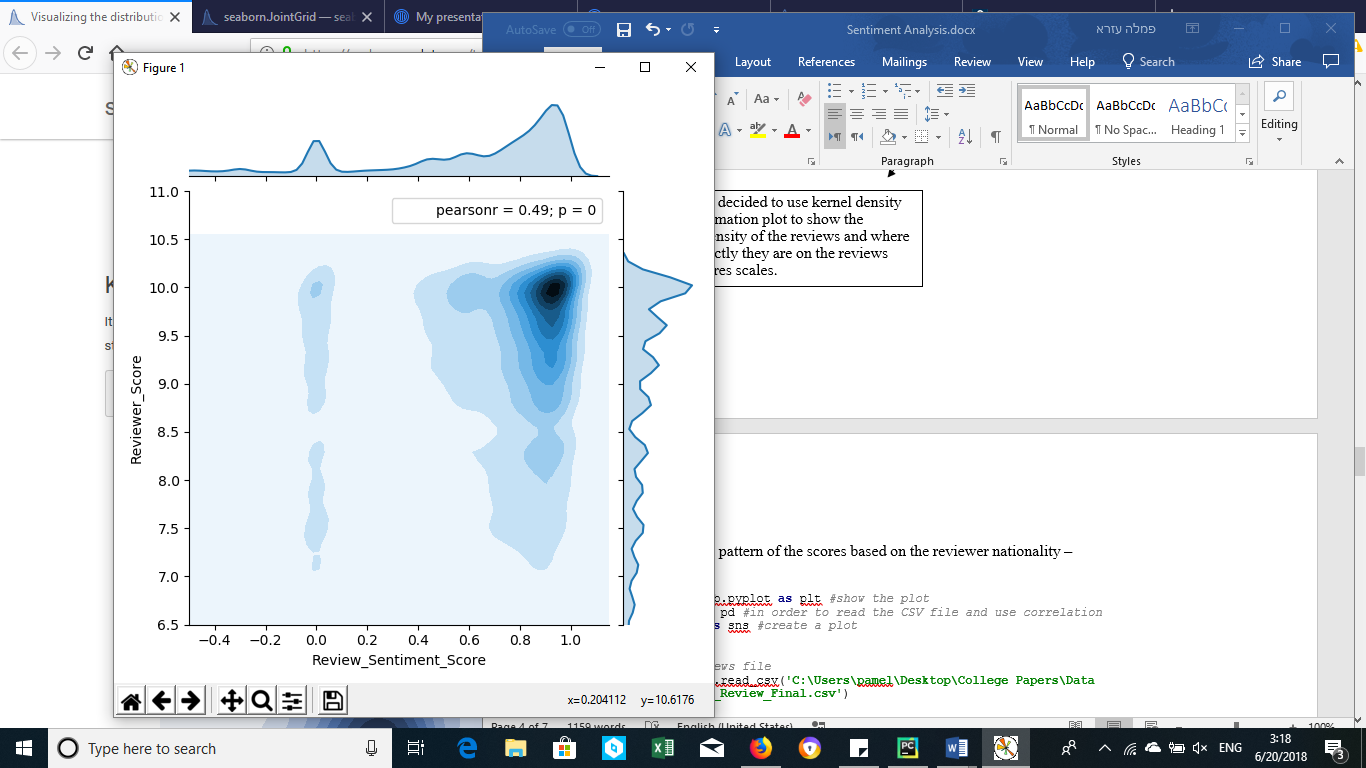


We can see that the scores from both the reviewers and the sentiment analysis tends to be positive, although in the sentiment analysis there’s also a massive amount of neutral reviews score.

Now let’s see what’s the connection between them –

*#create a kde plot*g = sns.jointplot(x, y, data=HotelReview, kind=**"kde"**, xlim=(-0.5,1.15), ylim=(6.5,11))  
plt.show()

We decided to use kernel density estimation plot to show the intensity of the reviews and where exactly they are on the reviews scores scales.



We can see that the strongest focus is on the higher scores of each axis, and we also have a weaker focus on the neutral sentiment score. Most of the reviewers scores are between 8.5-10 and most of the sentiment scores are between 0.6-1. The Pearson’s correlation coefficient is 0.49, if the value lies between 0.30-0.49 it reflects a moderate degree of a medium correlation. However, our PCC is closer to the high degree which reflects strong correlation. It means that there is a positive connection between the reviewer score and the review sentiment score.

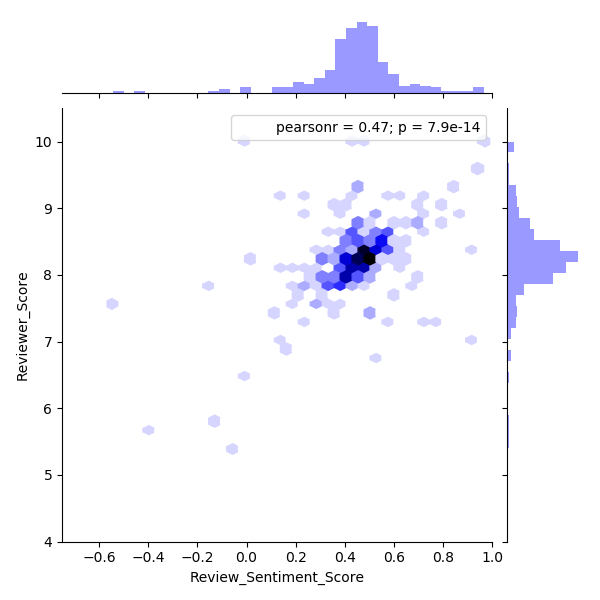
Let’s try to see the pattern of the average scores based on the reviewer nationality –

**import** matplotlib.pyplot **as** plt *#show the plot***import** pandas **as** pd *#in order to read the CSV file and use correlation***import** seaborn **as** sns *#create a plot  
  
  
#read Hotel Reviews file*HotelReview = pd.read\_csv(**'C:\Users\pamel\Desktop\College Papers\Data Analytics\\Hotel\_Review\_Final.csv'**)  
  
  
*#group by each nation and get mean of review sentiment score an reviewer score*HotelReview2 = HotelReview.groupby(**'Reviewer\_Nationality'**)[[**'Review\_Sentiment\_Score'**,**'Reviewer\_Score'**]].mean()  
  
  
*#create x and y axis*x = HotelReview2[**'Review\_Sentiment\_Score'**]  
y = HotelReview2[**'Reviewer\_Score'**]  
  
  
*#create an hexbin plot*g = sns.jointplot(x, y, data=HotelReview2, kind=**"hex"**, color=**"b"**, xlim=(-0.75,1), ylim=(4,10.5))  
plt.show()

We decided to use hexagonal plot since we have a lot of points and it can cause overlapping points that won’t be visible, in that way we can analyze the cluster of the reviews scores.

We can see that the division is normal, most of the values are in the center of the division between 8-9 on the reviewer score axis and between 0.4-0.6 on the review sentiment score. The Pearson’s correlation coefficient is 0.47, which reflects again of a medium connection between those two.

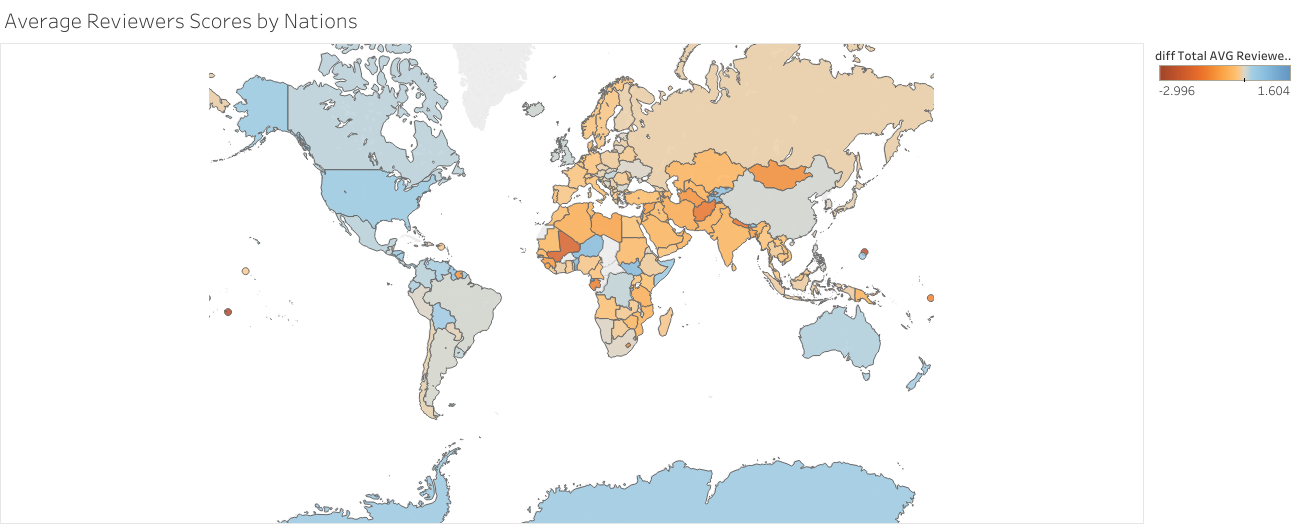
The most interesting thing about this graph and the one before is that even though we can see some negative and neutral scores in the sentiment score the reviewers scores don’t go below 5, which means that even if the reviewer had a negative experience in the hotel, he provided a neutral score or even a high one.

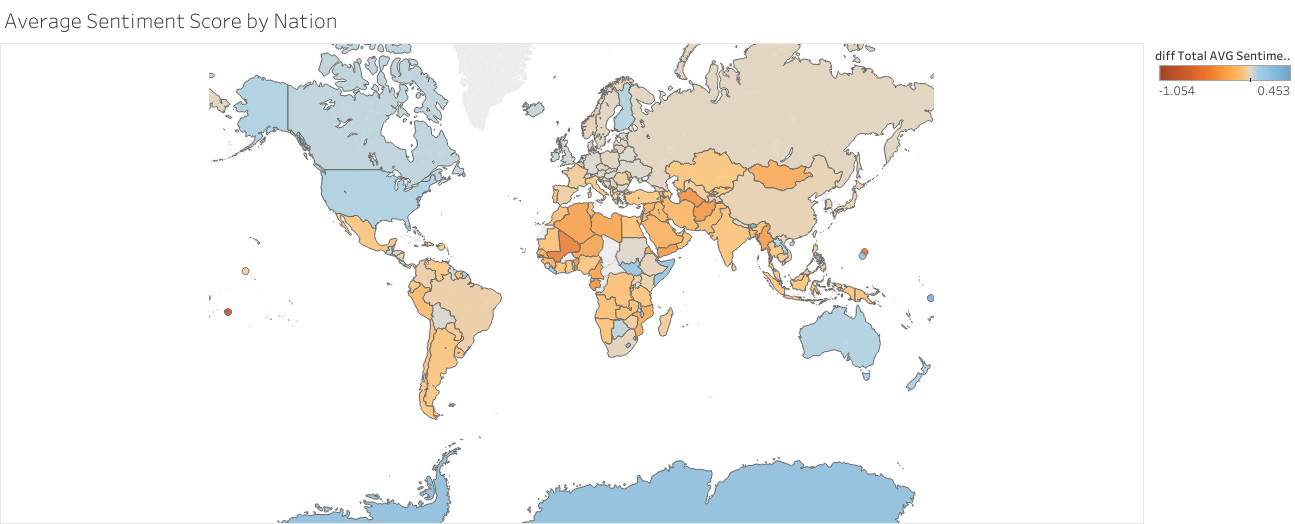


Those graphs aren’t enough.

We want to visualize it better so that we can see the differences between each nation and the differences of the scores.

Let’s see the reviewers scores and the sentiment scores –





We can see there’s countries that their review score changes entirely through the analysis. Some countries gave above the average score, but their sentiment score shows they were actually below the average, and some countries were exactly the opposite. It can reflect their thinking of the score, while some countries gave high score they interpreted the score as low one, or perhaps they are more generous with their scoring.

Moreover, the best scores were from people outside Europe, like North America, Antarctica, Australia etc. perhaps it reflects the fact those visitors traveled long distance and they already came with more positive thoughts and excitement then people inside Europe and close to it.

For further analysis:   
We can check for each hotel which nation gave them the highest average score , this can help targeting the hotel goal for those visitors.