by Cesar Cisneros, Albina Chowdhury, Estelle Hooper, and Eva Ruse

In [6]:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from sklearn.linear_model import LinearRegression as lr
```

Table of Contents

I. Introduction: How do you measure happiness?

Motivation

Music is widely used to cope with any mood by most individuals across the world. Spotify has taken advantage of music as therapy, generating several playlists related to emotion, such as Mood Booster (https://open.spotify.com/playlist/37i9dQZF1DX3rxVfibe1L0?si=1ac560999d7f4489) to "get happy" and Life Sucks (https://open.spotify.com/playlist/37i9dQZF1DX3YSRoSdA634?si=ce50da9a976e41d9) for when you feel like "everything in life sucks." In fact, for every song on Spotify, Spotify has calculated individual scores to measure the "mood" of the song (danceability, valence, energy, tempo).

According to the Sustainable Development Solutions Network (SDSN, an initiative launched by the United Nations), a *country's* happiness can be calculated by economic production , life expectancy, and more. Though the music the people of said country listen to is obviously not one of these factors, we are more interested in discovering if there are any notable differences between popular music in "happy" countries and in "sad" countries.

Research Question

> **Is there a relationship between how happy a country is and the music the people of this country listen to?**

In our project, we will test the worthiness of these Spotify-generated scores related to happiness and explore any biases in Spotify's playlist creation algorithm. For example, according to the SDSN, the happiest country in 2020 is Finland, but the top-streamed songs in Finland are absent in Spotify's playlists to encourage happiness (eg Mood Booster (https://open.spotify.com/playlist/37i9dQZF1DX3rxVfibe1L0? si=1ac560999d7f4489)).

With more than 155 million (as of 2020) (https://www.statista.com/statistics/244995/number-of-payingspotify-subscribers/#:~:text=How%20many%20paid%20subscribers%20does,than%20doubled%20since%20early% **paid** listeners, it is imperative that Spotify provide them with user-centered content. The SDSN has even greater influence, partnering and receiving funds from organizations all around the world to create these happiness datasets.

Our hope with this project is **not** to argue that Spotify's happiness scores should be made with consideration to the SDSN or vice versa, but to initiate conversation on the inclusivity of Spotify's algorithms and question the validity of how powerful organizations (such as Spotify and the SDSN) quantify happiness.

Our Process

Using the SDSN's "Worldwide Happiness Report 2019", we identified the top 10 and bottom 10 "happiest" countries. It is important to note that we selected these countries based on availability of Spotify data, not the actual happiness scores. For example, South Sudan is the last happy country at rank 156. But likely because of economic reasons, South Sudan does not use Spotify. Our least happy country is South Africa, at rank 106. We found the top-streamed songs from each of the 20 countries using Spotify Charts and used the Spotify API to get the songs' happiness-related scores (eg. valence). Finally, we ran statistical analyses to explore if there was a difference in these song scores between the top 10 and bottom 10.

Results Summary

Overall, we found that there was no notable differences in the song attributes between happy and sad countries. Though we initially found that the top 10 countries tended to have slightly higher valence scores, it was not enough to conclude a general trend that happier countries tended to listen to happier music, seeing as we are comparing two extreme groups. We found this conclusion reasonable, as individual tastes of music are highly variable. Some people enjoy listening to happy music when they are sad, and some people like)melodramatic music when they are happy.

However you are feeling at the moment, we have created two playlists for you to optionally listen to as you read our notebook.

1. Happy: the happiest songs of 2020 of the 10 happiest and 10 least happy countries (some overlap between countries) (https://open.spotify.com/playlist/2BrmDGLVRlD7r2Cf8culSw? si=324b4d727744488d)
2. Sad: the saddest songs of the 10 happiest and 10 least happy countries (some overlap between countries) (https://open.spotify.com/playlist/4ntdqpIQ5Eus34Ds9gDQNm?si=b2280c1cc4a84f6f)

II. Data Descriptions: SDSN World Happiness Reports, Spotify Charts, Spotify API

1. "World Happiness Report 2019" (https://www.kaggle.com/unsdsn/world-happiness) by Sustainable Development Solutions Network
(The following is a cleaned version of the dataset. The raw can be found here)

- variable assignment throughout notebook: `happy2019`
- .csv file: `happy2019.csv`

1. "World Happiness Report 2020 (https://www.kaggle.com/londeen/world-happiness-report-2020?select=WHR20_DataForFigure2.1.csv) by Michael Londeen (Kaggle User, adapted from Sustainable Development Solutions Network)
(The following is a cleaned version of the dataset. The raw can be found here.)

- variable assignment throughout notebook: `happy2020`
- .csv file: `happy2020.csv`

In [7]:

```
happy2019=pd.read_csv('happy2019.csv')
happy2019=happy2019.drop(columns=['Unnamed: 0'])
happy2019
```

Out[7]:

	happiness_rank	country	happiness_score
0	1	Finland	7.769
1	2	Denmark	7.600
2	3	Norway	7.554
3	4	Iceland	7.494
4	5	Netherlands	7.488
5	6	Switzerland	7.480
6	7	Sweden	7.343
7	8	New Zealand	7.307
8	9	Canada	7.278
9	10	Austria	7.246
10	66	Portugal	5.693

11	69	Philippines	5.631
12	76	Hong Kong	5.430
13	77	Dominican Republic	5.425
14	79	Turkey	5.373
15	80	Malaysia	5.339
16	82	Greece	5.287
17	92	Indonesia	5.192
18	94	Vietnam	5.175
19	106	South Africa	4.722

In [8]:

```
happy2019=pd.read_csv('happy2019.csv')
happy2019=happy2019.drop(columns=['Unnamed: 0'])
happy2019
```

Out[8]:

	happiness_rank	country	happiness_score
0	1	Finland	7.769
1	2	Denmark	7.600
2	3	Norway	7.554
3	4	Iceland	7.494
4	5	Netherlands	7.488
5	6	Switzerland	7.480
6	7	Sweden	7.343
7	8	New Zealand	7.307
8	9	Canada	7.278
9	10	Austria	7.246
10	66	Portugal	5.693
11	69	Philippines	5.631
12	76	Hong Kong	5.430
13	77	Dominican Republic	5.425
14	79	Turkey	5.373
15	80	Malaysia	5.339

16	82	Greece	5.287
17	92	Indonesia	5.192
18	94	Vietnam	5.175
19	106	South Africa	4.722

The happiness score dataset’s rows are the countries that were included in the Gallup World Poll; its columns are country, happiness score, happiness rank, GDP per capita, family, life expectancy, freedom, generosity, government corruption, and dystopia residual. Factors such as GDP per capita, freedom, etc. are used to compute a country’s happiness score, and then their rank once their score is viewed in relation to other countries.

According to the Kaggle description, this dataset was created to measure progress in nations by calculating their happiness, and to explain why people of some countries are happier than others. It answers questions such as “What countries or regions rank the highest in overall happiness and each of the six factors contributing to happiness?” “Did any country experience a significant increase or decrease in happiness?”. Since this dataset is published by the Sustainable Development Solutions Network, we can assume that the United Nations funded the creation of this dataset – or at least the data within it.

While Gallup has a very thorough methodology when it comes to carrying out the Gallup World Poll there is still the possibility that people are not honest when they answer the question, or that the poll is not reaching a large/diverse enough group. Additionally, the data that is collected through this poll is affected by the type of person who typically responds to polling questions; for example, people who have very strong opinions about the topic are more likely to agree to be polled than those who have weaker opinions. This may skew the responses to the data.

To compile this dataset, the Sustainable Development Solutions Network had to take the data from the world happiness reports (which have separate reports for each year) and append the poll data into the necessary columns for each country. This required the Sustainable Development Solutions Network to pick and choose the data in the actual world happiness report that they believed was important to include in the Kaggle dataset.

The happiness scores are calculated using data that is collected through the Gallup World Poll. The people who answer these questions are aware of the data collection. This is a well-known report that is compiled, however, we are unsure if they are aware of the purpose of the survey. This is the link to the dataset that was uploaded to Kaggle by the SDSN: https://www.kaggle.com/unsdsn/world-happiness (https://www.kaggle.com/unsdsn/world-happiness)

We used four datasets for this project. One was a Kaggle dataset that compiled the happiness scores and happiness ranks for countries in 2019 and then we also used a similar dataset that compiled these scores and ranks for countries in 2020. The other two datsets we used pertained to Spotify data; we created these datasets by webscraping the Spotify Charts website. One dataset included the top 50 songs per country in 2019 and the other dataset included the top 50 songs per country in 2020.

World Happiness Report Dataset

The happiness score dataset's rows are the countries that were included in the Gallup World Poll; its

columns are country, happiness score, happiness rank, GDP per capita, family, life expectancy, freedom, generosity, government corruption, and dystopia residual. Factors such as GDP per capita, freedom, etc. are used to compute a country's happiness score, and then their rank once their score is viewed in relation to other countries.

Much of the data for the world happiness reports comes from the Gallup World Poll. This poll conducts randomized telephone or face-to-face interviews depending on the percentage of a country's population that has a telephone (cite). The world happiness report is usually published by the Sustainable Development Solutions Network, a branch of the United Nations. The Sustainable Development Solutions Network uploaded the data set for 2015-2019 onto Kaggle. We only use the 2017 dataset. According to the Kaggle dataset, the inspiration for this dataset was to discover the answers to questions like, "What countries or regions rank the highest in overall happiness and each of the six factors contributing to happiness?", "How did country ranks or scores change between the 2015 and 2016 as well as the 2016 and 2017 reports?", "Did any country experience a significant increase or decrease in happiness?" (cite). Since this dataset is published by the Sustainable Development Solutions Network, we can assume that the United Nations funded the creation of this dataset – or at least the data within it.

While Gallup has a very thorough methodology when it comes to carrying out the Gallup World Poll there is still the possibility that people are not honest when they answer the question, or that the poll is not reaching a large/diverse enough group. Additionally, the data that is collected through this poll is affected by the type of person who typically responds to polling questions; for example, people who have very strong opinions about the topic are more likely to agree to be polled than those who have weaker opinions. This may skew the responses to the data.

To compile this dataset, the Sustainable Development Solutions Network had to take the data from the world happiness reports (which have separate reports for each year) and append the poll data into the necessary columns for each country. This required the Sustainable Development Solutions Network to pick and choose the data in the actual world happiness report that they believed was important to include in the Kaggle dataset.

The happiness scores are calculated using data that is collected through the Gallup World Poll. The people who answer these questions are aware of the data collection. This is a well-known report that is compiled, however, we are unsure if they are aware of the purpose of the survey. This is the link to the dataset that was uploaded to Kaggle by the SDSN: https://www.kaggle.com/unsdsn/world-happiness (https://www.kaggle.com/unsdsn/world-happiness)

Spotify Dataset

Ultimately, the spotify datasets for 2019 and 2020 incudes data from the happiness datasets. Each row of the dataset corresponds to some top song that a country listened to. The country column contains the name of the country for which the song was a top song. Other columns are `happiness_score` , `happiness_rank` , `date` , `position` , `track_name` , `artist` , `streams` , `danceability` , `energy` , `key` , `loudness` , `mode` , `speechiness` , `acousticness` , `instrumentalness` , `liveness` , `valence` , `tempo` , `id` , and `duration_ms` . Columns such as `happiness_score` and `happiness_rank` come from the Kaggle dataset that we described above. The rest of the columns contain the audio feature data for each song that we got using the Spotify API. We have two datasets that followt this model, one that has the top 50 songs for each country in 2019 and then one that has the top 50 songs for each country in 2020. For each dataset, we look only at the top 10 and bottom 10 happiest countries in that year according to their happiness score provided by the Kaggle dataset.

We initally were going to use a Kaggle dataset that compiled the top 200 songs per country in 2017, however, for our data analysis we wanted to create a training model to test our hypothesis. In order to do this, we needed another year of data so we could test our model on another set of data that it wasn't trained on. After we learned how to scrape data from Spotify Charts, we decided to do the same for 2019 instead of 2017. While we could have scraped the data for 2017, we decided to use 2019's data since we believed that training a model and testing it on datasets that are closer together in age would remove the possibility that changing trends in music would obscure the relationship between a country's happiness and the songs they listen to. Overall, the purpose of creating this dataset was to be able to use it in comparison with countries' happiness scores. No one funded the creation of this dataset.

When webscraping the Spotify Charts website, some countries (particularly countries that had the smallest happiness scores) did not have data listed for every day of the year that we were looking at. Additionally, we only chose to include countries for which we also had their happiness scores for. Lastly, we only chose to look at the top 50 songs instead of the top 200 due to Spotify's limit on calls to their API.

This is the link to the Spotify Charts website, which we used to webscrape the top songs for countries in 2019 and 2020: https://spotifycharts.com/regional (https://spotifycharts.com/regional) To use the website, you select the region that you want to look at and the day.

1. Worldwide Daily Spotify Ranking 2019 by us, compiled by webscraping Spotify Charts (https://spotifycharts.com) and calling the Spotify API (https://developer.spotify.com/documentation/web-api/reference/)
   (The following is a cleaned version of the dataset. The raw can be found here.)

- variable assignment: `spotify2019`
- .csv file: `spotify2019.csv`

1. Worldwide Daily Spotify Ranking 2020 by us, compiled by webscraping Spotify Charts

(https://spotifycharts.com) and calling the Spotify API (https://developer.spotify.com/documentation/web-api/reference/)
(The following is a cleaned version of the dataset. The raw can be found here.)

- variable assignment: `spotify2020`
- .csv file: `spotify2020.csv`

III Pre-registration Statements

The two analyses that we have chosen to perform in our final project are:

1. What is the relationship between the happiness score of a country and the average danceability score of its top songs over two-week periods for a year?
2. What is the relationship between the happiness score of a country and the average valence score of its top songs over two-week periods for a year?

   Analysis #1
   Does the danceability of a song have any correlation with the happiness score of a country? For our first analysis, we will be performing the rolling average of two weeks (14 days) for our data frame. In sum, this rolling average will take the average danceability score over 14 days by collecting the 50 songs for this period for the top happiest and bottom happiest countries. In particular, we believe there may be a relationship between those countries that score high in the happiness rank and those countries with songs that have high danceability scores. Danceability is often correlated with happiness, hence we believe the top ten [happiest] countries may exhibit particularly high danceability scores. The goal of our project is to understand whether there is any meaningful relationship between happy countries and the music they listen to. As a result, performing a rolling average of the danceability scores of the top happiest and least happy countries will allow us to observe one dimension of happiness from the lenses of danceability. As stated, we expect to see the top happiest countries (e.g., Finland and Denmark) to have high danceability rolling average scores compared to the least happy countries (e.g., South Africa and Vietnam).

Analysis #2
Can we predict the happiness ranking of a country based on Spotify-provided happiness scores (“valence”)? For every song, the Spotify API provides its own “happiness” score, called “valence.” The higher the valence score, the more positive the mood for the song is. Using these provided happiness scores, we want to compare the valence for popular songs in sad countries and popular songs in happy countries. If happy countries do have higher valence scores than sad ones, we may find a relevant connection between happiness ranks and the valence scores. If sad countries have higher valence scores than happy countries, we may still be able to find a connection between the variables and presume that sad countries may want to listen to happier songs. Even if there is no significant difference between the valence scores between happy and sad countries, we can still analyze the results and conduct more outside research as to why there is no correlation. This may be related to other Spotify-provided scores, such as “danceability”. Some countries, perhaps Hispanic ones, may have dance built into their cultures even though the happiness scores are variable. All in all, we may find that Spotify’s valence score does not accurately indicate happiness.

IV Data Analysis

Evaluation of significance

We performed a rolling average of two weeks for the songs of the top and bottom 10 happiest countries to discern any relationships. An assumption we had to make was that fourteen days was sufficient to conclude for the general music patterns observed in these countries, which limits our interpretations to this window. This was done for optimization reasons. As it pertains to our reasoning behind deciding to run a rolling average, we wanted to understand the possibility of there being a relationship between happiness score and individual features over fourteen days. A rolling, or moving, average analyzes data points by creating a series of averages of different subsets of a full data set, which seemed appropriate for our purposes since we were working with a dataset of 20 countries for one year. Although we acknowledge that it would have been more compelling to evaluate a rolling average over a longer period beyond fourteen days, this was impractical given time constraints. We attempted to do this, and the calculation took over four hours; further, the Spotify API – which we are using to obtain all the features for our Spotify songs – could only be run a specific number of times before we got a timing error. Spotify updated its usage terms for its API, hence doing a longer period resulted virtually impossible.

The rolling average results were consistent with our hypothesis, asserting that happier countries listened to music with higher danceability and valence scores. In particular, the line of best fit was steeper for top countries, and its distribution of country rolling averages of danceability and valence scores started at a higher score than the bottom countries. By the same token, the distribution of the bottom countries was scattered below the valence distribution of the top countries, which means that top countries were scoring relatively higher in valence score than the bottom countries. Most compellingly, we observed an outlier with the Dominican Republic as one of our bottom happy countries, which unexpectedly ranked extraordinarily high for its danceability and valence score. In fact, it was the highest valence and danceability score out of all 20 countries in this dataset. We understood this extreme outlier likely influenced our interpretation results of significance, so we resorted to performing the correlation of these rolling averages to see which features were correlated and better for further analysis.

We noticed that the happiness score had a relatively weak (near zero) correlation with valence and danceability – our two main analysis features – so we proceeded to do a single-variable linear regression model to predict happiness score for 2020 (including all individual features). Given that the Dominican Republic resulted in an extreme outlier, we felt it was appropiate to run the single-variable linear regression without the Dominican Republic, though we did this **after** running our model with the country included. Specifically, we focused on using certain features (danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo, and duration_ms) as our predictor variables and our happiness score as our outcome variable. Doing a linear regression model was appropriate for our model because we had already done the correlation of some of these features and happiness score, which all resulted in insignificant results. Therefore, modeling the relationship between happiness score and these features seemed most adequate by fitting a linear equation to the observed data. Our r^2 values for each feature, excluding the outlier Dominican Republic, were as followed: danceability had a value of -0.12, energy had a value of -0.144, key had a value of -0.125, loudness had a value of -0.123, mode had a value of -0.131, speechiness had a value of -0.124, acousticness had a value of -0.114, instrumentalness had a value of -0.126, liveness had a value of -0.122, valence had a value of -0.12, tempo had a value of -0.126, and duration_ms had a value of -0.083. Our r^2 values were negative for all features, which indicates that our model using these features is a bad predictor of happiness score. "Bad" in our context means these features are virtually trivial to predict what kind of music a country listens to. It is worth noting these features included danceability and valence.

With this in mind, we understood individual features were not particularly helpful to predict the happiness score of countries from 2020, so we felt compelled to try a multivariable linear regression model to observe how impactful all the [aforementioned] features were in predicting 2020 countries' happiness scores.The r^2 values we had for the multivariable model were -0.1(with the outlier) and -0.08 (without the outlier). All our r^2 values were negative, which means our multivariable model was not particularly good at predicting the happiness score of these countries. Specifically, this means features like danceability and valence, among others – together – are not good tools to predict what kind of music a country listens to. It should be noted that we ran a single-variable linear regression model after this multivariable regression model, but without the Dominican Republic included. Similarly, we also ran a multivariable linear regression model without the Dominican Republic. Both of these models resulted in similar results; their r^2 were also negative, thereby making our model ineffective to predict happiness scores for 2020.

Interpretation and conclusions

We predicted a country’s happiness score could be predicted by the type of music the people of that country listen to. Specifically, we thought features like how danceable a song is would ultimately give us more insight into the overall happiness of people from this country – irrespective of social or political factors affecting this score. Following conventional wisdom, we believed features like danceability, valence, instrumentalism, and liveness were representative of happiness. However, our results indicate precisely the opposite, thus making these features unfit for determining a country’s happiness. As such, we can only conclude that the selected features for Spotify are not a good predicting tool for a country’s happiness, as determined by its happiness score.

Without prior knowledge about this potential relationship between a country’s happiness and these Spotify features, analysts could potentially predict a country’s general happiness using features like liveness and danceability. This can be seen as a useful metric for this purpose, given the ability of music to uplift or distress people, which would be generalized to the entire population of certain countries. In particular, we often associate music with mood, given the ability of music to stimulate certain emotions or sensations. A prediction of this sort would theoretically impulse policy-makers – or anyone striving for the overall wellbeing of a given country – to implement measures and activities using music. Countries ranking low on their happiness score would attempt to assimilate to the music followings of higher-ranking countries on their happiness score. As it relates, pursuing a goal like this is not indicative of a country’s happiness and will likely produce a negligible effect on the country’s happiness.

Notwithstanding, there may be positive outcomes in following through with this approach. Although this would require further research on the subject matter, the exposure to “happy” music – as determined by high scores of danceability, or other similar features – can improve the well-being and emotional state of people. Indeed, authors writing on behalf of the American Psychological Association studied precisely this relationship in 2010, asserting that “happiness ratings were elevated for fast-tempo and major key-stimuli and sadness ratings were elevated for slow-tempo and minor-key stimuli" (Hunter). A different relationship may be observed in a longer period than 14 weeks, but this would require the use of more developed tools beyond only using the Spotify API, given the constraints it poses.

V Data Limitations

When conducting this analysis, our group used only the top ten and bottom ten happiest countries due to the limit on the number of calls you can make to the Spotify API. This limits our data size because we excluded the rest of the countries that are included within the happiness data. Further, this affects our data by also excluding countries with different cultures and the trends within those countries. We recognize that within our analysis that the cultures of the countries that we selected are not significantly varying, so we assumed that the type of music listened to within the different countries would not differ greatly. By way of example, a good portion of the top countries were Scandinavian countries, so this already influences the analyses in the top countries with the musical trends of Scandinavian countries.

Using data from 2019 further limits our data set because it is merely considering the songs that were popular during this period, which may not be representative of the typical songs listened to in the country. Using data from 2019, however, was the most logical to explore our research question given it was the most recent data set, providing the most recent musical trends of these countries.

Using the music platform Spotify is another limitation because it excludes the other music platforms that may be more popular in different countries. Spotify is not representative of all music platforms' data on global music streaming. Specifically, we are drawing conclusions about the music people listen to, which already limits the music in these countries to whatever is available on Spotify. Spotify is only available in 92 countries, which excludes the majority of African countries. For instance, Spotify is not available in China or Iran due to other popular music-streaming services (like QQ Music and NetEase Music for China) or because of laws preventing streaming music services to serve the public as is the case in Iran. In particular, whichever generalizations we make will be limited to countries in Spotify. Any countries where Spotify was not available were excluded from the analyses in this project.

While attempting to extract information from the Spotify API, our group came across issues with the security of Spotify. When re-running our code, some errors stated that we used our maximum amounts of retries. To mitigate this issue, we had to install a Spotify extension for our use. Additionally, Spotify installed an extra layer of security on their Spotify charts website, which hindered our ability to web-scrape the data for 2020. We were able to get 2020 data; however, some days may be missing for some countries (something out of our control), which poses a limitation in the interpretations we can draw since not all countries contain the same number of observations.

Another limitation our group recognized was that music listened to globally became more versatile since people of different nationalities are listening to international music due to the popular trends during that given period. As different countries listen to music from other countries, it may steer away from its popular cultural music, which can affect the data significantly. Similarly, certain songs may play a more influential role during certain holidays or special events (i.e., the month of December will likely correspond to similar songs across all countries).

Concerning our rolling average, rolling averages often include extreme outliers, which have the potential of affecting the data by skewing some of the output rolling averages and impacting the accuracy of our models.

Additionally, when computing the rolling average for each of the bottom ten and the top ten countries, we used data from only 14 days. This limits our data to only 2 weeks of the entire year for each country, instead of the full 365 days. This could have left out potential changes in trends during the days excluded within the year for each of the countries. In particular, we made the assumption two weeks was sufficient to generalize the trends observed throughout the year, which may require further analyses to confirm.

Having extreme outliers is a data limitation because it can lead to the skewing of certain values, which can lead to the misrepresentation of the trends shown in the graphs of our models. To make our models as accurate as possible, we created multiple models with and without the outliers --each model was differentiated and labeled-- to display trends both inclusive and exclusive of outliers.

When constructing our training model, we used data from 2020, which limits our analysis even further. Given the circumstances of 2020 (i.e., the Novel Coronavirus), our data from the previous years may not be the most accurate model in its predictions because it does not take into account potential outside factors that may impact the data. Thus, it may be argued our predicted values will likely not be representative of the musical trends observed for countries during the year, though this is assuming music drastically changed during this year.

Moving averages are calculated based on past generated data and hence are not good at accounting for future changes that might have an impact. Although we do not use our rolling averages to predict future data, this is ultimately an objective of our analyses. In particular, we wanted to gain more insight into the music happy and less happy countries listen to, so performing the rolling average of these countries would in hindsight tell us more about what “happy” and “less happy” countries are likely to listen to. Additionally, moving averages likely did not capture any meaningful trends if there were any abnormalities in our data.

As it pertains to our linear regression models, we first understand linear regression is limited to linear relationships, which is already an assumption we made during our preregistered analysis about the kind of observation we expected to see. In particular, we assumed there would be a straight-line relationship and that these attributes are independent. Most likely, such a relationship does not exist between happiness score and music features like danceability and valence. Linear regression is further sensitive to outliers, so this impacts the influence certain countries – in our case, the Dominican Republic – have over the rest of the data. Although we tackle this problem by running various regression models with and without this outlier, there are still other quasi-outliers in our model (Canada and Norway which are happy countries with unusually low danceability scores) that may play an influence on our analysis, albeit not as significant as the Dominican Republic. Most importantly, linear regression looks at the relationship between the mean of the dependent variables (happiness scores) and the independent variables (song features like danceability and valence). Suffice to say, means are not a complete description of a single variable, so a linear regression model does not paint a complete picture of the relationship of these variables.

Importing useful packages

In [6]:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from sklearn.linear_model import LinearRegression as lr
```

In [7]:

```
finalsongs2019=pd.read_csv("spotify2019.csv", index_col=0)
finalsongs2019.dropna(inplace=True) topbot=pd.read_csv("happy2019.csv",index_col=0)
```

In [8]:

```
happy2020=pd.read_csv('happiness2020.csv')
happy2020=happy2020[['Country name','Ladder score']]
happy2020['happiness_rank']=happy2020.index.copy()
happy2=happy2020.happiness_rank.to_numpy()
happy2=happy2+1 happy2020['happiness_rank']=happy2
```

In [9]:

```
cols2020= [x.lower() for x in happy2020.columns] cols2020=
[x.replace(" ","_") for x in cols2020]
happy2020.columns=cols2020
```

In [10]:

```
happy2020
```

Out[10]:

	country_name	ladder_score	happiness_rank
0	Finland	7.8087	1
1	Denmark	7.6456	2
2	Switzerland	7.5599	3
3	Iceland	7.5045	4
4	Norway	7.4880	5
...	...	...	...
148	Central African Republic	3.4759	149
149	Rwanda	3.3123	150
150	Zimbabwe	3.2992	151
151	South Sudan	2.8166	152
152	Afghanistan	2.5669	153

153 rows × 3 columns In [11]:

```
finalsongs2020=pd.read_csv("finalsongs2020.csv")
```

In [12]:

```
finalsongs2020.drop(["Unnamed: 0"], axis=1, inplace=True) In [13]:
```

```
finalsongs2020.head()
```

Out[13]:

	country	happiness_score	happiness_rank	date	position	track_name	artist	streams
0	Finland	7.8087	1.0	2020-01-01	1	Hei rakas	BEHM	39681
1	Finland	7.8087	1.0	2020-01-01	2	Pintakaasulla	JVG	31179
2	Finland	7.8087	1.0	2020-01-01	3	Luota Muhun	ibe	30339
3	Finland	7.8087	1.0	2020-01-01	4	Blinding Lights	The Weeknd	30208
4	Finland	7.8087	1.0	2020-01-01	5	Dance Monkey	Tones And I	28966

5 rows × 21 columns

In [14]:

```
countries2020=happy2020.country_name
spotifycountries2020=pd.unique(finalsongs2020['country'])
happy2020=happy2020.loc[happy2020['country_name'].isin(spotifycountries2020)]
happy2020=happy2020.reset_index(drop=True)
```

In [15]:

```
#df with happiness ranking for 2020
happy2020
```

Out[15]:

	country_name	ladder_score	happiness_rank
0	Finland	7.8087	1
1	Denmark	7.6456	2
2	Switzerland	7.5599	3

3	Iceland	7.5045	4
4	Norway	7.4880	5
5	Netherlands	7.4489	6
6	Sweden	7.3535	7
7	New Zealand	7.2996	8
8	Austria	7.2942	9
9	Canada	7.2321	11
10	Dominican Republic	5.6892	68
11	Greece	5.5150	77
12	Malaysia	5.3843	82
13	Vietnam	5.3535	83
14	Indonesia	5.2856	84
15	Turkey	5.1318	93
16	Morocco	5.0948	97
17	South Africa	4.8141	109
18	Egypt	4.1514	138
19	India	3.5733	144

In [16]:

```
#more cleaning and merging
happy2020.rename({"country_name":"country"}, axis="columns", inplace=True)
happy2020.rename({"ladder_score":"happiness_score"}, axis="columns", inplace=Tru e)
spotify2020=happy2020.merge(finalsongs2020, on="country")
spotify2020 Out[16]:
```

	country	happiness_score_x	happiness_rank_x	happiness_score_y	happiness_rank_y	
0	Finland	7.8087	1	7.8087	1.0	20
1	Finland	7.8087	1	7.8087	1.0	20
2	Finland	7.8087	1	7.8087	1.0	20
3	Finland	7.8087	1	7.8087	1.0	20
4	Finland	7.8087	1	7.8087	1.0	20

...	...			...		...	...	...
364961	India	3.5733	144	3.5733	144.0	1		2
364962	India	3.5733	144	3.5733	144.0	1		2
364963	India	3.5733	144	3.5733	144.0	1		2
364964	India	3.5733	144	3.5733	144.0	1		2
364965	India	3.5733	144	3.5733	144.0	1		2

364966 rows × 23 columns

The following code is basically the same routine as the 2017 to get a dataframe with the audiofeatures of the Spotify 2020 songs. We have commented out the code because we saved the .csv file in our first time running it and now pd.read_csv rather than running the cells again

In [17]:

```
happy2020
```

Out[17]:

	country	happiness_score	happiness_rank
0	Finland	7.8087	1
1	Denmark	7.6456	2
2	Switzerland	7.5599	3
3	Iceland	7.5045	4
4	Norway	7.4880	5
5	Netherlands	7.4489	6
6	Sweden	7.3535	7
7	New Zealand	7.2996	8
8	Austria	7.2942	9
9	Canada	7.2321	11
10	Dominican Republic	5.6892	68
11	Greece	5.5150	77
12	Malaysia	5.3843	82
13	Vietnam	5.3535	83
14	Indonesia	5.2856	84
15	Turkey	5.1318	93
16	Morocco	5.0948	97
17	South Africa	4.8141	109
18	Egypt	4.1514	138
19	India	3.5733	144

In [18]:

```
finalsongs2019
```

Out[18]:

	position	streams	date	url	track_name
0	1	33717	2019-01-01	https://open.spotify.com/track/6MWtB6iiXyIwun0...	Wow.
1	2	29651	2019-01-01	https://open.spotify.com/track/25sgk305KZfyuqV...	Sweet but Psycho
2	3	28329	2019-01-01	https://open.spotify.com/track/4RYtaqxjDJUOY2G...	Harmaa Rinne
3	4	23977	2019-01-01	https://open.spotify.com/track/2rPE9A1vEgShuZx...	thank u, next
4	5	22435	2019-01-01	https://open.spotify.com/track/00WO1oBxZcj9aBo...	Tavallinen
...	...	...	...	...	...
363995	46	3427	2019-12-30	https://open.spotify.com/track/3jjujdWJ72nww5e...	Adore You
363996	47	3416	2019-12-30	https://open.spotify.com/track/72Yg5qdIqpTnXrN...	Dames
363997	48	3413	2019-12-30	https://open.spotify.com/track/6XHVuErjQ4XNm6n...	No Guidance (feat. Drake)

363998	49	3407	2019-12-30	https://open.spotify.com/track/7h0d2h0fUmzbs7z...	HIGHEST IN THE ROOM (feat. ROSALÍA & Lil Baby)...
363999	50	3385	2019-12-30	https://open.spotify.com/track/57vxBYXtHMk6H1a...	Heartless

364000 rows × 24 columns

In [19]:

```
finalsongs2020
```

Out[19]:

	country	happiness_score	happiness_rank	date	position	track_name	artist	s
0	Finland	7.8087	1.0	2020-01-01				
1	Finland	7.8087	1.0	2020-01-01				
2	Finland	7.8087	1.0	2020-01-01				

2020-

```
          3   Finland
```

```
                7.8087  1.0
                01-01

2020-

          4   Finland
                7.8087  1.0
                01-01

      ...       ...
      ...       ...           ...

2020-

364961      India
                3.5733  144.0
                12-30

2020-

364962      India
                3.5733  144.0
                12-30

2020-

364963      India
                3.5733  144.0
                12-30

2020-

364964      India
                3.5733  144.0
                12-30

2020-

364965      India
                3.5733  144.0
                12-30
```

364966 rows × 21

columns In

[47]:

1		Hei rakas BEHM
2		Pintakaasulla JVG
3	Luota Muhun	ibe
4	Blinding	The Lights Weeknd

```
from scipy.stats import ttest_ind
from scipy import stats
```

5	Dance Monkey	Tones And I
...	...	...
45	Namo Namo	Amit Trivedi
46	Kalank (Title Track)	Arijit Singh
47	Keh Len De	Kaka, Inder Chahal, Himanshi Khurana
48	Pal Pal Dil Ke Paas- Title Track	Arijit Singh, Parampara Tandon
49	Chidiya	Vilen

In [32]:

```
def permute(input_array):
    # shuffle is inplace, so copy to preserve input
    permuted = input_array.copy()
    np.random.shuffle(permuted)
    return permuted
```

In [36]:

```
def plot_model_line(df, model, is_resampled=True):
    '''
    Takes a dataframe and a fitted model
    Plots a line of best fit to the data
    '''
    if is_resampled:
        color="grey"
        alpha=0.1
    else:
        color="steelblue"
        alpha=0.7
    plt.plot(df['danceability'], df['danceability'] * model.coef_[0] + model.int
ercept_, color=color, alpha=alpha)
```

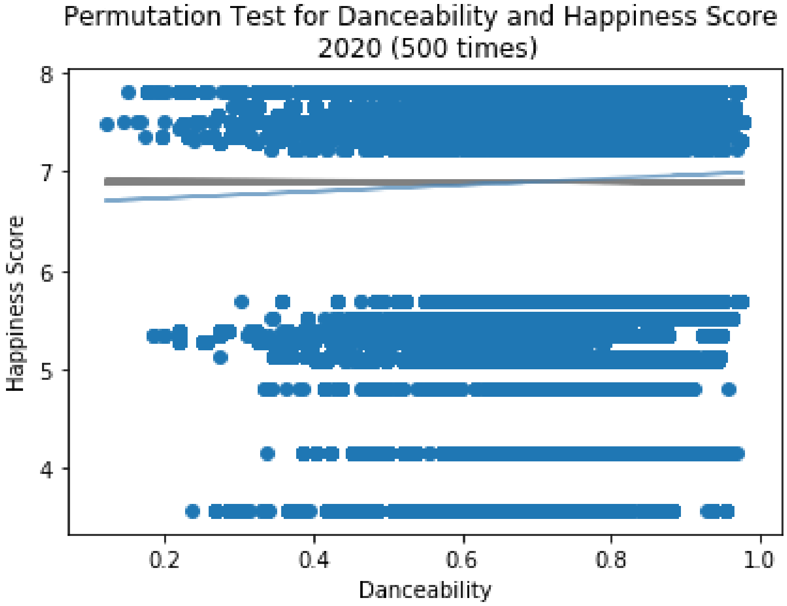
In [54]:

```
# plot true data
plt.scatter(finalsongs2020['danceability'], finalsongs2020['happiness_score'])

# plot 100 iterations of models on permuted data
for i in range(500):
    fake_model = lr().fit(finalsongs2020[["danceability"]], permute(finalsongs2020['happiness_score']))
    plot_model_line(finalsongs2020, fake_model)

# model true data and plot
model = lr().fit(finalsongs2020[["danceability"]], finalsongs2020["happiness_score"])
plot_model_line(finalsongs2020, model, is_resampled=False)
plt.xlabel("Danceability")
plt.ylabel("Happiness Score")
plt.title("Permutation Test for Danceability and Happiness Score  \n 2020 (500 times)")

plt.show()
```



Graph Description Above

We begin by using a permutation test to test for statistical signifance in our data. A permutation test helps us build sampling distribution rather than simply assuming this is the case. In our case, we are shuffling the happiness scores (our y, or dependent variable) without any replacement. The line in gray represents the line of best fit for the actual observed happiness and danceability scores for 2020. The blue line of best fit tells us the slope of the 500 permutations we did.

We want to create a situation in which there is no relationship between our variables (under the null hypothesis) by breaking any relaitonship between x and y, and this is precisely what our graph proves by centering the slopes around 0.

Training Model

After seeing from our exploratory graphs that our top countries seemed to have higher valence scores and slightly higher danceability scores, we believed that performing a training model would help cement whether or not this apparent relationship is just due to the data in 2019, or due to a significant relationship between audio features and happiness. While our graphs seemed to show some sort of relationship between happiness score, valence, and danceability, our heatmaps did not show a significant correlation between these two audio features and happiness score. Therefore, we decided to create a single-variable model for each audio feature and also a multivariable regression model that looked at all of these features.

All of our models use our data from 2019 and then test it on our data from 2020. We chose to create both single-variable and multi-variable models in order to compare the accuracy of their predictions and see if one performed better than the other. It is important to note that we created a set of single- and multi-variable models that included data about our outlier, the DR, and then another set that excluded data from the DR. We weren't sure what the effect of an outlier would be on our model so we chose to compare the model in regards to this aspect as well.

Single variable linear regression with outlier

In [52]:

```
newcol=['pred_danceability', 'pred_energy', 'pred_key',
       'pred_loudness', 'pred_mode', 'pred_speechiness', 'pred_acousticness', 'p
red_instrumentalness',
       'pred_liveness', 'pred_valence', 'pred_tempo', 'pred_duration_ms'] index=0
coeff_determination=[]
predictions=pd.DataFrame()
for feature in ['danceability', 'energy', 'key',        'loudness', 'mode',
'speechiness', 'acousticness', 'instrumentalness',
       'liveness', 'valence', 'tempo', 'duration_ms']:
linear_model=lr()
    linear_model.fit(finalsongs2019[[feature]],finalsongs2019['happiness_score']
)
    preds=linear_model.predict(finalsongs2020[['happiness_score']])

    predictions[newcol[index]]=preds
index=index+1
    coeff_determination.append(round(linear_model.score(finalsongs2020[[feature]
],finalsongs2020['happiness_score']),2))
```

```
    print('Feature: '+feature)
    print('Regression slope of '+feature+' : '+str(round(linear_model.coef_[0],3
)))
    print('r^2 of '+feature+' : '+str(round(linear_model.score(finalsongs2020[[f
eature]],finalsongs2020.happiness_score),3))+'\n')
```

```
Feature: danceability
Regression slope of danceability : 0.373
r^2 of danceability : -0.151

Feature: energy
Regression slope of energy : 0.552
r^2 of energy : -0.168

Feature: key
Regression slope of key : -0.002
r^2 of key : -0.156

Feature: loudness
Regression slope of loudness : 0.017
r^2 of loudness : -0.154

Feature: mode
Regression slope of mode : -0.176 r^2
of mode : -0.159

Feature: speechiness
Regression slope of speechiness : 0.44
r^2 of speechiness : -0.154

Feature: acousticness
Regression slope of acousticness : -0.53
r^2 of acousticness : -0.144

Feature: instrumentalness
Regression slope of instrumentalness : -0.647
r^2 of instrumentalness : -0.156

Feature: liveness
Regression slope of liveness : 0.265
r^2 of liveness : -0.151

Feature: valence
Regression slope of valence : 0.257
r^2 of valence : -0.151

Feature: tempo
Regression slope of tempo : -0.0
r^2 of tempo : -0.156
```

```
Feature: duration_ms
Regression slope of duration_ms : -0.0
r^2 of duration_ms : -0.106
```

Description for Work Above

Here, we fit a linear regression model for each of the audio features we can obtain through the Spotify API. We first fit the model on our data from 2019 with the audio features forming the training set and the happiness score being the target variable. We then used this model to try and predict the happiness scores of countries based on their top song audio features from 2020. Above, we print the regression slope of each feature and the coefficient of determination for each model.

We also append the predicted happiness score to a dataframe named `Predictions` so we can later visually compare the accuracy of the predicitons in relation to the actual happiness score.

In [127]:

```
import statsmodels.formula.api as smf
from scipy import stats
# sm_model = smf.ols('happiness_score ~ danceability', data=finalsongs2019)
# result = sm_model.fit()
# result.summary()
print(stats.linregress(finalsongs2020['valence'].astype(float), finalsongs2020['
happiness_score'].astype(float)).pvalue) 2.778848880638121e-23
```

In [53]:

```
predictions['country']=finalsongs2020['country']
predictions=predictions.merge(happy2020,on='country')
predictions.drop_duplicates(subset=["country"],inplace=True, ignore_index=True)
predictions.rename({"happiness_score":"obs_happiness_score"}, axis="columns", in
place=True)


first_column = predictions.pop('obs_happiness_score')
predictions.insert(0, 'obs_happiness_score', first_column)
first_column = predictions.pop('country') predictions.insert(0,
'country', first_column) predictions.drop(["happiness_rank"],
axis=1, inplace=True)
```

Description for Work Above

The `Predictions` dataframe contains the true happiness scores of each country in the `obs_happiness_score` column. The other columns are the predicted happiness score based on the audiofeature. For example, `pred_danceability` is the predicted happiness scores for each country based on a country's danceability scores for their top songs.

In [54]:

```
predictions
```

Out[54]:

	country	obs_happiness_score	pred_danceability	pred_energy	pred_key	pred_loudness
0	Finland	7.8087	9.044893	10.363976	6.385693	6.641999
1	Denmark	7.6456	8.984134	10.273988	6.386071	6.639170
2	Switzerland	7.5599	9.044893	10.363976	6.385693	6.641999
3	Iceland	7.5045	9.044893	10.363976	6.385693	6.641999
4	Norway	7.4880	9.044893	10.363976	6.385693	6.641999
5	Netherlands	7.4489	9.044893	10.363976	6.385693	6.641999
6	Sweden	7.3535	8.875319	10.112827	6.386749	6.634102
7	New Zealand	7.2996	9.044893	10.363976	6.385693	6.641999
8	Austria	7.2942	8.952208	10.226705	6.386270	6.637683
9	Canada	7.2321	9.044893	10.363976	6.385693	6.641999
10	Dominican Republic	5.6892	8.952208	10.226705	6.386270	6.637683
11	Greece	5.5150	8.190430	9.098463	6.391014	6.602206
12	Malaysia	5.3843	9.044893	10.363976	6.385693	6.641999
13	Vietnam	5.3535	8.141741	9.026352	6.391317	6.599938
14	Indonesia	5.2856	9.044893	10.363976	6.385693	6.641999
15	Turkey	5.1318	8.047678	8.887039	6.391903	6.595558
16	Morocco	5.0948	8.033894	8.866624	6.391989	6.594916
17	South Africa	4.8141	9.044893	10.363976	6.385693	6.641999
18	Egypt	4.1514	9.044893	10.363976	6.385693	6.641999
19	India	3.5733	7.467096	8.027161	6.395519	6.568519

In [55]:

```
coeff=pd.DataFrame(columns=['danceability','energy', 'key', 'loudness', 'mode',
'speechiness', 'acousticness', 'instrumentalness',
                  'liveness', 'valence', 'tempo', 'duration_ms'], index=["Coef
ficient of Determination With Outlier"]) coeff.loc["Coefficient of Determination
With Outlier"] = coeff_determination In [56]:
```

```
coeff
```

Out[56]:

	danceability	energy	key	loudness	mode	speechiness	acousticness	instrum
Coefficient of Determination With Outlier	-0.15	-0.17	-0.16	-0.15	-0.16	-0.15	-0.14	

Description for Work Above

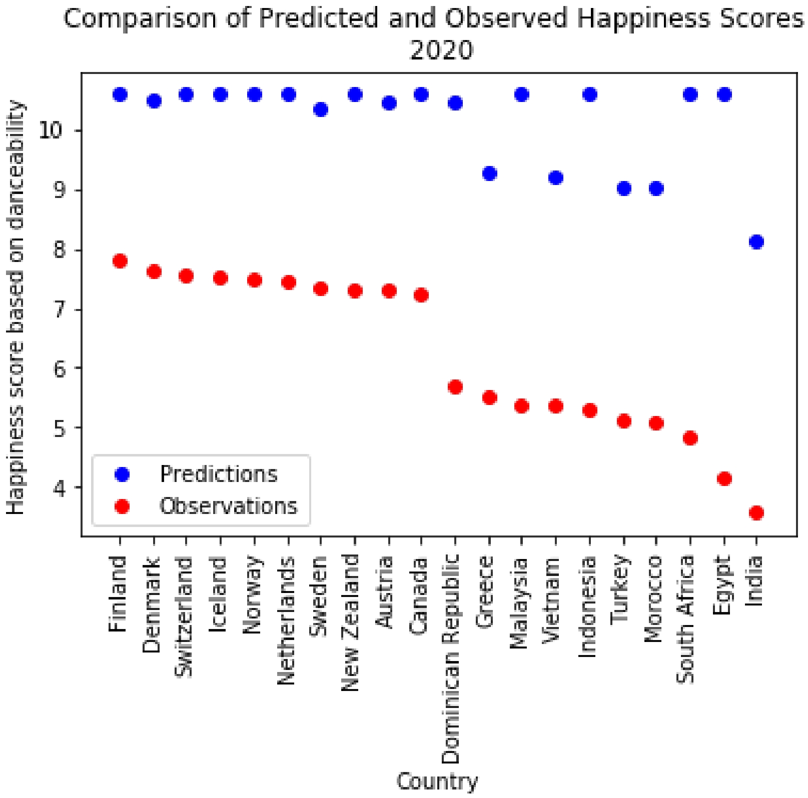
We also saved the coefficient of determination for each model in its own dataframe so we can later compare the coefficient of determinations for the single-variable linear regression model that included the outlier and the model that excluded it.

As you can see, the coefficient of detrmination for each feature is negative. The coefficient of determination basically quantifies how well the variance in happiness score can be explained by an audio feature. A negative coefficent of determination says that 1) the model is not doing a good job at explaining the variance and 2) that a horizontal line would fit better than the model does. Oddly `duration_ms` (the length of a song) has the highest coefficient of determination. However, this coefficient is still extremely small and doesn't mean that duration score is a predictor for a country's happiness score. Further, most songs are usually the same length so it wouldn't make sense if the length of a song is able to predict happiness score since happiness scores vary while duration does not.

As we can see with these determination scores and the following graphs, the model we created for a single variable is not very accurate.

In [87]:

```
plt.scatter(predictions['country'],predictions['pred_danceability'],c='blue', la
bel='Predictions')
plt.scatter(predictions['country'],predictions['obs_happiness_score'],c='red', l
abel='Observations')
plt.legend()
plt.title('Comparison of Predicted and Observed Happiness Scores  \n 2020')
plt.xlabel('Country')
plt.ylabel('Happiness score based on danceability' )
plt.xticks(rotation="vertical")
plt.show()
```



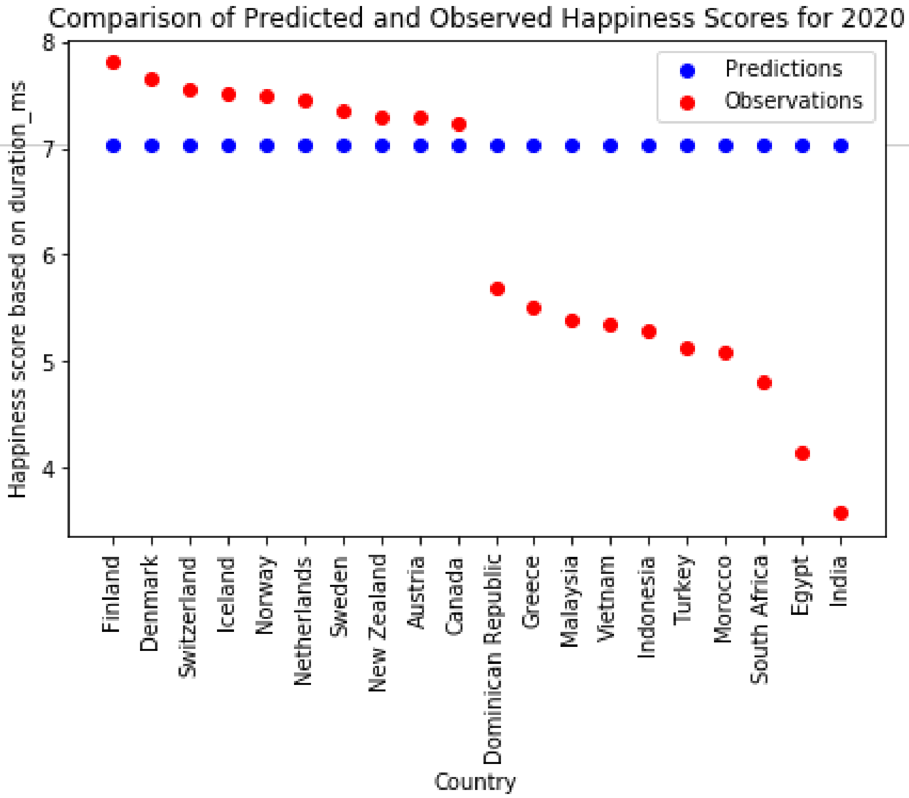
Description for Work Above

Here we graph the happiness scores that were predicted based on a song danceability scores. The dots in red represent the true happiness scores per country while the dots in blue represent the predicted happiness scores.

We can see that the happiness scores were predicted to range between around 7.8 and 9 for both the top and bottom 10 countries. However, the true happiness scores range between around 3 and 8. Further, since we look at top and bottom 10 countries, there is a distinct split between the happiness scores for the top and bottom 10 countries. The predicted values do seem to decrease/vary a bit starting with the Dominican of Republic. However, overall this is not enough to say that danceability predicted the happiness scores well.

In [88]:

```
plt.scatter(predictions['country'],predictions['pred_duration_ms'],c='blue', lab
el='Predictions')
plt.scatter(predictions['country'],predictions['obs_happiness_score'],c='red', l
abel='Observations') plt.legend() plt.xlabel('Country')
plt.ylabel('Happiness score based on duration_ms') plt.tight_layout()
plt.title('Comparison of Predicted and Observed Happiness Scores for 2020')
plt.xticks(rotation="vertical") plt.show()
```

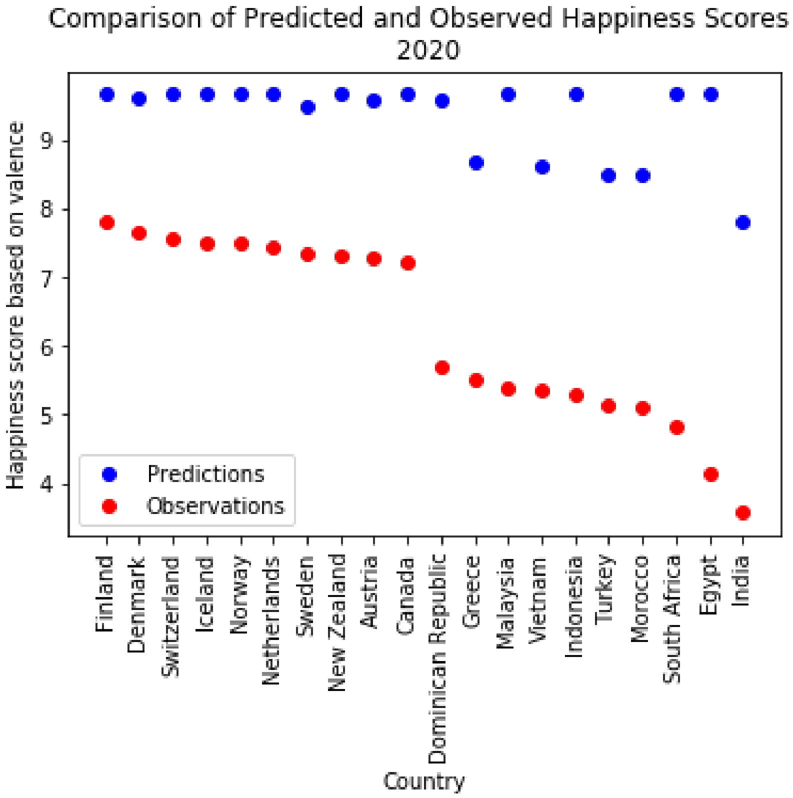


Description for Work Above

Since our duration model did seem to have the highest coefficient of determination, we decided to examine its predicted happiness scores in comparison to the true happiness scores. However, we can see that the predicted happiness scores form a linear line, likely due to the fact that the length of a song does not vary that much. The predicted happiness score is around 7 for every country. We see that duration is not a good predictor for a country's happiness score.

In [89]:

```
plt.scatter(predictions['country'],predictions['pred_valence'],c='blue', label='Predictions')
plt.scatter(predictions['country'],predictions['obs_happiness_score'],c='red', label='Observations')
plt.legend()
plt.title('Comparison of Predicted and Observed Happiness Scores  \n 2020')
plt.xlabel('Country')
plt.ylabel('Happiness score based on valence' )
plt.xticks(rotation="vertical")
plt.show()
```



Description for Work Above

Here we graph the predicted happiness scores that were based on valence scores. This graph is very similar to our graph for danceability. The same conclusion can be reached; valence is not a good predictor of happiness and therefore there doesn't seem to be a relationship between valence and a country's happiness.

Multivariable Linear Regression

We decided to also create a multivariable linear regression model to see if looking at multiple audio features, in comparison to individual audio features, would yield better predictions. Multivariable Linear Regression with Outlier

In [60]:

```
multi_lin_model=lr()
multi_lin_model.fit(finalsongs2019[['danceability','energy', 'valence', 'tempo']
],finalsongs2019['happiness_score'])

for x, feature in enumerate(['danceability','energy', 'valence', 'tempo']):
print('{} coefficient: {:.2f}'.format(feature, multi_lin_model.coef_[x]))

print('r^2 of the model: '+str(round((multi_lin_model.score(finalsongs2019[['dan
ceability','energy', 'valence', 'tempo']],finalsongs2019['happiness_score'])),2)
))
```

```
danceability coefficient: 0.24
energy coefficient: 0.51
valence coefficient: 0.06
tempo coefficient: -0.00 r^2
of the model: 0.01
```

Description for Work Above

We created a multivariable linear regression model by fitting a model based on only the danceability, valence, energy, and tempo features of a song. We chose to fit the model on these features as the Spotify API designates these features as the ones that make up the 'mood' of a song. Our target variable was again a country's happiness score.

The model itself has a coefficient of determination of 0.01, which means that around 1% of the variation in happiness scores can be explained by a song's danceability, valence, energy, and tempo features. This is higher than any of the coefficients of determination we saw from the single variable regression models, however, the model hasn't been used to predict 2020 yet, which better implies whether the model is good or not. This is what we do next.

In [61]:

```
clist=pd.unique(finalsongs2020.country) mult_pred_happy=[] for
country in clist:
country_pred=finalsongs2020[finalsongs2020['country']==country]
mean_scores=pd.DataFrame(country_pred.mean()).T
    mult_pred=multi_lin_model.predict(mean_scores[['danceability','energy', 'val
ence', 'tempo']])
```

```
    mult_pred_happy.append(mult_pred[0])
    coeffnum=round(multi_lin_model.score(finalsongs2020[['danceability','energy'
, 'valence', 'tempo']],finalsongs2020['happiness_score']),2) In [62]:
```

```
coeffnum
```

Out[62]:

```
-0.16
```

Description for Work Above

Here, we use the multivariable model that we created to predict the happiness scores for countries in 2020. The model has a coefficient of determination of -0.16. Again a negative coefficient of determination implies that the model is a poor fit for countries' actual happiness scores. A coefficient of determination of -0.16 is around the same as the coefficents of determination for the single-variable models. Therefore, this multivariable model doesn't do a better job at predicting countries' happiness scores than the singlevariable models.

In [63]:

```
multhap=pd.DataFrame()
multhap['country']=clist
multhap['pred_happiness_score']=mult_pred_happy
multhap['obs_happiness_score']=predictions['obs_happiness_score']
```

In [64]:

```
multhap
```

Out[64]:

	country	pred_happiness_score	obs_happiness_score
0	Finland	6.410400	7.8087
1	Denmark	6.414613	7.6456
2	Switzerland	6.406901	7.5599
3	Iceland	6.376591	7.5045
4	Norway	6.376803	7.4880
5	Netherlands	6.404344	7.4489
6	Sweden	6.389398	7.3535
7	New Zealand	6.390633	7.2996
8	Austria	6.414807	7.2942

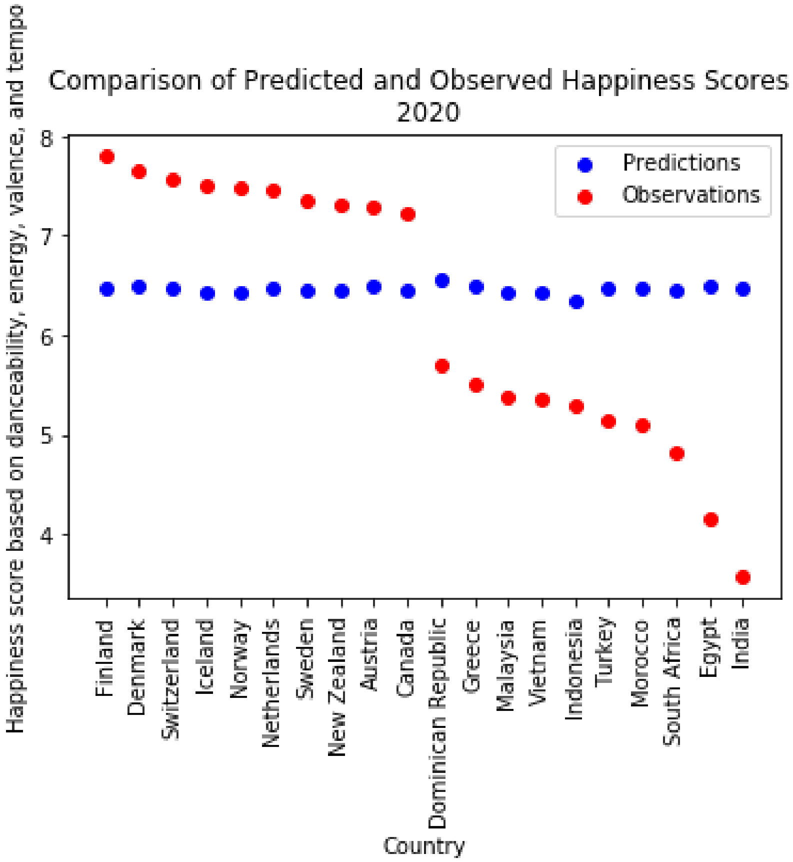
9	Canada	6.390204	7.2321
10	Dominican Republic	6.461349	5.6892
11	Greece	6.416449	5.5150
12	Malaysia	6.374380	5.3843
13	Vietnam	6.377937	5.3535
14	Indonesia	6.323483	5.2856
15	Turkey	6.405384	5.1318
16	Morocco	6.410890	5.0948
17	South Africa	6.391208	4.8141
18	Egypt	6.414743	4.1514
19	India	6.411903	3.5733

Description for Work Above

As we did with our `predictions` dataset, here we compiled the predicted happiness scores that were calculated using the model and the true happiness score for each country. This will help us visualize the discrepancies between the model's outputs and the correct outputs.

In [95]:

```
plt.scatter(multhap['country'],multhap['pred_happiness_score'],c='blue', label='Predictions')
plt.scatter(multhap['country'],multhap['obs_happiness_score'],c='red', label='Observations')
plt.legend()
plt.title('Comparison of Predicted and Observed Happiness Scores  \n 2020')
plt.xlabel('Country')
plt.ylabel('Happiness score based on danceability, energy, valence, and tempo' )
plt.xticks(rotation="vertical")
plt.show()
```



Description for Work Above

Above, we plot the happiness scores that our multivariable linear regression model predicts for countries in 2020. The predicted happiness scores are all around 6.4 and they seem to form a linear graph. However, again the predicted and actual observations are very different and emphasize that even with a multivariable graph, the relationship between songs and happiness score is not strong enough to create a model.

Single-Variable Linear Regression Model without the Outlier

As we said before, we also wanted to create the single- and multivariable linear regression models with data that excluded the Dominican Republic since the DR did seem to be an outlier in terms of its valence score. Here we create the single-variable model is fitted on 2019 data that excludes the DR.

In [66]:

```
finalsongs2019nooutlier=finalsongs2019[finalsongs2019["country"]!="Dominican Rep
ublic"] finalsongs2019nooutlier.reset_index(inplace=True, drop=True)
```

In [100]:

```
newcol=['pred_danceability', 'pred_energy', 'pred_key',
       'pred_loudness', 'pred_mode', 'pred_speechiness', 'pred_acousticness', 'p
red_instrumentalness',
       'pred_liveness', 'pred_valence', 'pred_tempo', 'pred_duration_ms'] index=0
coeff_determination=[]
predictions=pd.DataFrame()
for feature in ['danceability', 'energy', 'key',         'loudness', 'mode',
'speechiness', 'acousticness', 'instrumentalness',
       'liveness', 'valence', 'tempo', 'duration_ms']:
    linear_model=lr()
    linear_model.fit(finalsongs2019nooutlier[[feature]],finalsongs2019nooutlier[
'happiness_score'])
    preds=linear_model.predict(finalsongs2020[['happiness_score']])

    predictions[newcol[index]]=preds
index=index+1
    coeff_determination.append(round(linear_model.score(finalsongs2020[[feature]
],finalsongs2020['happiness_score']),2))

    print('Feature: '+feature)
    print('Regression slope of '+feature+' : '+str(round(linear_model.coef_[0],3
)))
    print('r^2 of '+feature+' : '+str(round(linear_model.score(finalsongs2020[[f
eature]],finalsongs2020.happiness_score),3))+'\n')
```

```
Feature: danceability
Regression slope of danceability : 0.585
r^2 of danceability : -0.12

Feature: energy
Regression slope of energy : 0.766
r^2 of energy : -0.144

Feature: key
Regression slope of key : 0.001
r^2 of key : -0.125
```

```
Feature: loudness
Regression slope of loudness : 0.034
r^2 of loudness : -0.123

Feature: mode
Regression slope of mode : -0.187 r^2
of mode : -0.131

Feature: speechiness
Regression slope of speechiness : 0.622
r^2 of speechiness : -0.124

Feature: acousticness
Regression slope of acousticness : -0.564
r^2 of acousticness : -0.114

Feature: instrumentalness
Regression slope of instrumentalness : -0.724
r^2 of instrumentalness : -0.126

Feature: liveness
Regression slope of liveness : 0.28
r^2 of liveness : -0.122

Feature: valence
Regression slope of valence : 0.441
r^2 of valence : -0.12

Feature: tempo
Regression slope of tempo : -0.0
r^2 of tempo : -0.126

Feature: duration_ms
Regression slope of duration_ms : -0.0
r^2 of duration_ms : -0.083
```

Description for Work Above

We fit the model on our data from 2019 and use the model to predict the happiness scores for 2020. We print the regression slope and the coefficient of determination for each model.

In [68]:

```
predictions['country']=finalsongs2020['country']
predictions=predictions.merge(happy2020,on='country')
predictions.drop_duplicates(subset=["country"],inplace=True, ignore_index=True)
predictions.rename({"happiness_score":"obs_happiness_score"}, axis="columns", inplace=True)
```

```
first_column = predictions.pop('obs_happiness_score')
predictions.insert(0, 'obs_happiness_score', first_column)
first_column = predictions.pop('country') predictions.insert(0,
'country', first_column) predictions.drop(["happiness_rank"],
axis=1, inplace=True)
```

In [69]:

```
predictions
```

Out[69]:

	country	obs_happiness_score	pred_danceability	pred_energy	pred_key	pred_loudness
0	Finland	7.8087	10.608841	11.958781	6.444277	6.939227
1	Denmark	7.6456	10.513488	11.833907	6.444137	6.933653
2	Switzerland	7.5599	10.608841	11.958781	6.444277	6.939227
3	Iceland	7.5045	10.608841	11.958781	6.444277	6.939227
4	Norway	7.4880	10.608841	11.958781	6.444277	6.939227
5	Netherlands	7.4489	10.608841	11.958781	6.444277	6.939227
6	Sweden	7.3535	10.342717	11.610267	6.443885	6.923670
7	New Zealand	7.2996	10.608841	11.958781	6.444277	6.939227
8	Austria	7.2942	10.463385	11.768293	6.444063	6.930724
9	Canada	7.2321	10.608841	11.958781	6.444277	6.939227
10	Dominican Republic	5.6892	10.463385	11.768293	6.444063	6.930724
11	Greece	5.5150	9.267870	10.202659	6.442299	6.860835
12	Malaysia	5.3843	10.608841	11.958781	6.444277	6.939227
13	Vietnam	5.3535	9.191459	10.102591	6.442187	6.856368
14	Indonesia	5.2856	10.608841	11.958781	6.444277	6.939227
15	Turkey	5.1318	9.043839	9.909270	6.441969	6.847739
16	Morocco	5.0948	9.022208	9.880942	6.441937	6.846474
17	South Africa	4.8141	10.608841	11.958781	6.444277	6.939227
18	Egypt	4.1514	10.608841	11.958781	6.444277	6.939227
19	India	3.5733	8.132690	8.716037	6.440625	6.794474

Description for Work Above

As we did when we fit single-variable linear regression models for the 2019 data with DR data included, we make another dataframe that compares the happiness scores that each single-variable model outputted. This will help us visually compare the model's predictions with the true happiness scores later.

In [70]:

```
coeffwithoutoutlier=pd.DataFrame(columns=['danceability','energy', 'key', 'loudn
ess', 'mode', 'speechiness', 'acousticness', 'instrumentalness',
                    'liveness', 'valence', 'tempo', 'duration_ms'], index=["Coef
ficient of Determination Without Outlier"])
coeffwithoutoutlier.loc["Coefficient of Determination Without Outlier"] = coeff_

determination coeff=coeff.append(coeffwithoutoutlier) In [71]:
```

```
coeff
```

Out[71]:

	danceability	energy	key	loudness	mode	speechiness	acousticness	instrum
Coefficient of Determination With Outlier	-0.15	-0.17	-0.16	-0.15	-0.16	-0.15	-0.14	
Coefficient of Determination Without Outlier	-0.12	-0.14	-0.13	-0.12	-0.13	-0.12	-0.11	

Description for Work Above

Above, we compare the coefficient of determination for all of the single-variable linear regression models. It appears that when we remove the Dominican Republic, our outlier, from the data used to train all the models, the coefficient of determination improves by 0.03 for each respect audio feature. However, all the coefficients of determination for each model are still negative, showing that removing the outlier did not improve the single-variable enough for it to now be significant.

We will now see if this pattern continues for the multivariable models.

In [72]:

```
multi_lin_model=lr()
multi_lin_model.fit(finalsongs2019nooutlier[['danceability','energy', 'valence',
'tempo']],finalsongs2019nooutlier['happiness_score'])

for x, feature in enumerate(['danceability','energy', 'valence', 'tempo']):
print('{} coefficient: {:.2f}'.format(feature, multi_lin_model.coef_[x]))

print('r^2 of the model: '+str(round((multi_lin_model.score(finalsongs2019nooutl
ier[['danceability','energy', 'valence', 'tempo']],finalsongs2019nooutlier['happ
iness_score'])),2)))
```

```
danceability coefficient: 0.38
energy coefficient: 0.66
valence coefficient: 0.17
tempo coefficient: -0.00 r^2
of the model: 0.02
```

Description for Work Above

We fit a multivariable linear regression for our 2019 data that excluded the Dominican Republic. This model has a coefficient of determination of 0.02, which means that around 2% of the variance in happiness scores in 2019 can be explained by a country's top songs, specifically the songs' danceability, energy, acousticness, liveness, valence, tempo, and duration. However, this is the score for the model when it predicts data in 2019. We want to know whether this model is able to predict the happiness scores for countries in a different year.

Now we use the model to predict happiness scores in 2020.

In [73]:

```
clist=pd.unique(finalsongs2020.country)

mult_pred_happy=[]
for country in clist:
    country_pred=finalsongs2020[finalsongs2020['country']==country]
mean_scores=pd.DataFrame(country_pred.mean()).T
    mult_pred=multi_lin_model.predict(mean_scores[['danceability','energy', 'val
ence', 'tempo']])
    mult_pred_happy.append(mult_pred[0])
    coeff=round(multi_lin_model.score(finalsongs2020[['danceability','energy', '
valence', 'tempo']],finalsongs2020['happiness_score']),2)
```

In [74]:

```
coeff
```

Out[74]:

```
-0.14
```

Description for Work Above

After running our multivariable model on the data from 2020, we get a coefficient of determination of -0.14. This is still a negative coefficient, meaning that this model does a very poor job at predicting a country's happiness scores. -0.14 is 0.02 higher than the coefficient of determinations for the multivariable model that was fitted on 2019 that included the outlier. However, as it is still a negative number and was only improved by 0.02 it is not significant enough to say that there is a significant relationship between happiness scores and songs features.

In [75]:

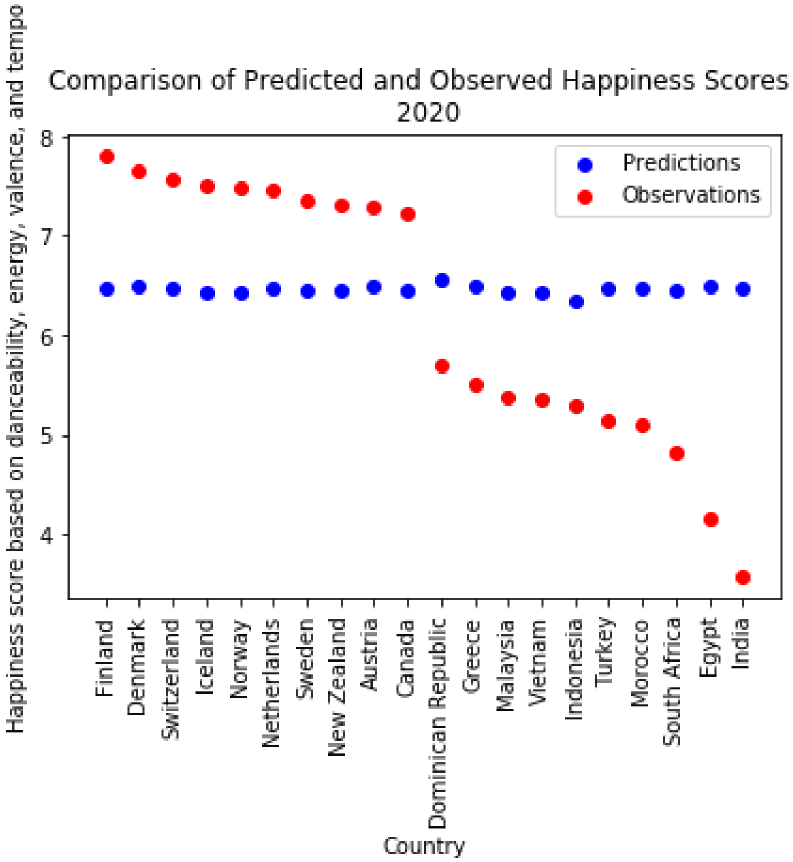
```
multhap=pd.DataFrame()
multhap['country']=clist
multhap['pred_happiness_score']=mult_pred_happy
multhap['obs_happiness_score']=happy2020['happiness_score']
multhap
```

Out[75]:

	country	pred_happiness_score	obs_happiness_score
0	Finland	6.477402	7.8087
1	Denmark	6.486772	7.6456
2	Switzerland	6.474490	7.5599
3	Iceland	6.433200	7.5045
4	Norway	6.428044	7.4880
5	Netherlands	6.468890	7.4489
6	Sweden	6.447135	7.3535
7	New Zealand	6.452899	7.2996
8	Austria	6.484584	7.2942
9	Canada	6.451473	7.2321
10	Dominican Republic	6.559552	5.6892
11	Greece	6.486898	5.5150
12	Malaysia	6.427739	5.3843
13	Vietnam	6.430605	5.3535
14	Indonesia	6.351562	5.2856
15	Turkey	6.470421	5.1318
16	Morocco	6.481248	5.0948
17	South Africa	6.451454	4.8141
18	Egypt	6.481534	4.1514
19	India	6.480973	3.5733

In [99]:

```
plt.scatter(multhap['country'],multhap['pred_happiness_score'],c='blue', label='
Predictions')
plt.scatter(multhap['country'],multhap['obs_happiness_score'],c='red', label='Ob
servations')
plt.legend()
plt.title('Comparison of Predicted and Observed Happiness Scores \n 2020')
plt.xlabel('Country')
plt.ylabel('Happiness score based on danceability, energy, valence, and tempo' )
plt.xticks(rotation="vertical")
plt.show()
```



Just by looking at these graphs, there does not seem to be any clear relationship between the song features and the happiness score of a country. With that in mind, multiple regression seems to be a better predictor than single linear regression.

IX Sources

Source Code

GitHub Repository: [https://github.com/Albina-C/INFO-2950-Project (https://github.com/Albina-C/INFO2950-Project)](https://github.com/Albina-C/INFO2950-Project)

Appendix

[Exploratory Analysis (./Exploratory_analysis.ipynb)](./Exploratory_analysis.ipynb)
[Web-Scraping Notebook (./Dataset-Creation.ipynb)](./Dataset-Creation.ipynb)

Acknowledgements

Spotipy: https://spotipy.readthedocs.io/en/2.18.0/ (https://spotipy.readthedocs.io/en/2.18.0/)
Cloudscraper: https://pypi.org/project/cloudscraper/ (https://pypi.org/project/cloudscraper/)
Web-Scraping Code Inspiration: https://gist.github.com/hktosun/d4f98488cb8f005214acd12296506f48 (https://gist.github.com/hktosun/d4f98488cb8f005214acd12296506f48)
Harvard Medical School Study: https://www.health.harvard.edu/staying-healthy/music-and-health (https://www.health.harvard.edu/staying-healthy/music-and-health) American Psychological Association Study (Hunter, et al): https://www.utm.utoronto.ca/~w3psygs/HunterEtAl2010.pdf (https://www.utm.utoronto.ca/~w3psygs/HunterEtAl2010.pdf)

Turn Up the Music!

Let's make a playlist of each country's happiest songs (according to valence)! We want you to be caught up with the music trends so let's use 2020 data.

In [ ]:

```
import spotipy
from spotipy.oauth2 import SpotifyClientCredentials

cid='c10b42de14134edfb7e9cafa42fc48a2'
secret='b41981d56a924e65a079138f9272e8de'
client_credentials_manager = SpotifyClientCredentials(client_id=cid, client_secr
et=secret)
sp = spotipy.Spotify(client_credentials_manager
=
client_credentials_manager)
```

In [ ]:

```
groups=finalsongs2020.groupby("country")
```

In [ ]:

```
max_urls=[]
min_urls=[]
for country, group in groups:
    max_valueid=group.valence.idxmax()
    max_urls.append(group.loc[max_valueid,'id'])
    min_valueid=group.valence.idxmin()
    min_urls.append(group.loc[min_valueid,'id'])
```

In [ ]:

```
begin="https://open.spotify.com/track/"
maxu=[]
minu=[]
for url in range(len(max_urls)):
    maxu.append(begin+max_urls[url])
for url in range(len(min_urls)):
    minu.append(begin+min_urls[url])
```

In [ ]:

```
maxu
```

In [ ]:

```
minu
```

Link to Happy Playlist: https://open.spotify.com/playlist/2BrmDGLVRlD7r2Cf8culSw?si=50e8700bc810496f (https://open.spotify.com/playlist/2BrmDGLVRlD7r2Cf8culSw?si=50e8700bc810496f)

Link to Sad Playlist: https://open.spotify.com/playlist/4ntdqpIQ5Eus34Ds9gDQNm?si=d4965381398e45e3 (https://open.spotify.com/playlist/4ntdqpIQ5Eus34Ds9gDQNm?si=d4965381398e45e3)