

They See Us Roller Coastin'

Summary Sheet

Problem

Our group was given the problem of designing a roller coaster ranking system/algorithm that numerically ranked roller coasters using purely objective data. We also had to determine the world's 10 best roller coasters using the algorithm and compare our list with at least two other lists found on the internet. In addition, we were to conceptualize an app that used user preferences and the ranking system to find potential roller coasters, and write a press release describing the algorithm, app, and results.

Summary

Our goal was to design our coaster ranking algorithm to be as flexible and comprehensive as possible. Based on research we conducted on other online ranking systems, we implemented 5 different parameters for our algorithm: coaster length, duration, maximum speed, maximum drop height, and number of inversions. Of these parameters, internet research failed to adequately complete the drop height data set. As such, we used kinematic equations to define an equation to extrapolate drop height from speed.

For our scoring method, we created a subscore between 0 and 1 for each of the five parameters using a normal distribution curve. Then, we weighted each subscore depending on how much that parameter's importance was valued. Subscore calculation utilized user-defined ideal values and tolerance for each parameter, with the ideal being the optimal value, and the tolerance being used to define the slope of the parameter's distribution curve, i.e., how willing one is to stray from their preferences.

For our app, the user would input a value from 1-10 for each parameter that denotes how much they care about that parameter, which determine how much the parameter is weighted. There would also be an advanced option that would allow the user to set their ideal values and tolerances for each parameter. The tolerance would be a value 1-10, 10 meaning high tolerance and 1 meaning low tolerance. If the user does not input their ideal values and tolerances, the app will use default values.

For our top ten list, we determined proper weights, set the ideal values to be maximum values, and used default tolerances for each parameter. Our top ten were, in order of best to worst: Kingda Ka at Six Flags Great Adventure New Jersey, Top Thrill Dragster at Cedar Point, Formula Rossa at Ferrari World Abu Dhabi, Ultimate at Lightwater Valley, Red Force at Ferrari Land, Smiler at Alton Towers, Steel Dragon at Nagashima Spa Land, Fury 325 at Carowinds, Superman: Escape from Krypton at Six Flags Magic Mountain, and Tower of Terror II at Dreamworld

News Release

Our application helps users find the roller coasters that they would be most interested in. Our special algorithm uses five parameters: Length of the ride, Duration of the ride, Maximum Drop Height, Maximum Speed, and Number of Inversions.

Users would input values 1-10 for each of these parameters to determine how much they care about the parameter. These values would determine how much each parameter is weighted in the final score. For example, someone who doesn't care at all about inversions, really enjoys a high drop height, would prefer a fast ride, and does not care how long the ride is might enter the following values: ten for drop height, five for the duration of the ride and for length, eight for max speed, and one for inversions.

There would be more advanced options for users who want to further personalize their rankings. The advanced options would show two more values for each parameter: the user's ideal value and their tolerance. The ideal value is the user's preferred value for that parameter. For example, if they want coaster to have a max speed of 60 mph, then they would list that as their ideal value for maximum speed. The tolerance is how willing the user is to stray from their ideal value for that parameter. This would be a number from 1-10, from low tolerance to high tolerance. If the user selects a low tolerance for speed, then a roller coaster's speed score would decrease quickly as it got further from their ideal value. If the user selects a high tolerance, then a roller coaster's speed score would decrease slowly as it got further from their ideal value. By default, the ideal value for each parameter is set to $\frac{3}{4}$ of the way towards the maximum for that parameter, while the tolerance levels would be set to 5.

For our list of the top 10 roller coasters in the world, we set the ideal to maximum possible and used default tolerance values for each parameter. We set weights for categories as such: 8 for speed, 6 for inversion, 7 for duration, 2 for length, and 8 for drop height.

Note : A link to the application would appear here

Top Ten List

- 1. Kingda Ka at Six Flags Great Adventure New Jersey
- 2. Top Thrill Dragster at Cedar Point
- 3. Formula Rossa at Ferrari World Abu Dhabi
- 4. Ultimate at Lightwater Valley
- 5. Red Force at Ferrari Land
- 6. Smiler at Alton Towers
- 7. Steel Dragon at Nagashima Spa Land
- 8. Fury 325 at Carowinds
- 9. Superman: Escape from Krypton at Six Flags Magic Mountain
- 10. Tower of Terror II at Dreamworld

Solution

Assumptions

- All roller coasters are safe and never break down (for our purposes)
 - According to online science websites “the chance of being injured on a fixed-site ride at one of those parks is 1 in 17 million” [2]
- Aesthetics of roller coasters do not affect roller coaster ratings
 - Aesthetics are nigh impossible to quantify, and most roller coaster riders care about riding on a roller coaster, as they can watch a video of riding on a roller coaster for free
 - Originality is not important for determining a roller coaster’s rate in and of itself
 - Roller coaster riders are effectively “blind” because they don’t care about what they see
- Price does not matter for determining the top ten roller coasters
 - Generally, the cost of riding roller coasters is fairly standard, and most roller coasters are at amusement parks which have one rate of admission which allows riders to ride many roller coasters
- The geographical location and name of a roller coaster does not affect the rating of the roller coaster
 - The statistics we are using are independent of location and name, and though location and name could affect aesthetics, we are not considering aesthetics for our ratings (see above point)
- Height/health requirements and wait times are irrelevant, as our rating targets the roller coaster itself and how the people that actually ride the roller coaster enjoy the roller coaster
- Construction materials do not matter
 - Certain techniques can be used to give wood very similar properties to steel, or wood can be coated with steel to give a smoother ride [6]
 - In addition, preferred construction type is highly subjective, and so we chose for it not to have bearing in our model
- Type of roller coaster does not have a bearing by the same logic as construction not having a bearing
- Correlations in data and classical mechanics (physics) can be used to fill holes in data
 - There is no friction
- Highest speed will occur after the highest drop
 - This comes from physics and classical mechanics [2]
 - We are assuming that roller coaster drop uses gravity, because that is what our knowledge of physics can handle
- At the top of the roller coaster the speed is ~0
- All roller coasters have approximately the same preferences (in the context of rating the top ten roller coasters)
 - Though this is not actually true, we are trying to come up with a general method for rating, so we consider the “average” rider to be representative of every rider

- For the app design, riders will be able to choose their favorite roller coaster(s) based off of the criteria speed, duration, length, drop height, and number of inversions.
- Roller coasters go both up and down

Datums/Data

Rationale for Excluding Data

- Name: Has no bearing on quality of roller coaster (see Assumptions)
- Location: Has no bearing on quality of roller coaster (see Assumptions)
- Construction: While there exist differences between steel and wooden roller coasters, the preferred construction type is heavily subjective, and so will not have bearing in our model (see Assumptions)
- Type: We excluded this by the same logic that we excluded construction (see Assumptions)
- Status: All the roller coasters in the Excel sheet are operating, so there is no need to take status into account in the model
- Year/Date Opened: While the year opened may affect aspects of the roller coaster such as safety and aesthetics, based on our assumptions, none of these elements apply to our rating.
- Height: Height and drop height are heavily related, so we are not including height. The effect of being high up in the air will not affect the quality of the roller coaster according to our model. (see Assumptions)

Rationale for Including Data

- Speed: The speed of the roller coaster definitely affects the enjoyment of the passengers
- Length: This is a factor in how long the passengers get to enjoy the roller coaster
- Number of Inversions: This affects how much force the passengers feel on themselves
- Drop Height: This affects the time that the passengers feel weightless
 - One of the features that roller coaster riders enjoy is air time. Though we could not find air time directly for many roller coasters, air time is related to drop height
- Duration: This is a factor in how long the passengers get to enjoy the roller coaster
- G-Force: This affects how much force the passengers feel on themselves
- Vertical Angle: This affects how weightless the passengers will feel on the initial drop

Note: See Appendix for the big data table with statistics for all the roller coasters.

Since the data had many holes, we filled it in using various methods:

Internet Research

The first way we attempted to fill the holes in the data was through internet research. There were several online databases that had stats on roller coasters consistent with those given. We went through

these databases to find any data cells that were missing. We were able to fill in a considerable amount of values through this method, especially drop height. (See Appendix for databases)

To fill in the missing duration values for each coaster, we watch POV videos of that coaster. With these videos, we were able to find the duration for every roller coaster. (see appendix for youtube channels)

Extrapolation

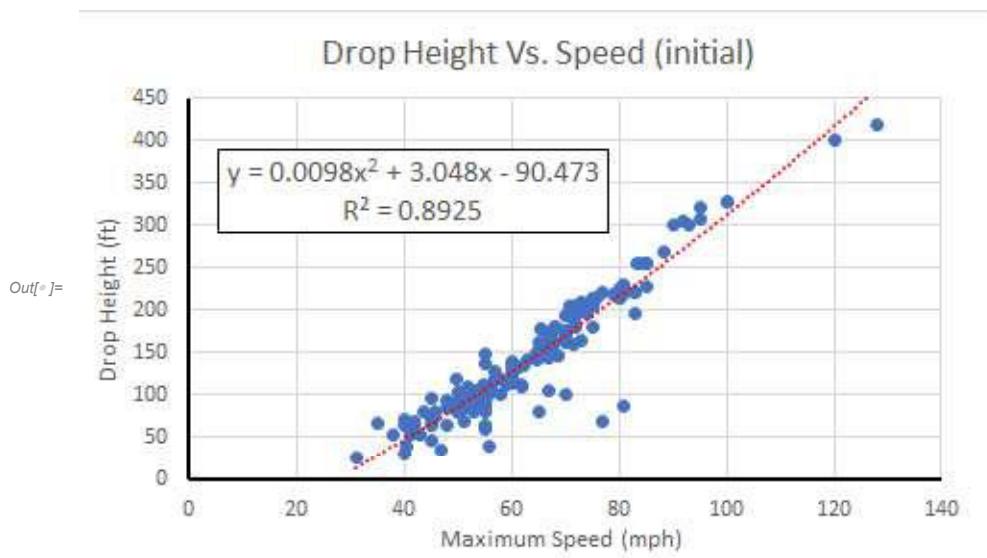
At this point, each column of values was either light or dense. Dense meant that almost all (>90%) of the values for the parameter were filled, and light meant that a considerable amount of the values for the parameter were missing. The light parameters were drop height, vertical angle, and g-force. The dense parameters were height, speed, length, inversions, amount of inversions, and duration. We did not consider the other data to be important in determining our rating (see assumptions). The goal of our extrapolation was to find a relationship between a dense parameter and a light parameter.

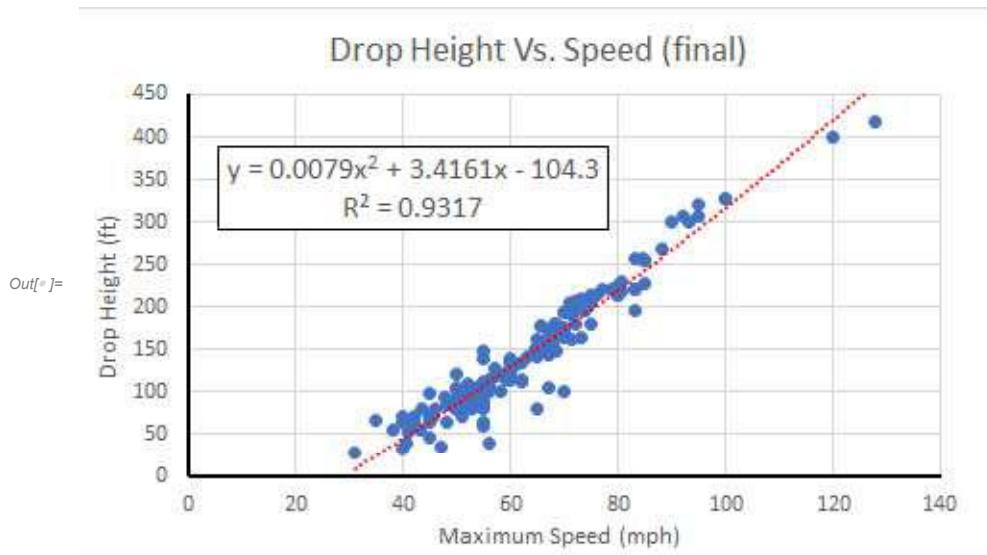
Drop Height and Speed

Since the highest speed occurs after the largest drop (see assumptions), there is a relationship between drop height and speed. During large drops on roller coasters, gravity is the primary force accelerating the cart. Using kinetic and potential energy equations (from physics), we can show that $v^2 \sim h$ where v is the speed and h is the drop height, so there is a quadratic relationship between v and h.

Using Microsoft Excel we created a scatter plot of drop height vs speed. We then used Excel to create a quadratic curve of best fit for the scatter plot.

The curve had equation $0.0098x^2 + 3.048x - 90.473$ and an R^2 value of 0.8925. We made the model stronger by removing 2 outlier points, which gave us the equation $0.0079x^2 + 3.4161x - 104.3$ and an R^2 value of 0.9317. The graph of both relationships are shown below.



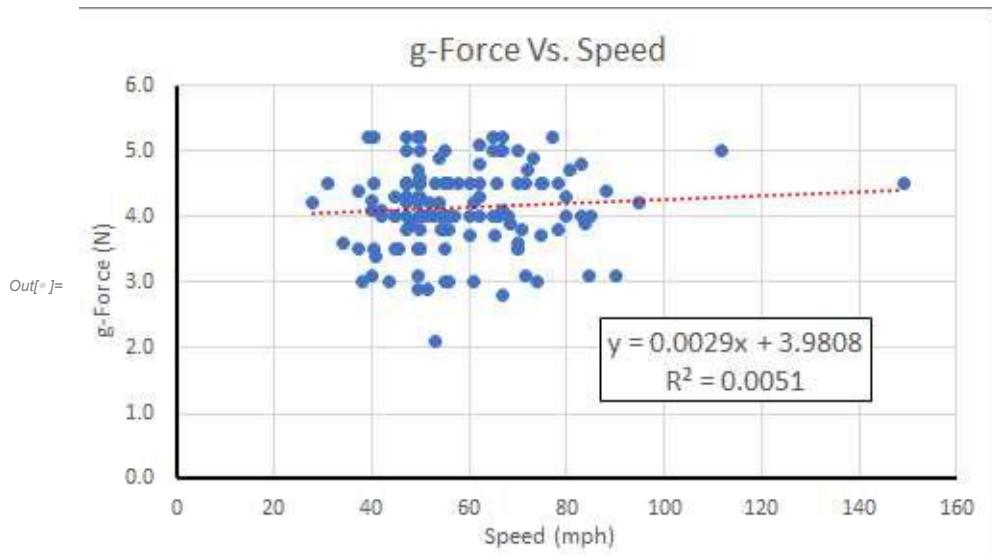


This model was much stronger, so we decided it could be used to model the drop height given speed. However, we noticed that there still was a noticeable difference between our projected and actual drop height values. This % error was likely a result of friction, which our initial kinematic equation assumed did not exist. Thus, to compensate for friction, we reduced our theoretical values by taking the average % error between the two data sets, and multiplied our theoretical values by 1 minus this number.

We used this equation to calculate the unknown drop heights for roller coasters, generating a complete list of roller coaster drop heights. There were some roller coasters where the calculated drop height was greater than the total height, so we set the drop height to the total height for some roller coasters.

Relationships Involving G-Force

As we could not find reliable online databases for g-force, we decided to try extrapolating g-force. Since g-force is tied to the apparent weight a rider experiences relative to the coaster, we tried thinking of locations in a roller coaster where the trajectory of the rider and coaster might contradict and thus create g-force. Thus, our two most likely candidates for sources of g-force were tall drops and sharp turns. However, since we were not provided with any information on turns, and it would be unreasonable to research every single turn in each roller coaster, we decided to focus on the theorized relationship between drop height and g-force. According to the online resources we researched, g-acceleration (which is g-force divided by mass) can be measured using the change in velocity and time that the rider experiences. Based on our previous equation for drop height and speed, we knew that the largest change in velocity usually occurred at the bottom of the largest drop. Assuming that the time taken at the bottom of a curve is constant, we could relate velocity and g-force. We then tried graphing this function linearly (graph below), because one of the two parameters used was assumed constant.



However, graphical analysis indicated that there was no significant correlation between the two. This is most likely because the actual layout of a roller coaster itself lends to different kinds of g-force, that may be enhanced or inhibited based on coaster elements. As we could not reasonably obtain this information for the roller coasters in the time given, we could not create a formula to reliably derive g-force, so we did not use g-Force in our rating calculations.

Relationship Involving Vertical Angle

We tried to find a relationship between a dense variable and vertical angle, but quickly reached a dead end, so we decided not to use vertical angle in our rating calculations.

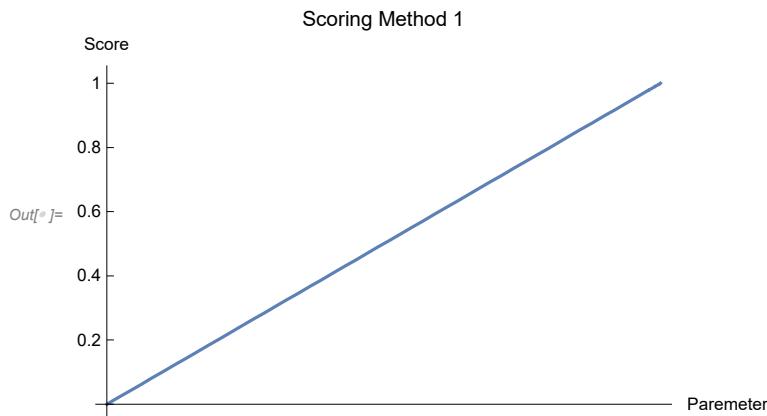
Rating Function

General Method

Our goal was to generate a subscore for each parameter (drop height, speed, duration, inversions, length) that we considered. This subscore was a number between 0 and 1. Then we weighted each category in terms of importance and found the weighted average of the subscores. The quintessential problem for determining subscore was to develop a function with range from 0 to 1 that assigns a subscore for each parameter.

Method 1

Our first function for determining subscore was to take the value for a parameter and divide it by the maximum value for that parameter (maximum value being the highest value of that parameter in the given Excel sheet). The rationale for this was that in general, when the parameters that we are considering get bigger, the rides get better. Therefore, the score of each parameter is determined by how close it is to the maximum value. This ratio will always be between 0 and 1, so it could be used for the subscore.



As the graph above shows, a score between 0 and 1 is assigned to each parameter in a linear fashion,

Pros:

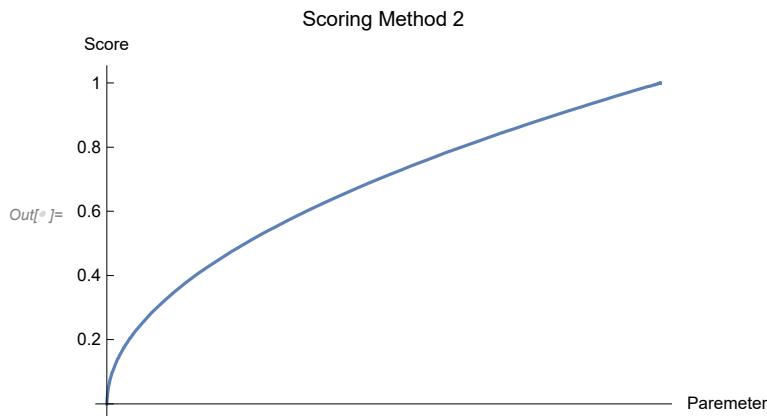
- Restricts subscore value between 0 and 1
- Subscore value based directly on parameter

Cons:

- Assumes the best value for each parameter is the maximum value, which is not the case for every user
- Subscore increases linearly, but in reality there would be diminishing returns on quality with increased values because the difference between 1 and 2 inversions should be more than the difference between 9 and 10 inversions

Method 2

Our second function for determining subscore was to take the subscore value from the first function and square root it. The rationale behind taking the square root of the first function was to have diminishing returns for increased parameter values, so the difference between 1 and 2 inversions is more than the difference between 9 and 10 inversions. This subscore value will always be between 0 and 1 since it is the square root of a ratio between 0 and 1.



As the graph above shows, a score between 0 and 1 is assigned to each parameter in a linear fashion,

Pros

- Restricts subscore value between 0 and 1
- Increasing parameter values yields diminishing returns
- Subscore value is related directly to the parameter

Cons

- Assumes that the best value for each parameter is the maximum value, which is not the case for every user

We could slightly modify this method by determining each subscore via the cube root, or fourth root of the ratio. This would cause a higher increase close to 0, but larger diminishing returns for higher values.

Method 3

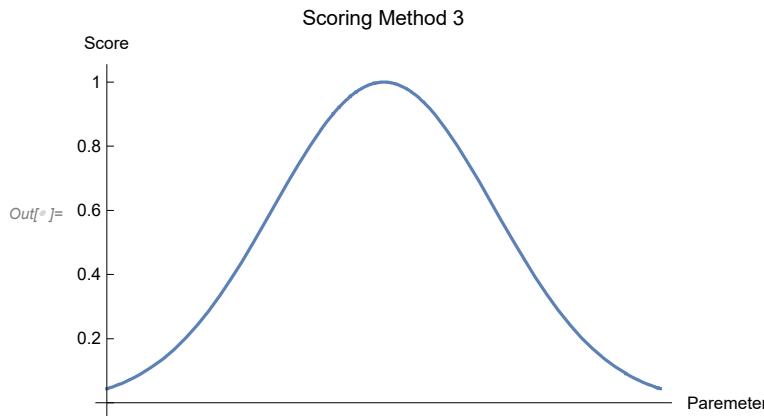
Our third function was determining subscore using a Normal Curve. This method could also take into account a person's ideal score for each category. The method is to center the normal curve around the person's ideal value (for speed, drop, length, duration, or inversions), so their ideal value would receive a rating of 1. The scores would get lower around that value at a rate based on the tolerance of the person. The tolerance of a person is how much they are willing to stray from their ideal parameter value. A higher tolerance would mean that the curve slopes more gradually towards the tails, while a lower tolerance would mean that the curve slopes more sharply towards the tails.

The equation for the subscore of a roller coaster with parameter value x , ideal value μ , and tolerance σ is

$$\text{score} = e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Notice that this is very similar to the formula for the normal curve, except there is no $\frac{1}{\sqrt{2\pi\sigma^2}}$ term in

the front (which corresponds to "amplitude"). We decided to delete this term because we desired an amplitude of 1, as the score of the ideal value should be 1.



As the graph above shows, a score between 0 and 1 is assigned to each parameter in a fashion similar to the normal curve

Pros:

- Restricts subscore between 0 and 1
- Score decreases as value gets farther from ideal on both sides
- Highly flexible formula, as ideal and tolerance can be set

Cons:

- The formula has many constants that must be inputted (like ideal, tolerance)

Combining subscores

To combine the subscore we took the weighted average of the individual subscores. We assigned each subscore a weight from 1-10 ourselves, but the weighted average formula is modular, so we could change the weights depending on user preference for our app's calculations.

Results

Our Rating

For our rating we used the function from method 3 to get a score for each roller coaster, then found the top ten roller coasters with the best scores. We set the ideal values to the maximum values for each parameter and set the tolerance as the standard deviation of the parameter.

We set the speed weight to 8, the inversion weight to 6, the duration weight to 7, the length weight to 2, and the drop height weight to 8.

Justification for Weights

Speed

Speed is one of the quintessential elements that is always thought of when it comes to roller coasters. Not only do other enjoyment factors like g-force and airtime depend on it, but it is also often the first factor a potential rider looks at when researching a roller coaster. With this in mind, we gave speed a weight of 8.

Drop Height

Like speed, drop height is also responsible for generating factors like g-force and airtime, and even generating the speed itself. It also serves as the primary marketing factor of a roller coaster -- even with all other factors being mediocre, people often flock to the most precipitous roller coasters because of the imposing, powerful image that a tall roller coaster strikes. Therefore, we gave drop height an 8.

Duration

This category should be self explanatory: Even a roller coaster with brilliant track design and ludicrous speed is useless without the ability for riders to experience it for long. Few things are as disappointing as getting off a roller coaster that was just getting good, so we rated duration at 7.

Inversions

Inversions can play a great role in inducing the sought-after feeling of weightlessness and disorientation that a roller coaster brings. However, inversions rely on great speed and ride design to work, and are often a relatively minor part of the roller coaster. As a result, we gave inversions a 6.

Length

The actual length of track a roller coaster matters little -- it might synergize with the speed to create duration, but duration is already a category that receives its own ranking. In the interest of including as many enjoyment factors as possible, length is factored into the equation, but it serves no purpose in our model besides aesthetics, which do not count as per our assumptions. Thus, we gave length a weight of 2.

Top Ten List

- 1. Kingda Ka at Six Flags Great Adventure New Jersey
 - Score: 0.369
- 2. Top Thrill Dragster at Cedar Point
 - Score: 0.301
- 3. Formula Rossa at Ferrari World Abu Dhabi
 - Score: 0.294
- 4. Ultimate at Lightwater Valley
 - Score: 0.283
- 5. Red Force at Ferrari Land
 - Score: 0.212
- 6. Smiler at Alton Towers
 - Score: 0.194
- 7. Steel Dragon at Nagashima Spa Land
 - Score: 0.131
- 8. Fury 325 at Carowinds
 - Score: 0.127
- 9. Superman: Escape from Krypton at Six Flags Magic Mountain
 - Score: 0.108
- 10. Tower of Terror II at Dreamworld

- Score: 0.108

Note: there was a tie between Superman and Tower of Terror II. The table of the top ten is below:

Name	Subscore-Speed	Subscore-Inversions	Subscore-Duration	Subscore-Length	Subscore-Drop Height	Final Score
Kingda Ka	0.429387166	5.23505E-07	0	0.002624706	1	0.369043545
Top Thrill Dragster	0.200291233	5.23505E-07	0	0.001206164	0.964841197	0.300757255
Formula Rossa	1	5.23505E-07	0	0.557941016	0.001060357	0.294334453
Ultimate	7.96308E-09	5.23505E-07	1	0.893256903	1.27357E-06	0.283137135
Red Force	0.0/122/23/	5.23505E-07	0	0.001499914	0./50265413	0.2120949/3
Smiler	2.25048E-08	1	4.26236E-07	0.012807592	1.25763E-05	0.194378031
Steel Dragon 2000	0.003857893	5.23505E-07	0.000427129	1	0.255127949	0.131447735
Fury 325	0.003857893	5.23505E-07	4.26236E-07	0.574670315	0.346130883	0.127395386
Superman: Escape from Krypton	0.0102//309	5.23505E-07	0	1.309//E-05	0.409504849	0.108331826
Tower of Terror II	0.010277309	5.23505E-07	0	1.30977E-05	0.409504849	0.108331826

Application Design

Our application was designed to suit the needs of a wide variety of users while still being easy to use for most users. On opening the app the user would see a screen listing our five criteria: drop height, duration of ride, max speed, length, and inversions. The user would then enter a number between one and ten determining how much he or she cares for certain elements (one means you don't care it at all and ten means you would definitely want it on your ride). For example, someone who doesn't care at all about inversions, really enjoys a high drop height, would prefer a fast ride, and does not care how long the ride is would enter the following values: ten for drop height, five for the duration of the ride and for length, eight for max speed, and one for inversions.

Our application would also contain an advanced options tab that would allow users to further personalize their rankings. This would contain two options for each criteria: ideal and tolerance. By default the ideal values would be set to one standard deviation in the positive away from the mean. (SEE FIGURE FOR VALUES) This allows average users who do not want the most extreme roller coaster to find a ride that is just below that extreme. By default the tolerance values will be set to five on a one to ten scale. If you have a lower tolerance (i.e. you want a value close to your ideal value for that criteria) you chose a lower number. If you have a higher tolerance, you chose a higher number.

Our program would then take the tolerance value, divide it by five and multiply it by the data's standard deviation to get our adjusted standard deviation value (for the spread of the score vs parameter graph). We would then calculate each roller coaster's subscores using the formula from method 3. The program would then list out the top ten roller coasters for the user.

Additionally there would be an option to eliminate roller coasters that are outside of a certain driving range from a certain location. This could be done by having the user input their location and maximum driving time and have google maps check each roller coasters' location and eliminate ones that take longer to get to than the maximum driving time, and then display the updated list.

Conclusion

Comparison to Other Ratings

Our first top ten list we found is from a site called the Top Tens. This site simply compiled a list of all the roller coasters in the world and had users vote for the ones they liked the most displaying the ones rated the highest. There are a few issues with this method. First it heavily subjective and relies on impressions of individuals. These impressions could have had a number of factors affecting them including but not limited to user's location, ride preferences, and experience riding roller coasters. Its top rated roller coaster, Millennium Force was rated as the 11th best roller coaster in the world according to our list. Its second rated roller coaster El Toro, was number 97 in our ranking system. Its third best roller coaster, Intimidator 305 was ranked number 13 on our list. The similarities between this list and our list include Fury 325, which was number 4 on their list and 8 on our list, and Kingda Ka, which was number 7 on their list and number 1 on our list. One observation about this list is that a majority of users seem to be localized in the United States, therefore skewing rollercoaster local to the United States much higher due to familiarity bias.

Our second top ten list came from a individual writing for the blog Useless Info Junkie. The list is a bit more objective than the previous one, examining roller coasters from around the world in a fairly unbiased manner (i.e. the author does not seem prefer a certain region over another), and examining ride elements to determine rankings. However, this list is still biased towards the author's opinion of what makes a ride interesting. Its top roller coaster was Takashiba in Japan. This roller coaster was rated number 71 according to our list. This disparity is due largely to Takishiba's interesting way it drops its riders by first holding them at the apex of the drop for a few seconds, followed by a p-shape drop that leads into an inversion. Our model cannot account for this "thrill" factor therefore only placing the roller coaster at position 71 on our list. Its second rated roller coaster, Gravity Max, was not on our original list, so therefore our algorithm did not consider it. Its third best roller coaster was Kingda Ka, which is rated number 1 on our list. One other similarities between our top ten list and their top ten list include The Smiler, which is number 5 on their list and number 6 on our list.

Strengths

- Our model was extremely flexible and adaptable and included many elements
 - Tolerance parameter allows user to determine ranges,
 - Weights and ideal values can also be set
 - If we had g-Force ratings we could easily add it to our rating function (it would just be another subscore)
- Covers a wide range of data
- Our app has a simple interface, but advanced options tabs allows users to feel technical

Weaknesses

- Our model did not consider vertical angle or g-Force because we could not g-Force and vertical angle for many roller coasters
- Fails to account for special types of roller coasters or special features
 - For example, our solution does not consider the possibility of launcher roller coasters
 - Special features like launchers could mess up the drop height extrapolation
- Fails to account for roller coaster structure
 - This could have been improved with a g-Force rating, but we didn't have one

Future Extensions

- Look at angle, g-Force
- Take into account path of coaster
- Take into account different ride elements
- Include a safety rating/a “nausea” rating
- For the app, the user could choose to prefer steel or wood, type, etc.

Appendix

Bibliography

- [1]Coasterpedia The Roller Coaster Wiki. (n.d.). Retrieved November 16, 2018, from <https://coasterpedia.net/>
- [2]Harris, T., & Threewitt, C. (2007, August 09). How Roller Coasters Work. Retrieved November 16, 2018, from <https://science.howstuffworks.com/engineering/structural/roller-coaster9.htm>
- [3]Ishak, R. (2018, March 26). The World's Best Roller Coasters will Leave you Breathless. Retrieved November 16, 2018, from <https://www.farandwide.com/s/best-roller-coasters-80c213fe16fb478a>
- [4]Levine, A. (2017, June 30). The 12 Best Roller Coasters in America. Retrieved November 16, 2018, from <https://www.thrillist.com/travel/nation/best-roller-coasters-in-the-us#>
- [5]Levine, A. (2018, April 8). Roller Coaster Airtime? What's That? Retrieved November 16, 2018, from <https://www.tripsavvy.com/what-is-roller-coaster-airtime-3226486>
- [6]"Rocky Mountain Construction Company". Coaster-Net. February 16, 2011. Archived from the original on December 6, 2011. Retrieved July 9, 2012.
- [7]Roller Coaster. (n.d.). Retrieved November 16, 2018, from <https://www.learner.org/exhibits/parkphysics/coaster.html>
- [8]Roller Coaster Data Base. (n.d.). Retrieved November 16, 2018, from <https://rcdb.com/>
- [9]The 10 Most INSANE Roller Coaster Rides In The World 2018 (With POV Videos). (2018, April 25). Retrieved November 16, 2018, from <https://theuijunkie.com/rollercoaster-rides-2018/>
- [10]ThemeparkreviewTPR. (n.d.). Theme Park Review. Retrieved November 16, 2018, from <https://www.youtube.com/user/themeparkreviewTPR>
- [11]Top Ten Best Roller Coasters. (n.d.). Retrieved November 16, 2018, from <https://www.thetoptens.com/best-roller-coasters/>
- [12]Www.COASTERFORCE.com. (n.d.). CoasterForce. Retrieved November 16, 2018, from <https://www.youtube.com/user/www.COASTERFORCE.com>

Data Table

Imported from Sheets:

