For office use only	Team Control Number	For office use only
T1 _____	8992	F1 _____
T2 _____		F2 _____
T3 _____	Problem Chosen	F3 _____
T4 _____	A	F4 _____

2018
HiMCM
Summary Sheet

Our final goal is to develop algorithms for ranking roller-coasters that can be employed in apps. Thus, we created two ranking algorithms for two different type of users: the casual explorer who just needs an objective ranking, and the professional who wants a personalized ranking optimized to their tastes. To do so, we utilized the dataset given to construct the methods.

To correct for missing data, we designed Linear Iterative Imputation, a technique that uses linear regression to iteratively impute the features based on which useful features (statistically significant at 5%) are selected by the linear model to impute the value of the missing one. In our problem, we first imputed speed, as the variable in the constructed linear model, height, was present in all entries where speed is missing. Then we imputed length with the variables duration and speed, as now all the entries with missing length had duration and speed. We then repeat this to create 6 linear models that imputes the 6 variables in which there are missing values. Majority of our models achieve over R^2 of 0.5, which exceeds greatly the R^2 of 0 from mean imputation.

After imputing the dataset, we grouped the descriptors of the rollercoaster into 3 groups: Thrill, Fundamental, and Inversion, based on their relations in the linear models identified above. The reason for clustering them is to reduce dimensionality in the comparisons conducted below, and improve ease of use for the app that we need to develop.

With these 4 groups, we develop two algorithms: the first one, denoted General Roller-coaster Ranking (GRR), uses a linear model to give a rating for each rollercoaster, with the variables normalized and the weights chosen to balance the importance of each variable group. We further corrected differences in construction methods to ensure fair comparisons.

The second one, called the Personalized Rollercoaster Ranking (PRR), employs the Analytic Hierarchy Process (AHP). As roller-coaster ranking is indeed a very subjective process, we ask the user to enter his/her own preference of the 3 variable groups. Then using AHP we can devise personalized regression weights to use in GRR in order to achieve a final ranking.

Based on our two algorithms, we developed our concept of the application that takes into account the casual and the enthusiast. In the first case, our baseline algorithm confirms the popularity of most highly ranked rollercoasters worldwide. In the second case, with the dimensionality reduction conducted above, we are able to expose a friendly interface to the enthusiast without overwhelming them with the potential levers they could move. The users are also able to limit the regions and the types of rollercoasters included in the ranking.

This results in an application that is able to provide an accurate reflection of universal acclaim of rollercoasters in one click, or alternatively provide one with a personalized ranking with just 3 simple questions.

Personalizing Rollercoasters

Mathematical Models for the Ranking of Rollercoasters Worldwide

2018 HiMCM Problem A

Team #8992

Table of Contents

1. Introduction

Rollercoaster is an extremely popular form of entertainment worldwide. After the end of the first golden age of coasters in the 1930s, it was revived again starting in the 1980s, when there were only 145 coasters operating worldwide. Now, there are over 4000 coasters all around the globe. This of course begs the question, which is the best?

This problem has been studied by enthusiasts and industry alike for many years. Every year, there is a closely watched industry event – the Golden Ticket Awards that give out awards for the best rollercoaster ride around the world. Many individuals and websites have conducted polls or keep rankings of various rollercoasters all over the world. However, all of these awards are based on popularity and human rating, which is extremely subjective. This causes most of these rankings to have high personal bias based on individual likings of the rollercoasters.

Therefore, there is a need for an objective rollercoaster rating system in which the physical characteristics of the rollercoaster determines the rating of it. Moreover, the objective rollercoaster rating system would allow us to predict ratings of roller coasters when they are just coming out, without the need to wait for reviews to ramp up. However, at the end of the day, the fundamental purpose of rating rollercoasters is so that we could select ones we would enjoy, so it is quite clear that there is a strong need to listen to true experiences and preferences of people. What is less clear, however, is the degree of incorporation of "people's experience/preference" into the model. Roughly, we can separate it into two levels:

1. *General Population Level*: We only incorporate information about what the general population thinks is a good rollercoaster. This means that we would want our rating to correspond on the universally acclaimed rollercoasters, as doing elsewise would be contradicting the general view of the rollercoaster.

2. *Specific Individual Level*: We incorporate personal tastes about what the individual thinks is important in a good rollercoaster. This is thus a "personalized" ranking and would have massive appeal due to its understanding of individual preferences. However, the drawback is that most people do not understand the technical details of rollercoasters, and thus asking them to explain the reasoning behind a good rollercoaster experience might be extremely difficult.

Thus, we see that there are two levels of personal preference we can include, with arguments for/against both sides. It is clear that the "average" user would in fact benefit more from the first approach than the second one as he/she would be able to check the ranking without entering any

information, and vice versa for the enthusiast. Thus, to provide a good experience to both types of people, we thus construct two models that share the underlying mathematical basis, but incorporate the personal preferences in different ways to give rise to different algorithms.

To construct these models, we would need data on the objective characteristics of the rollercoasters, which is provided in the dataset given. This dataset however, contains much missing data, and therefore before we can talk about the algorithm for ranking, we first need to develop a model for imputing the data, which we would explore in the next section.

2.Data and Imputation

2.1 Introduction to Data and Preprocessing

We are given a dataset of 300 roller-coasters with 19 characteristics. Excluding the non-informative statistics (Status, as every coaster included here is operating), and factors dealing with name and places, we are left with 12 statistics: Construction, Type, Year Opened, Height, Speed, Length, Inversions (Yes/No), Number of Inversions, Drop, Duration, G Force, and Vertical Angle.

We identify that many statistics are missing, with in fact only the Construction, Type, Year, and Inversions statistics being fully present. We identify that Height is also only missing in one roller coaster – the Harbin Happy Angel Coaster. A deeper dive onto this specific coaster reveals that it is a very recent coaster that has only been in operation for less than a few months. As most of its statistics is missing, we decided to remove this coaster from this database – we do not believe that it could reliably imputed from other coasters as it utilizes “a new construction method” and is “a new concept”, according to Xinhua News.

Thus, in the 299 coasters we are left with, the following is a table of the amount of missing entries in each column:

Column	Number of Entries with Missing Data
Speed	4
Length	4
Drop	157
Duration	75
G Force	216
Vertical Angle	208

Table 1: Number of Missing Entries for Each Column

We see in Table 1 that both Speed and Length are almost fully known while Drop, G Force, and Vertical Angle are highly missing. To impute the values of these data, we utilize linear regression to iteratively impute these values, based on what variables is needed. We call this Linear Iterative Imputation.

2.2 Variables

Variable Symbol	Name/Meaning
y_m	The missing variable that we want to predict
x_i	A potential covariate for imputing the missing variable
β_i	The coefficient for the potential covariate for imputing the missing variable

2.3 Assumptions

- We assume that the underlying true relations of these variables are linear

Justification: It is hard to justify that the true underlying relationships between these variables are actually linear, but to be able to impute the data, we nevertheless need some assumption, and in this case a linear model is suffice as we show through our high accuracy.

- We assume this dataset is a representative dataset of the entire set of rollercoasters

Justification: If this dataset was not a representative dataset from the entire sample of rollercoasters, then the data that we are imputing might be biased or generate wrong relations due to how this sample was generated. In this case, we assume that the 200 rollercoasters we are given is not strongly skewed in any particular covariate.

2.4 Linear Iterative Imputation

To conduct Linear Iterative Imputation, we first need to understand what variables each missing variable need to predict it well. We do this by constructing linear models for each variable that has missing data against at most 9 statistics – all 12 identified statistics excluding Drop, G Force, and Vertical Angle are considered. These three are excluded due to the massive amount of

missing data. That is for every missing variable y_m, we select variables $x_1, x_2, \dots, x_k$ so that we have the model:

$$y_m = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$

To ensure that we are only selecting useful variables, we only keep variables x_i that are significant on the 5% level.

In the interest of clarity, we would only illustrate the process of building the model for one such variable as the process is similar for all variables.

2.4.1 Illustration of Linear Model Construction

We first run a standard linear regression for the variable Drop against all other variables, to reveal the following estimates of the coefficients and p-values:

Variable	Estimated Coefficient	*p*-value
Construction (Wood)	-404.7	0.01
Type (Inverted)	-11.34	0.88
Type (Sit Down)	1.620	0.89
Type (Stand Up)	1.785	0.59
Type (Wing)	-8.585	0.99
Year Opened	-14.74	0.17
Height	0.1579	$1.12 * 10^{-6}$
Speed	3.116	$<1 * 10^{-16}$
Length	-0.001287	0.50
Inversions (Yes/No)	-9.996	0.12
Number of Inversions	0.09277	0.93
Duration	0.04418	0.38

Table 2: Coefficients for First Linear Model for Drop

We see that only Construction, Height, and Speed are significant. We then rerun the model with only these three variables:

Variable	Estimated Coefficient	*p*-value
Construction (Wood)	-5.77	0.08
Height	0.33804	$1.69 * 10^{-11}$
Speed	2.956	$<1 * 10^{-16}$

Table 3: Coefficients for Second Linear Model for Drop

Now we see that the Construction variable is not significant on the 5% level. Therefore, by our criteria, we drop it further:

Variable	Estimated Coefficient	p-value
Height	0.3513	$2.67 * 10^{-12}$
Speed	2.94	$<1 * 10^{-16}$

Table 4: Coefficients for Third Linear Model for Drop

In Table 4 we see that both remaining variables are significant, and therefore, we define Height and Speed to be the useful variables in imputing Drop.

The following table illustrates the variables identified in the final linear models:

Missing Variable	Useful Variables	R^2
Speed	Height	0.696
Length	Duration, Speed	0.526
Drop	Speed, Height	0.954
Duration	Length, Construction	0.514
G Force	Length, Construction	0.477
Vertical Angle	G Force	0.233

Table 5: Linear Models for Imputation

We see that in most models, we manage to select only a few variables while retaining relatively high accuracy.

Here there is one caveat: The final variable, Vertical Angle, cannot be accurately imputed by any of the 9 variables identified above. However, when we use G Force to regress against it, it is significant at the 5% level. Thus, even though the G Force and the Vertical Angle are both missing many entries, we decided to use the fitted relation between G Force and Vertical Angle

Once we now have this set of relations, we can construct an ordering of imputation so that we can use these 6 models effectively. For example, we cannot impute Drop before Speed, as Speed is required to impute Drop. However, here we have a potential problem as to impute Length we need Duration but to impute Duration we need Length!

Fortunately, for all the entries in which length is missing (which there are only 4), duration exists, so we can first use the existing duration to impute the length, and then use the fully complete length to impute the rest of the durations.

Thus, a valid ordering is:

$$Speed \rightarrow Length \rightarrow Drop \rightarrow Duration \rightarrow G\ Force \rightarrow Vertical\ Angle$$

And we thus impute iteratively accordingly. This gives us a full dataset.

2.4.2 Final Results

To illustrate the effectiveness of our imputation, we would show the correlation between the numeric variables in the 9 statistics before and after imputation. Ideally we would like the correlation to not change before and after imputation so that the data can be considered similar. The following graph is a plot of the correlation before the imputation (on the subset in which these statistics exist):

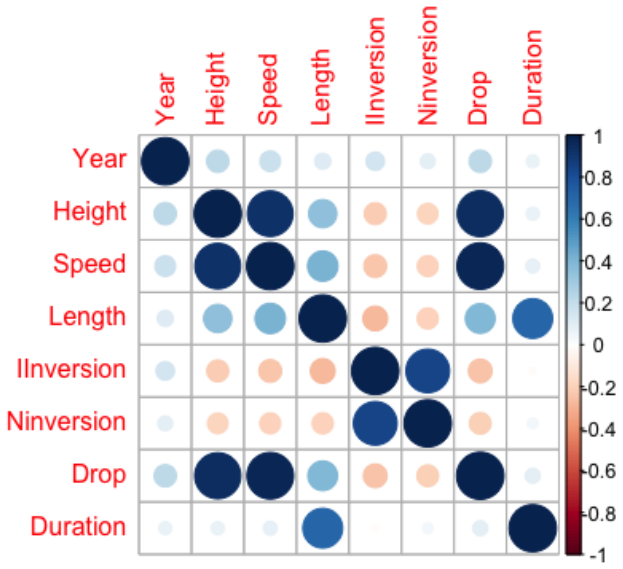


Figure 1: Correlation of Variables before Imputation

The next graph shows the correlation after Imputation;

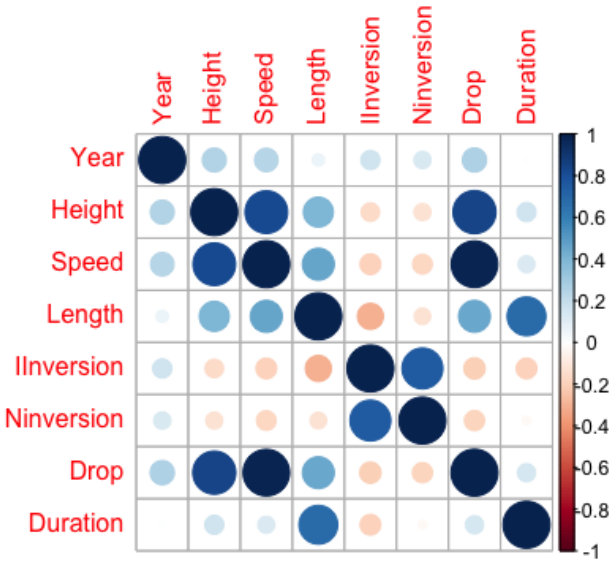


Figure 2: Correlation of Variables after Imputation

One could see that these two graphs are very similar, which suggests that our imputation algorithm has done a good job of keeping the relations between the original variables intact.

3.General Rollercoaster Ranking (GRR) Model

3.1 The Approach

Through the imputation process, we can see that the variables which form natural groups with each other due to the high relevance/correlation with each other. We can roughly see three groups:

- **Thrill (Speed, Drop, Height):** As it is clear in Table 5, we can use Height to Impute Speed while using Height and Speed to impute Drop with high confidence. This implies that these variables share deep relationship with each other. It is quite clear that all of these variables define the excitement or more specifically the thrill of the ride: Drop, Height and Speed are what would commonly be used to define how extreme a ride is. Note that we did not artificially create this grouping – it fell out naturally from the imputation process.

- **Fundamentals (Vertical Angle, G Force, Length, Duration):** These 5 variables form a second group as can be seen through the high correlation between each variable in Figure 2, and the linear model relations in Table 5. One could describe this group of variables as the fundamentals – these are characteristics that define the construction of the rollercoaster and is more hampered by the physical and engineering constraints than the first group.
- **Inversion (Number of Inversions, Inversions(Yes/No)):** These 2 variables were not present in any linear model and they share high correlation with each other, as it is clear that one is merely a masked version of the other. Inversions on a rollercoaster is another common attraction point, but from Table 5 it is clear that it is in its own group and not highly related to the other descriptive variables. For the model, since Inversions (Yes/No) is just a masked version of the Number of Inversions statistic, we would only use Number of Inversions.

Note here three variables have not been classified out of the 12 statistics: The Year of Opening, Construction and the Type of Ride. We would treat them individually, as reasoned below:

- The year of opening had low correlation with any other variable and was not important in the linear model. Moreover, "when the ride was opened" should not be a factor in determining if a ride's experience is good or not if the ride has been kept in good condition. Thus, we would remove this variable.
- On the other hand the type of ride, on the other hand, might potentially impact people's tastes based on individual preferences. However, as explained in the introduction, this model is only designed to capture general population trends and thus we would ignore it.
- The construction variable is again a very subjective variable, with many articles arguing for and against the merit of a wood construction against a steel construction. It is generally accepted that the steel coasters have better statistics than the wood coasters due to physical capabilities of steel, however wood coasters often have a much more "thrilling" ride due to the wobbly nature of the track. We would consider this variable in the model, but in a separate way from the grouped variables above.

Therefore, with these three groups, we can devise a linear model for a ranking, where the rating of a rollercoaster is r_G:

$$r_G = \beta_t x_s + \beta_t x_{dr} + \beta_t x_h + \beta_f x_{va} + \beta_f x_g + \beta_f x_l + \beta_f x_d + \beta_I x_n$$

Where x_s is the variable representing speed, x_{dr} drop, x_h height, x_{va} vertical angle, x_g G Force, x_l length, x_c construction, x_d duration, and x_n number of inversions. The three coefficients β_t, β_f and β_I are group coefficients that represent how important each group is, as identified above.

To make these three coefficients meaningful, we first need to scale the x variables appropriately. For example, it is not appropriate for length to have the same coefficient as G force as the former takes values of thousands while the latter never exceeds 10. Thus, for each variable, we would first apply a scaling such that the mean of the variable in the dataset is 0, and the variance of it is 1. The following is a table of the various means and variances of the variables:

Variable	Mean μ	Variance σ^2
x_s	59.5	264.4
x_{dr}	129.3	5475.8
x_h	135.5	4408.6
x_{va}	77.1	55.71
x_g	4.18	0.21
x_l	3150.8	2102981
x_d	121.2	2139.5
x_n	2.21	6.75

Table 6: Means and Variances for various Variables

Thus, we would scale our variables so that we have:

$$x_i' = \frac{x_i - \mu_i}{\sigma_i}$$

Where σ_i denotes the standard deviation (square root of variance) for variable i, and μ_i the mean for variable i.

Therefore, in fact our rating equation is actually:

$$r_G = \beta_t x_s' + \beta_t x_{dr}' + \beta_t x_h' + \beta_f x_{va}' + \beta_f x_g' + \beta_f x_l' + \beta_f x_d' + \beta_I x_n'$$

Then, for this general model, we would assume that for the general population, the fundamentals are just as important as the thrill, which is more important than the number of inversions (as inversions are usually only pursued by enthusiasts). In accordance with the Analytic Hierarchy Process, we assign an overall weight of 5 to groups Thrill and Fundamentals, and a weight of 1 to group Inversion. Then, we further correct the weights by the number of variables in each group to reach the individual coefficients:

$$\beta_t = \frac{5}{3} \qquad \beta_f = \frac{5}{4} \qquad \beta_I = 1$$

As there are 3 variables in the thrill group, 4 in the fundamentals group and only 1 in the inversion group.

Therefore, our rating model now becomes:

$$r_G = \frac{5}{3}(x'_s + x'_{dr} + x'_h) + \frac{5}{4}(x'_{va} + x'_g + x'_l + x'_d) + x'_n$$

Here, we have ignored the effect of different construction. As noted in Table 5, the construction has a major effect on the duration and G Force of the ride. Thus, to compare Steel and Wood rides fairly, we have to correct for it. The linear models suggests that using wood construction on average makes the duration of the ride 26.2 seconds shorter and decreases the G Force by 0.7620. Thus, we correct by that amount in the Steel trains. Therefore, our final model is:

$$\begin{cases} r_G = \frac{5}{3}(x'_s + x'_{dr} + x'_h) + \frac{5}{4}\left(x'_{va} + x'_g + x'_l + x'_d\right) + x'_n & \text{for Wood Coasters} \\ r_G = \frac{5}{3}(x'_s + x'_{dr} + x'_h) + \frac{5}{4}\left(x'_{va} + x'_g + x'_l + x'_d + \frac{-0.7620}{\sqrt{0.21}} + \frac{-26.2}{\sqrt{2139.5}}\right) + x'_n & \text{for Steel Coasters} \end{cases}$$

3.2 Variables

Variable Symbol	Name/Meaning
x_s	Speed of the Rollercoaster
x_{dr}	Drop length of the Rollercoaster (Feet)
x_h	Height of the Rollercoaster (Feet)
x_{va}	Vertical Angle of the Rollercoaster
x_g	G Force of the Rollercoaster
x_l	Length of the Rollercoaster
x_d	Duration of the Rollercoaster
x_n	Number of Inversions on the Rollercoaster
x_s	Speed of the Rollercoaster
x'_i	Scaled versions of the variables above
μ'_i	Mean of the variables above
σ'_i	Standard Deviation of the variables above
$\boldsymbol{\beta_t}$	Linear coefficient for the Thrill Group
$\boldsymbol{\beta_f}$	Linear coefficient for the Fundamentals Group
$\boldsymbol{\beta_I}$	Linear coefficient for the Inversions Group
$\boldsymbol{r_G}$	Rating for the Rollercoaster

3.3 Assumptions

- We assume that the rating of the rollercoaster is a linear model of its respective variables.

Justification: For most of the variables, more means better, and in a linear way. For example, a 5km rollercoaster is the "same amount of better" than a 4km rollercoaster as a 6km one is to a 5km one. However, arguably, there are variables such as g force and vertical angle that are physically constrained and should not be linear. However, in the interest of simplicity, we would assume they are linear in this model. It is further discussed in the Strengths and Weaknesses section.

- We assume the coefficients within each respective group (Thrill, Fundamentals, Inversion) are the same

Justification: It is conceivable that the coefficients can be different within these groups, especially among different people. However, our imputation showed strong correlation between these variables, suggesting that whenever one variable changes, the other does also. For example, when you go for a higher rollercoaster, the G Force you are experiencing most likely increases – which is quite intuitive. Thus, grouping them together is a justified decision, and has many benefits down the road in reducing complexity and enabling more complex processes to be added later, as shown in the PRR model.

3.4 Results and Comparison

Using the rating model identified above, our ranking results in the following. Note we extend our list from Top 10 as required by the question to Top 20 to achieve a more holistic comparison:

Ride	Rating
Kingda Ka (USA)	0.994
Formula Rossa (UAE)	0.872
Top Thrill Dragster (USA)	0.598
Red Force (Spain)	0.360
Steel Dragon 2000 (Japan)	-0.041
Wildfire (Sweden)	-0.070
Superman: Escape from Krypton (USA)	-0.073

Leviathan (Canada)	-0.176
Fury 325 (USA)	-0.201
Lightning Rod (USA)	-0.249
Tower of Terror II (Australia)	-0.262
Goliath (USA)	-0.272
Wodan Timbur Coaster (Germany)	-0.343
Intimidator 305 (USA)	-0.376
El Toro (USA)	-0.390
T Express (South Korea)	-0.394
Beast (USA)	-0.395
Voyage (USA)	-0.448
Do-Dodonpa (Japan)	-0.573
Millennium Force (USA)	-0.599

Table 7: Top 20 Ranking of GRR

We compare our results to two other ranking systems. The first ranking system is the Coasterbuzz rating system, which is an average of all submitted user ratings, and is thus a purely subjective rating mechanism. The top 20 in that are:

Top 20 of Coasterbuzz
Steel Vengeance
Fury 325
El Toro
Lightning Rod
Twisted Timbers
Millenium Force
Twisted Colossus
Boulder Dash
Wicked Cyclone
Voyage
Maverick
Iron Rattler
Superman The Ride
Goliath
Leviathan
Ravine Flyer II
Outlaw Run
Mako

Nemesis
Phoenix

Table 7: Top 20 Ranking of Coasterbuzz

Coasterbuzz mainly has North American reader base, so it is not surprising that none of the foreign coasters in the top 20 we chosen are in the top list in Coasterbuzz. Out of the 12 USA/Canadian coasters we chose, 7 also appeared in the top 20 list of Coasterbuzz, suggesting that we are able to replicate the preference of people towards rollercoasters with a purely objective basis.

The second baseline we will compare to is Lifed.com ranking, which is an expert ranking. The top 20 list is:

Top 20 of Lifed
Bizarro
Millennium Force
El Toro
Expedition GeForce
Voyage
Kingda Ka
Intimidator 305
Goliath
Behemoth
Nemesis
Balder
Top Thrill Dragster
X2
T Express
Katun
Boulder Dash
Ravin Flyer
Formula Rossa
Maverick
Nitro

Table 8: Top 20 Ranking of Lifed

Here we share 8/20 names, which again is a respectable percentage. We note that compared to this list, (in which the expert team is again US based) which has 4/20 rides outside of

USA/Canada, we contain 8/20 rides outside of Canada, suggesting a much more global reach while able to replicate many of the top opinions across the community.

3.5 Sensitivity Analysis

There are three parameters in our model: β_t, β_f and β_l. Therefore, we would conduct sensitivity analysis against each of the parameters

3.5.1 Sensitivity Against β_t

We increase β_t by 0.1 to test how does the ranking change with changing β_t. The New Top 20 from our algorithm is:

Top 20 after β_t Change
Kingda Ka (USA)
Formula Rossa (UAE)
Top Thrill Dragster (USA)
Red Force (Spain)
Superman: Escape from Krypton (USA)
Steel Dragon 2000 (Japan)
Leviathan (Canada)
Fury 325 (USA)
Tower of Terror II (Australia)
Wildfire (Sweden)
Intimidator 305 (USA)
Lightning Rod (USA)
Goliath (USA)
El Toro (USA)
Wodan Timbur Coaster (Germany)
Do-Dodonpa (Japan)
T Express (South Korea)
Millennium Force (USA)
Beast (USA)
Voyage (USA)

Table 9: Top 20 Ranking after β_t Change

The italicized ranking shows the ranking where something has changed, and the new Top 20 rides under this scenario are highlighted. We can see that even after a 10% change, there are no new rides coming into the Top 20, with others basically a slight reshuffling of the ranking. More importantly, the Top 4 did not even move, suggesting that their rankings are extremely robust against change.

3.5.2 Sensitivity Against β_f

We increase β_f by 0.1 to test how does the ranking change with changing β_f. The New Top 20 from our algorithm is:

Top 20 after β_t Change
Kingda Ka (USA)
Formula Rossa (UAE)
Top Thrill Dragster (USA)
Red Force (Spain)
Steel Dragon 2000 (Japan)
Superman: Escape from Krypton (USA)
Leviathan (Canada)
Wildfire (Sweden)
Fury 325 (USA)
Tower of Terror II (Australia)
Lightning Rod (USA)
Intimidator 305 (USA)
Goliath (USA)
Wodan Timbur Coaster (Germany)
Beast (USA)
T Express (South Korea)
El Toro (USA)
Voyage (USA)
Coaster Through the Clouds (China) -- New
Do-Dodonpa (Japan)

Table 10: Top 20 Ranking after β_f Change

The italicized ranking shows the ranking where something has changed, and the new Top 20 rides under this scenario are highlighted. We can see that even after a 10% change, there is only 1 new ride coming into the Top 20, with others basically a slight reshuffling of the ranking. Again, the Top 4 did not even move, suggesting that their rankings are extremely robust against change.

3.5.3 Sensitivity Against β_I

We increase β_I by 0.1 to test how does the ranking change with changing β_I. The New Top 20 from our algorithm is:

Top 20 after β_I Change
Kingda Ka (USA)
Formula Rossa (UAE)
Top Thrill Dragster (USA)
Red Force (Spain)
Wildfire (Sweden)
Steel Dragon 2000 (Japan)
Superman: Escape from Krypton (USA)
Leviathan (Canada)
Goliath (USA)
Fury 325 (USA)
Lightning Rod (USA)
Tower of Terror II (Australia)
Wodan Timbur Coaster (Germany)
Intimidator 305 (USA)
El Toro (USA)
T Express (South Korea)
Beast (USA)
Voyage (USA)
Outlaw Run (USA) – New
Do-Dodonpa (Japan)

Table 11: Top 20 Ranking after β_I Change

The italicized ranking shows the ranking where something has changed, and the new Top 20 rides under this scenario are highlighted. We can see that even after a 10% change, there is only 1 new ride coming into the Top 20, with others basically a slight reshuffling of the ranking. Again, the Top 4 did not even move, suggesting that their rankings are extremely robust against change.

3.6 Strengths and Weaknesses

3.6.1 Strengths

- The GRR model only uses objective factors and does not incorporate personal preferences.
- The GRR model does not attempt to group variables together in an ad-hoc way. Instead, it employs the information from the Linear Iterative Imputation to create natural groupings of the variables.
- The GRR model corrects for the differences between steel and wood coasters, and thus allow comparisons across these categories to be fair.
- The model is linear, so that a scaling of the weights would not impact the final result. This is important in hierarchical decision models, as the scale of the weights itself does not mean anything, and thus should not affect the final ranking.

3.6.2 Weaknesses

- The GRR model does not capture any personal preferences, which means that it can only serve as a general ranking without the capability of correcting against different tastes.
- The model does not try to consider interaction effects between different variables and only considers the original variables on a linear model; non-linear models might improve explaining of some factors. For example, G- Force can never exceed 2G due to human constraints – thus instead of using a linear model where the G Force seems like it could be extrapolated infinitely far, one might be better off using a model that tapers off at large values (such as logistic model).
- The model focuses only on the specific factors that were given in the dataset and does not consider additional factors that are objective but could affect the rating/experience of a person – this includes information about the general park, the area, and the country. As such, Kingda Ka, which holds many top records, stays comfortably at the top even in our sensitivity tests. While it does appear in most top rollercoaster rankings, its position usually isn't as stable as it may seem from our ranking, as the physical attributes of the ride might not be everything in determining a ranking of the best roller coaster.

4.Personalized Rollercoaster Ranking (PRR) Model

4.1 The Approach

With the GRR model detailed in the last section, we created a model that was able to grasp the ratings of top rollercoasters reasonably well. However, it was not flexible and could not adjust for personal preferences. Thus, in this section, we introduce a new layer based on the Analytic Hierarchy Process (AHP) that allows us to personalize the ranking.

Let us first reproduce the general formula for the GRR from the last model:

$$r_G = \beta_t x'_s + \beta_t x'_{dr} + \beta_t x'_h + \beta_f x'_{va} + \beta_f x'_g + \beta_f x'_l + \beta_f x'_d + \beta_I x'_n$$

In the GRR, we took a specific combination of $(\beta_t, \beta_f, \beta_I)$ based on the general population's preference. However, it is clear that everyone could be different. Thus, to solve this problem, we would employ AHP to decide the coefficients for this model.

We want to specifically highlight here that it is the grouping of the variables, powered by our imputation method that allowed us to utilize the AHP efficiently. If we directly utilized AHP on the 8 variables above, we would have to compare the relative importance of all pairs of the 8 variables – that would be 28 pairs. Conscious that our algorithm is user-facing, it is infeasible to ask the user 28 questions just to get a personalized rollercoaster ranking. With 3 groups however, we only need to ask 3 comparative questions, which is much more feasible.

The process of the PRR model is as followed:

1. For each possible Pair (Thrill vs Fundamentals, Fundamentals vs Inversion, etc), we would ask the person to rate the relative importance of one to another based on the Fundamental Scale for Pairwise Comparisons, reproduced below.

The Fundamental Scale for Pairwise Comparisons		
Intensity of Importance	**Definition**	**Explanation**
1	Equal importance	Two elements contribute equally to the objective
3	Moderate importance	Experience and judgment moderately favor one element over another
5	Strong importance	Experience and judgment strongly favor one element over another
7	Very strong importance	One element is favored very strongly over another; its dominance is demonstrated in practice
9	Extreme importance	The evidence favoring one element over another is of the highest possible order of affirmation
Intensities of 2, 4, 6, and 8 can be used to express intermediate values. Intensities of 1.1, 1.2, 1.3, etc. can be used for elements that are very close in importance.		

Figure 3: Fundamental Scale Diagram

2. Then, in the Criteria matrix below, we fill in the respective numbers. Here (t, f) represents the relative importance of thrill to fundamentals, rated on the fundamental scale above. Similarly (t, I) represents the relative importance of thrill to Inversions.

$$C = \begin{matrix} 1 & (t,f) & (t,I) \\ 1/(t,f) & 1 & (f,I) \\ 1/(t,I) & 1/(f,I) & 1 \end{matrix}$$

3. After filling in the matrix, we calculate the principal normalized eigenvector of C, v_C. Denote the 3 values coordinates of the principal eigenvector as (v_{C1}, v_{C2}, v_{C3}) . Then the weights for the linear model would be:

$$\beta_t = \frac{v_{C1}}{3} \qquad \beta_f = \frac{v_{C2}}{4} \qquad \beta_I = v_{C3}$$

4. Run the GRR Model with the specified combination and show the ranking.

Note there is no Sensitivity Test nor Results section for this model as it innately depends on the input of the user to perform the ranking. For the general overview of how stable is the algorithm, one could see the sensitivity tests conducted in the GRR Model.

4.2 Variables

We only list the additional new variables that appeared here:

Variable Symbol	Name/Meaning
(t, f)	The relative importance of Thrill with respect to Fundamentals
(t, I)	The relative importance of Thrill with respect to Inversions
(f, I)	The relative importance of Fundamentals with respect to Inversions
v_C	The principal eigenvector of the Criteria Matrix
v_{C1}, v_{C2}, v_{C3}	The three dimensions of the principal eigenvector
C	The Criteria Matrix (AHP)

4.3 Illustration of PRR

In this section, we would illustrate a particular case for using PRR. We would assume the answers to the questions are as below:

- "What is the relative importance of Thrill to Fundamentals?" – "Thrill is extremely important compared to Fundamentals (9)"
- "What is the relative importance of Fundamentals to Inversion?" – "Fundamentals is equally important compared to Inversion (1)"
- "What is the relative importance of Thrill to Inversion" – "Thrill is extremely important compared to Inversion (9)"

Here we are describing the profile of a person who is extremely thrill-seeking but is not really into inversions, and would prefer a balanced ride fundamentals with number of inversions. The matrix for him/her is thus:

$$C = \begin{matrix} 1 & 9 & 9 \\ 1/9 & 1 & 1 \\ 1/9 & 1 & 1 \end{matrix}$$

The principle eigenvector is thus:

$$v_C = \begin{pmatrix} \frac{9}{\sqrt{83}} \\ \frac{1}{\sqrt{83}} \\ \frac{1}{\sqrt{83}} \end{pmatrix}$$

Thus, our beta values are:

$$\beta_t = \frac{3}{\sqrt{83}} \qquad \beta_f = \frac{1}{4\sqrt{83}} \qquad \beta_I = \frac{1}{\sqrt{83}}$$

Then we run the GRR model, to produce the following ranking:

Top 20 for PRR Run
Kingda Ka (USA)
Top Thrill Dragster (USA)
Formula Rossa (UAE)
Red Force (Spain)
Superman: Escape from Krypton (USA)
Tower of Terror II (Australia)
Fury 325 (USA)
Steel Dragon 2000 (Japan)
Do-Dodonpa (Japan)
Leviathan (Canada)
Millenium Force (USA)
Intimidator 305 (USA)
Soaring Dragon & Dancing Phoenix (China) -- New
Hyperion (Poland) -- New
Eejanaika (Japan) -- New
Coaster Through the Clouds (China) -- New
Titan (USA) -- New
Shambhala (Spain) -- New
Outlaw Run (USA) – New
Goliath (USA)

Table 12: Top 20 Ranking for PRR

Note that this ranking is vastly different from the original GRR ranking - There are 7 coasters appearing on the Top 20 list that did not appear below. Even the Top 4, which were extremely stable in the sensitivity tests, swapped orders as Top Thrill Dragster over took Formula Rossa. This is mainly due to the fact that Top Thrill Dragster has a height of 420m compared

to Rossa's 171m, with similar speed and a higher drop. Thus, though Rossa's ride is much longer, the thrill of the Top Thrill Dragster moved its place from third to second.

4.4 Strengths and Weaknesses

4.4.1 Strengths

- The PRR model is able to capture significant personal preferences within 3 simple questions that the users need to answer.
- It retains all of the desirable properties of the GRR Model (robustness, ease to use, etc)

4.4.2 Weaknesses

- The PRR model can only capture personal preferences on the two dimensions that we have defined – more granular personal preferences are lost in the process.
- The PRR though solving one of the weaknesses of GRR by adding personal preferences, still retains most of the weaknesses of GRR.

5.News Release

New Way to find your favorite Rollercoaster!

Tired of deciding which rollercoaster is the best? Need some suggestions on what rollercoaster to next go to? With the efforts of the entire Team #8992, we have devised a new way to provide you all the answers!

Unlike previous rankings which heavily relied on experts and/or personal ratings which are subject to much bias, we only use objective mathematical quantities, such as how long the ride is, how much force you would experience on the ride, what's the maximum speed on the ride, etc. We then develop a model to use all of these features and provide a rating that is objective and reflects the true performance of these rollercoasters.

If you want more, there is even a personalized ranking algorithm in which in just three questions, we would create a ranking that is suited just for you. This algorithm is asks you about the relative importance of different qualities of the ride to determine which ride would be best for you. Then it produces a personalized list that is unique for you.

All of this is available in an easy to use app, along with other features such as region/type selection.

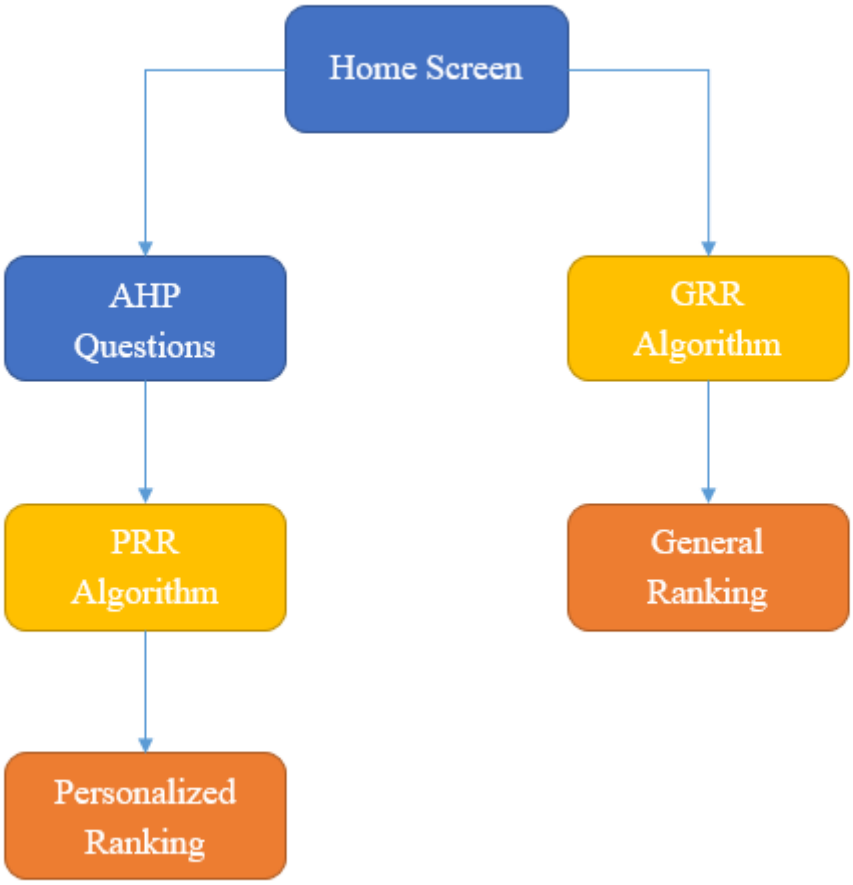
Best

-Team #8922

6.App Design and Development

As requested in the question, we would design an app for people to check the current top rankings of the rollercoasters. As in the algorithm development, we have already thought about the two different type of users that would use this app, so we would prepare two workflows in the app to accommodate the two different types of users.

6.1 Flowchart of App



As one could see above, there are two flows within the app. When you open the app, you would see two buttons – one for the general ranking, and one for the personalized ranking. If you choose general ranking, you would immediately see the general ranking of rollercoasters

worldwide as generated by the GRR Algorithm. You can then further limit your choices to specific regions or type of rollercoasters you would like to enjoy.

If you choose the advanced path, then you would be prompted 3 AHP type questions as detailed in the PRR algorithm. Then the PRR Algorithm would calculate a ranking that adheres to your tastes.

6.2 App Design

To illustrate the app, we would show a few mockup of the screens. The home page would just be populated by two buttons as shown below:

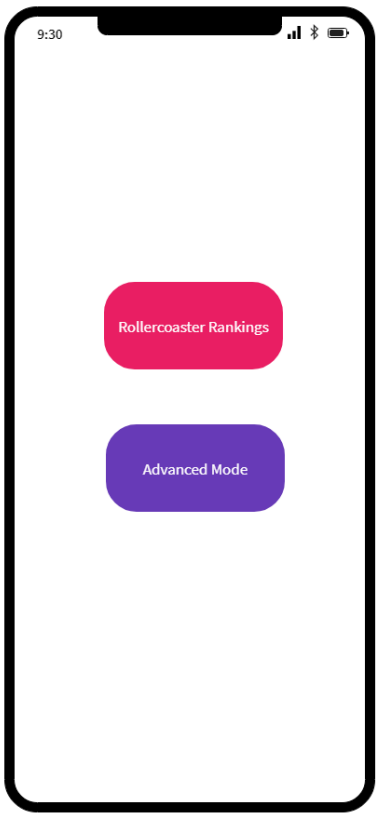


Figure 4: Home Screen

These two buttons would take you down either the General Ranking Path or the Personalized Ranking path. If you choose the Advanced Mode (Personalized), then you get the following screen:

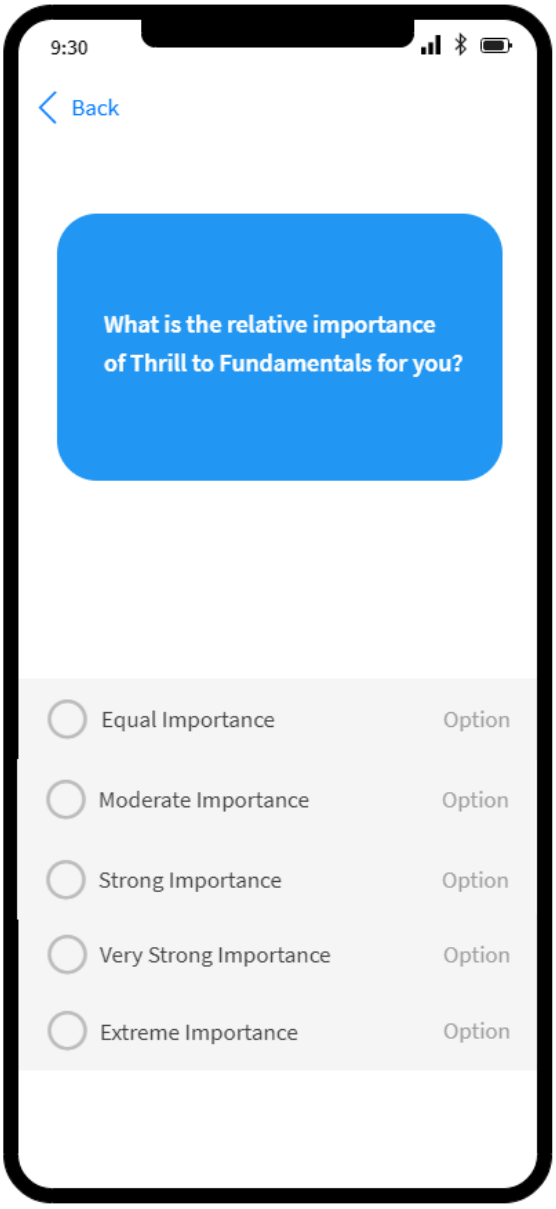


Figure 5: Questions

Then after the user finish answering the questions, the ranking would appear:

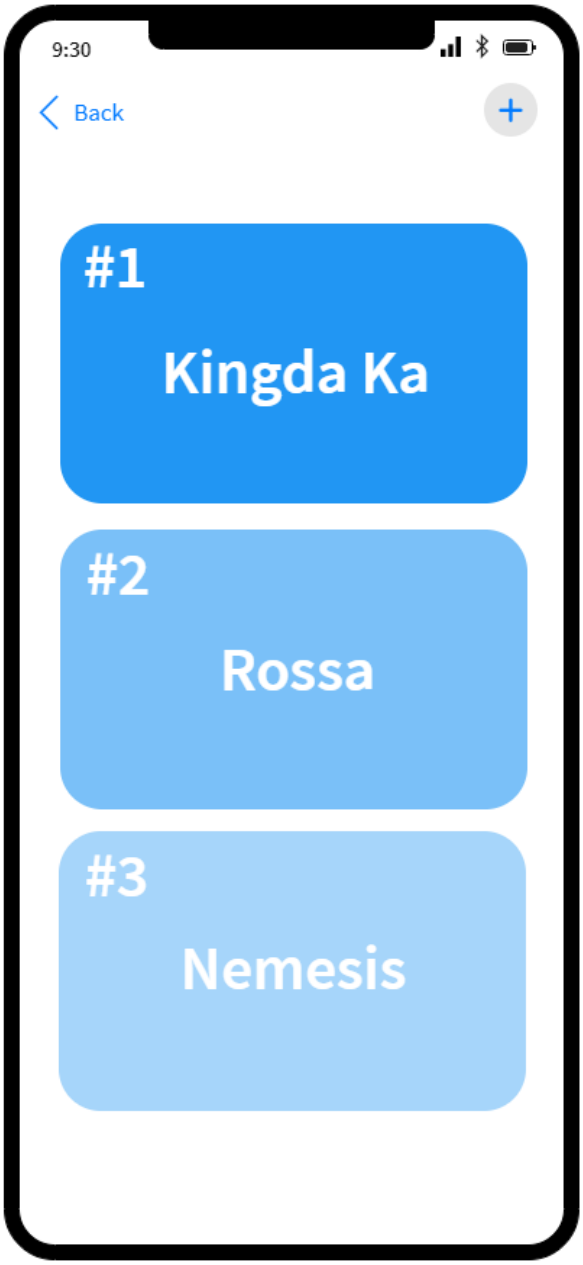


Figure 6: Ranking

The plus icon on the top side indicates that the user could add additional options to filter his ranking. This includes filtering by region, ride, and others.

7.Appendix

7.1 References

https://www.rcdb.com.

https://www.ultimaterollercoaster.com/

https://coasterpedia.net/

https://abcnews.go.com/WNT/story?id=130111&page=1

http://goldenticketawards.com/

https://www.lifed.com/top-25-best-roller-coasters-in-the-world/

https://coasterbuzz.com/

7.2 Code for GRR and PRR

```
library(tidyverse)
library(lubridate)
library(ggplot2)
library(ggthemes)
library(corrplot)
comapdata=read_csv("COMAP_RollerCoasterData_2018.csv")
comapdata$`Height (feet)`=as.numeric(comapdata$`Height (feet)`)
ecdf1=ecdf(comapdata$`Height (feet)`)
plot(ecdf1)
qqnorm(comapdata$`Height (feet)`)

comapdata$`G Force`=as.numeric(comapdata$`G Force`)
ecdf1=ecdf(comapdata$`G Force`)
plot(ecdf1)
qqnorm(comapdata$`G Force`)

comapdata$`Inversions (YES or NO)`[comapdata$`Inversions (YES or NO)`=="YES"]=1
comapdata$`Inversions (YES or NO)`[comapdata$`Inversions (YES or NO)`=="NO"]=0
comapdata$`Inversions (YES or NO)`=as.numeric(comapdata$`Inversions (YES or NO)`)
comapdata$`Duration (min:sec)`=as.numeric(comapdata$`Duration (min:sec)`)/60
comapdata$`Speed (mph)`=as.numeric(comapdata$`Speed (mph)`)
ecdf1=ecdf(comapdata$`Speed (mph)`)
plot(ecdf1)
qqnorm(comapdata$`Speed (mph)`)
```

```
cor(comapdata[complete.cases(comapdata[,10:19]),10:19])

ggplot(comapdata, aes(x=`Height (feet)`, y=`Speed (mph)`)) + geom_point(color="blue") +
theme_few()+geom_smooth(method="lm", se=TRUE, fullrange=FALSE, level=0.95)

ggplot(comapdata, aes(x=`Length (feet)`, y=`Speed (mph)`)) + geom_point(color="blue") +
theme_few()+geom_smooth(method="lm", se=TRUE, fullrange=FALSE, level=0.95)


ggplot(comapdata, aes(x=`Duration (min:sec)`*`Speed (mph)`, y=`Length (feet)`)) + geom_point(color="blue")
+ theme_few()+geom_smooth(method="lm", se=TRUE, fullrange=FALSE, level=0.95)



lm1<-lm(`Speed (mph)`~`Height (feet)`,data=comapdata)


lm2<-lm(`Length (feet)`~`Duration (min:sec)`:`Speed (mph)`,data=comapdata)


lm3<-lm(`Drop (feet)`~`Speed (mph)`,data=comapdata)


ggplot(comapdata, aes(x=`Number of Inversions`, y=`Drop (feet)`)) +

 geom_point(color="blue") +

 theme_few()+geom_smooth(se=TRUE, fullrange=FALSE, level=0.95)



ggplot(comapdata, aes(x=`Length (feet)`/`Speed (mph)`, y=`Duration (min:sec)`)) +

 geom_point(color="blue") +

 theme_few()+geom_smooth(method="lm", se=TRUE, fullrange=FALSE, level=0.95)


lm4<-lm(`Duration (min:sec)`~I(`Length (feet)`/`Speed (mph)`)+I(`Drop (feet)`/`Speed
(mph)`)+Construction,data=comapdata)
```

```
comapdata=read_csv("COMAP_RollerCoasterData_2018_Fixed.csv")[1:299,1:19]

lm5<-lm(GForce~Length+Construction,data=comapdata)

length(lm5$fitted.values)
comapdata$GForce=predict(lm5,newdata = comapdata)

ggplot(comapdata, aes(x=`Length (feet)`, y=`G Force`)) +
  geom_point(color="blue") +
  theme_few()+geom_smooth(method="lm", se=TRUE, fullrange=FALSE, level=0.95)

ggplot(comapdata, aes(x=`Length (feet)`, y=`Vertical Angle (degrees)`)) +
  geom_point(color="blue") +
  theme_few()+geom_smooth(method="lm", se=TRUE, fullrange=FALSE, level=0.95)


lm6<-lm(Vangle~GForce,data=comapdata)

comapdata$Vangle=predict(lm6,newdata = comapdata)
write_csv(comapdata,"COMAP_RollerCoasterData_2018_Full.csv")
```

```
comapdata=read_csv("COMAP_RollerCoasterData_2018_Full.csv")

comapdata[,11:19]=scale(comapdata[,11:19])

comapdata=comapdata[comapdata$Construction=="Steel",]

# comapdata$Construction=as.numeric(factor(comapdata$Construction))-1

comapdata$Construction=NULL

comapdata$Status=NULL

comapdata$Type=NULL

comapdata$Year=NULL

comapdata$IInversion=NULL

comapdata$Park=NULL

comapdata$District=NULL

comapdata$City=NULL

comapdata$Region=NULL


weights=c(3/sqrt(83),1/(4*sqrt(83)),1/(sqrt(83)))

coefs=c(weights[1]/3,weights[1]/3,weights[2]/4,weights[3],weights[1]/3,weights[2]/4,weights[2]/4,weights[2
]/4)

ans=as.vector(data.matrix(comapdata[,3:10])%*%coefs)-weights[2]*(0.7620/sqrt(0.21)+26.2/sqrt(2139.5))



comapdata=read_csv("COMAP_RollerCoasterData_2018_Full.csv")

comapdata[,11:19]=scale(comapdata[,11:19])

comapdatawood=comapdata[comapdata$Construction=="Wood",]

# comapdata$Construction=as.numeric(factor(comapdata$Construction))-1

comapdatawood$Construction=NULL

comapdatawood$Status=NULL
```

```
comapdatawood$Type=NULL

comapdatawood$Year=NULL

comapdatawood$IInversion=NULL

comapdatawood$Park=NULL

comapdatawood$District=NULL

comapdatawood$City=NULL

comapdatawood$Region=NULL


weights=c(3/sqrt(83),1/(4*sqrt(83)),1/(sqrt(83)))

coefs=c(weights[1]/3,weights[1]/3,weights[2]/4,weights[3],weights[1]/3,weights[2]/4,weights[2]/4,weights[2
]/4)

ans2=as.vector(data.matrix(comapdatawood[,3:10])%*%coefs)


comapdata=read_csv("COMAP_RollerCoasterData_2018_Full.csv")

comapdata$ans=0

comapdata$ans[comapdata$Construction=="Steel"]=ans

comapdata$ans[comapdata$Construction=="Wood"]=ans2




corrplot(cor(comapdata[,10:19]))

comapdataold=read_csv("COMAP_RollerCoasterData_2018.csv")

colnames(comapdataold)<-colnames(comapdata)

comapdataold$Height=as.numeric(comapdataold$Height)

comapdataold$IInversion[comapdataold$IInversion=="YES"]=1

comapdataold$IInversion[comapdataold$IInversion=="NO"]=0

comapdataold$IInversion=as.numeric(comapdataold$IInversion)
```

```
comapdataold$Duration=as.numeric(comapdataold$Duration)/60

comapdataold$Speed=as.numeric(comapdataold$Speed)

corrplot(cor(comapdataold[complete.cases(comapdataold[,10:17]),10:17]))

corrplot(cor(comapdata[,10:19]))
```