




+

## Why the topic was chosen

- We like cars, we like fast and furious movies, we like races, but we don't like sorting large sets of data
- **GOAL:**
- Group cars in different categories (heats) that would consider similar variables.
- **ISSUE:**
- We had a large dataset of unique cars with multiple variables
- It would be impossible to properly sort all of the cars into their groups for the race
- Given the amount of variables it was hard to determine what each group should be based on. There was no clear pattern.



- **SLIDE FIVE - WHY THE TOPIC WAS CHOSEN**
- We chose this topic because of a mutual interest in cars, fast and furious movies and because we dislike having to sort large datasets.
- Our goal was to group cars in different categories or heats, that would consider similar variables and have them race against each other. By grouping cars into different categories we could guarantee that they compete in a fair manner.
- After finding a dataset that included multiple variables for over 10 thousand cars we thought it would be interesting to find a way to sort this dataset in a manner they could race against each other in a fair way. Our goal was to group cars in different categories or 'heats' that would consider similar variables.
- That being said, because the dataset included so many variables we had no clear pattern to follow. We realized that it would be unfair to only consider one or two variables for our goal.
- For example, if we only considered horsepower, it would be unfair to pair a Porsche Cayman against a Hummer, even if both cars have 300 horsepower.
- The same thing happens if we chose to group cars by style and pair a Pontiac Le Mans from 1991 against a Bugatti Veyron. Both are categorized as coupes but it is clear the Bugatti would have an unfair advantage.

## VARIABLES USED & DEFINITIONS

- **AGE** - Based on minimum age of car from the date it was first released
- **FUEL TYPE** - Type of fuel the cars engine uses (Unleaded ,Diesel, Electric, etc.)
- **HORSEPOWER** - Unit of Power.
- **CYLINDER** (Count) - The amount of cylinders used by the engine
- **TRANSMISSION TYPE** - Type of gear system the car uses
- **WHEEL DRIVE** - Which wheels are primarily powered by the engine
- **SIZE** - A general overview of the Cars volume and weight
- **STYLE** - A general overview of the shape of the car (In regards to air resistance)
- **HIGHWAY MPG** - Amount Gasoline used in a stopless ride

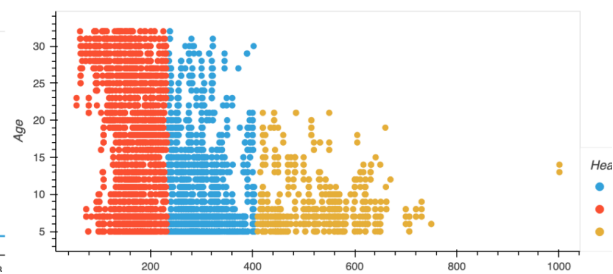
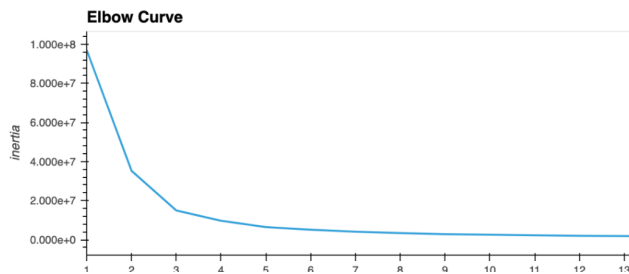


### • SLIDE SIX - VARIABLES USED & DEFINITIONS

- So before we jump into our model we want to break down the variables we chose to consider and explain them briefly.
- Age - This one is straight forward. This is the difference between the current year and the year the car was built.
- Fuel Type - the type of fuel a particular car uses such as unleaded, diesel and electric.
- Horsepower - unit of power used to measure the power of a car engine.
- Cylinder (count) - the amount of cylinders used by the engine. the more cylinders an engine possesses, the faster power can be generated.
- Transmission type - manual vs automatic
- Wheel Drive - which wheels are powered by the engine.
- Size - General overview of the car's volume and weight
- Style - shape of the car, like coupe, sports car, sedans and SUVs.
- Highway MPG - amount of gasoline used in a stopless ride

# CLUSTERING WITH K-MEANS

- Model pick: Clustering and K-means.
- Determining how many clusters or K values were needed.
- Trial and Error was used along with “elbow data” to determine best groupings
- We later opted for PCA because we had too many columns and needed a better way to group the clusters
- Finding K.
- PCA values were “joined” to the original data set to make interpretation easier and presentable



## SLIDE SEVEN - ANALYSIS & CLUSTERING WITH K-MEANS

- After we set out to group cars by multiple variables we decided that the best option for our model would be to do Clustering with the K-Means algorithm.
- As a refresher the K-means algorithm groups a dataset into clusters, or K values. In this case, each one of these clusters holds different cars with similar variables. So for example, the graph on the right is showing three clusters.
- One of the disadvantages Kmeans had for us was that we had too many variables to be able to define the ideal number of clusters. To try to solve this, we did some trial and error along with an elbow plot but eventually opted to use Principal Component Analysis or PCA.
- PCA as a refresher as well, is a technique that helps us define the number of clusters when the number of features is too high. algorithms when the number of input features (or dimensions) is too high. PCA reduces the number of dimensions by transforming a large set of variables into a smaller one that contains most of the information in the original large set.
- Once we used PCA we found that our ideal K would be four and proceeded to plot our results.
- Scaling = we take each feature value, subtract the mean of that feature and then divide by the feature standard deviation. We use sklearn's StandardScaler algorithm.
- Clustering is a type of unsupervised learning that groups data points together. This group of data points is called a cluster.

- K-means is an unsupervised learning algorithm used to identify and solve clustering issues.
- K represents how many clusters there will be. These clusters are then determined by the means of all the points that will belong to the cluster.
- The K-means algorithm groups the data into K clusters, where belonging to a cluster is based on some similarity or distance measure to a centroid.
- The centroid is found by taking the mean of all the x values in a cluster, and the mean of all the y values in a cluster.
- Inertia measures the amount of variation in the dataset.
- Reason not to choose Hierarchical Clustering
- The weaknesses are that it rarely provides the best solution, it involves lots of arbitrary decisions, it does not work with missing data, it works poorly with mixed data types, it does not work well on very large data sets, and its main output, **the dendrogram, is commonly misinterpreted.**

### Model Steps

1. Decide how many clusters you want, i.e. choose k
2. Randomly assign a centroid to each of the k clusters
3. Calculate the distance of all observation to each of the k centroids
4. Assign observations to the closest centroid
5. Find the new location of the centroid by taking the mean of all the observations in each cluster
6. Repeat steps 3-5 until the centroids do not change position